University of Zurich^{UZH}

MSc Thesis September 13, 2020

Mining Data Management Tasks in Computational Notebooks: an Empirical Analysis

Santiago Miguel Cepeda
of Zurich, Switzerland

Student-ID: 12-741-385
santiagomiguel.cepeda@uzh.ch

Advisor: **Cristina Sarasua**

Prof. Abraham Bernstein, PhD
Institut für Informatik
Universität Zürich
http://www.ifi.uzh.ch/ddis

Acknowledgements

I would like to thank Prof. Abraham Bernstein for giving me the opportunity to write my thesis at the Dynamic and Distributed Information System Group of the University of Zurich.

I would also like to give my sincerest gratitude to Cristina Sarasua, who was my supervisor for the duration of this thesis. She went above and beyond to give me the right guidance and tools that were necessary for me to do my work. Furthermore, she not only made sure that I always stayed on track, but her constant support and valuable insights were invaluable to this thesis.

Zusammenfassung

Das Ziel dieser Arbeit ist, das Verständnis darüber zu vertiefen, wie Datenwissenschaftler arbeiten und dies insbesondere im Hinblick auf die Aufgaben des Datenmanagements. Die Motivation hinter dieser Arbeit ist, die vorherrschende Lücke in Bezug auf die mangelnde empirische Evidenz zu den konkreten Datenmanagementaufgaben in der Datenwissenschaft zu füllen. Ebenfalls von Interesse ist zu erkennen, welche Rolle die Datenmanagementaufgaben in Bezug auf den gesamten datenwissenschaftlichen Prozess spielt. Darüber hinaus wird das Hauptaugenmerk auf die Analyse spezifischer Datenbereinigungs- und Datenintegrationsaufgaben innerhalb des Datenmanagements gelegt. Dieses Ziel wird durch Etikettierung, Data-Mining und die Anwendung statistischer Tests auf Daten-Wissenschaft-Notebooks aus der realen Welt erreicht. Dabei erhält man ein Schlüsselwort-Kennzeichnungssystem, das in der Lage ist, mehrere Arten von Zellen innerhalb von Daten-Wissenschaft-Notebooks zu identifizieren und zu kennzeichnen. Es resultieren drei verschiedene Datensätze. Es handelt sich dabei um einen Datensatz für jeden Notebook-Typ, der im Rahmen dieser Arbeit identifiziert wird: einfach deskriptiv, deskriptive und prädiktive Daten-Wissenschaft-Notebooks. Auf der Grundlage der empirischen Analyse kann der Schluss gezogen werden, dass es im Durchschnitt 6,56 Gesamtaufgaben zur Datenbereinigung und 5,38 Gesamtaufgaben zur Datenintegration pro Notebook über alle Notebooktypen hinweg gibt. Darüber hinaus werden im Durchschnitt je nach Notebook-Typ zwischen 5,7 und 6,9 Dateien innerhalb eines Notebooks importiert. Die Ergebnisse deuten auch darauf hin, dass die Datenbereinigung in einem datenwissenschaftlichen Projekt, je nach Notebook-Typ im Durchschnitt nur zwischen 10,18% bis 10,98% eines ganzen Data-Mining Notebooks ausmacht. Bei Datenintegrationsaufgaben sind es zwischen 9,55% bis 11,31%. Die empirische Evidenz unterstützt die Behauptung von Krishnan et al. (2016), dass Datenbereinigung ein nichtlinearer und iterativer Prozess ist. Diese Masterarbeit kommt zum Schluss, dass auch die Datenintegration ein nichtlinearer und iterativer Prozess ist.

Abstract

The aim of this thesis is to further our understanding of how data scientist work, specifically with regards to data management tasks. The motivation behind this goal is the prevalent gap in respect to the empirical evidence showcasing concrete data management tasks in data science, and the role which it plays in relation to the entire data science process. Furthermore, the main focus has been narrowed down to analyze specifically data cleaning and data integration tasks within data management. This goal was achieved by labelling, mining and applying statistical tests to real-world data science notebooks. A keyword labelling system was created in the process, which was able to identify and label multiple types of cells within notebooks. The end results were three different annotated datasets. This constitutes one dataset for each notebook type identified during this thesis: simple descriptive, descriptive mining and predictive mining notebooks. Based on the empirical analysis, it can be concluded that on average there are 6.56 total data cleaning tasks, and 5.38 total data integration tasks per notebook across all notebook types. Furthermore, there are on average between 5.7 to 6.9 files being imported inside of a notebook. The results also indicate that data cleaning amounts on average between 10.18% and 10.98% of an entire notebook, depending on the notebook type . For data integration tasks it is between 9.55% and 11.31%. This research also backs Krishnan et al. (2016) claim that data cleaning is a non-linear and iterative process. Moreover, this thesis has shown that data integration as well, is a non-linear and iterative process.

Contents

1

Introduction

1.1 Motivation

With the advancement of digitalization came an enormous supply of digital data, which continues to increase with every passing day. Data science has established itself as an important discipline within this context, since it combines a variety of techniques to extract important insights from vast amounts of data points. This is achieved by combining multiple methods, which range from data management to applied statistics (Schutt and O'Neil, 2013). However, there is still limited knowledge regarding the best practices prevalent in data science, as well as the challenges that data scientists face in academia and in the industry. An interesting insight brought forth by multiple surveys indicates that data scientists spend the majority of their time cleaning and organizing their data, as opposed to mining for patterns (CrowdFlower, 2016) (Review, 2018). However, there is still a gap when it comes to the empirical evidence illustrating concrete data management tasks in data science, and the role that data management assumes with regards to the entire data science process. Therefore, the aim of this thesis is to mine real-world data science notebooks in order to identify specific data management tasks applied by data scientists. The main focus will be on data cleaning tasks, and possibly data integration tasks.

Related works, such as the one by Rule et al. (2018) who empirically analyzed the data science process, and the work of a master thesis intended to classify cells in a data science notebook by Ramasamy (2019), will be considered.

1.2 Research Questions

The goal of this master thesis is to empirically analyze data science notebooks in terms of data management tasks. The research questions addressed are the following:

1. How many data sets do data scientists analyze simultaneously?
2. What are the least and most frequent data cleaning tasks?
   (a) What are the least and most frequent data cleaning tasks in the outlier detection versus clustering group?

   (b) What are the least and most frequent data cleaning tasks in the simple statistics versus statistical tests group?
   (c) What are the least and most frequent data cleaning tasks in the regression versus classification group?

3. What are the least and most frequent data integration tasks?
   (a) What are the least and most frequent data integration tasks in the outlier detection versus clustering group?
   (b) What are the least and most frequent data cleaning tasks in the simple statistics versus statistical tests group?
   (c) What are the least and most frequent data integration tasks in the regression versus classification group?

4. Are there differences between different types of data science notebooks?
   (a) Is the mean length of predictive notebooks equal to the mean length of simple and descriptive mining notebooks?
   (b) Is the mean of data cleaning tasks in predictive notebooks equal to the mean in simple descriptive and descriptive mining notebooks?
   (c) Is the mean of data integration tasks in predictive notebooks equal to the mean in simple descriptive and descriptive mining notebooks?
   (d) Is the mean of all data cleaning cells across all notebook types equal to the mean of all data integration cells across the three notebook types?

5. What is the percentage of code dedicated to data cleaning in different types of data science notebooks?

6. What is the percentage of code dedicated to data integration in different types of data science notebooks?

7. What is the relation between data cleaning and other data science steps?
   (a) Does data cleaning happen iteratively?
   (b) What is the ratio between data cleaning and predictive cells?
   (c) What is the ratio between data cleaning and descriptive mining cells?
   (d) What is the ratio between data cleaning and simple descriptive cells?
   (e) What is the ratio between data cleaning and data visualization cells?

8. What is the relation between data integration and other data science steps?
   (a) Does data integration happen iteratively?
   (b) What is the ratio between data integration and predictive cells?
   (c) What is the ratio between data integration and descriptive mining cells?
   (d) What is the ratio between data integration and simple descriptive cells?
   (e) What is the ratio between data integration and data visualization cells?

1.3 Contributions of the Thesis

The contributions of this master thesis are the following:

1. Descriptive statistics on data management tasks, e.g., mean length of data cleaning tasks in different types of data science notebooks.

2. A keyword-based labelling method to identify predictive or descriptive data science notebooks, which also contain data management tasks. The method was evaluated for precision, recall and accuracy.

3. Three annotated datasets, which contain predictive or descriptive cells and data management activities.

1.4 Thesis Structure

The thesis is structured in the following way: Chapter 2 discusses the related work and the current background knowledge in the field. In Chapter 3, the concepts that are discussed throughout the thesis will be introduced. Chapter 4 illustrates the keyword-based labelling system used to preprocess the notebooks. Chapter 5 contains information regarding the dataset and the methodology. This is followed by the descriptive statistics as a result of the analysis. Chapter 6 discusses the results and the insights with more detail. The rest of the chapters wind up the thesis with a conclusion and a section on possible future work that could be investigated.

2

Background & Related Work

Methods such as surveys or interviews have been explored extensively in order to get a deeper understanding on how data scientists work, and specifically what they work on by Figure-Eight (2018), formerly known as CrowdFlower (2016), and Mooney (2018). However, the scientific literature still lacks empirical evidence on how the data science code is really implemented. Which is why this thesis tries to gain new insights with regards to data management tasks prevalent in data science within the context of computational notebooks, and especially from the source code generated by these literate programming tools. This chapter will give some background and discuss the existing literature relevant to the data science process, data management tasks and computational notebooks.

2.1 Data Science Process

Understanding the data science process is important in order to be able to identify which cell inside of a computational notebook pertains to which step of the data science process. While there are many sources attributing different activities to a data science process, Figure 2.1 shows a common pipeline as used in practice.

Schutt and O'Neil (2013) partition the data science process into different steps:

1. Raw data is collected from different sources.

2. The raw data is then processed and prepared by using various data wrangling methods (eg. joining, scraping, etc.).

3. The data is cleaned of outliers, duplicates and inconsistencies (e.g. formatting and missing values).

4. A primary exploratory data analysis is conducted. The point of this data exploration is to find possible relationships between variables. At this point of the process, it may become apparent that some part of the data might be missing or that the data is in need of more cleaning. If this is the case then collecting new data might be necessary, or an increase in the amount of time allocated to cleaning the data.

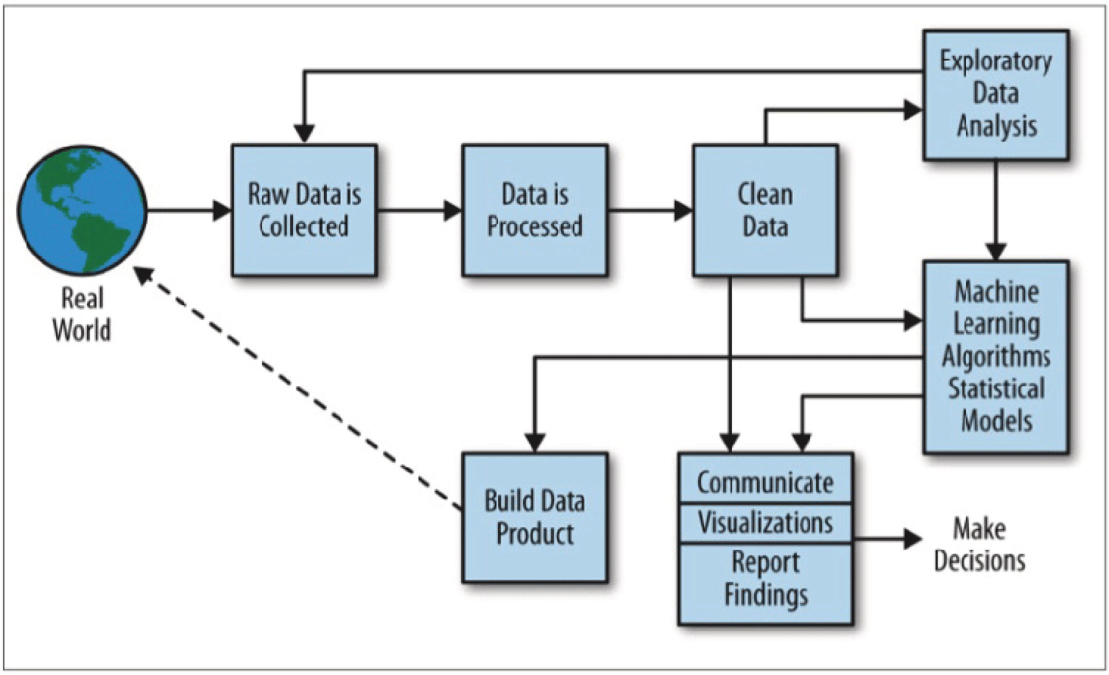

Exploratory Data Analysis
Raw Data is Collected
Data is Processed
Clean Data
Real World
Machine Learning Algorithms Statistical Models
Build Data Product
Communicate Visualizations Report Findings
Make Decisions

Figure 2.1: Data science process as illustrated by Schutt and O'Neil (2013)

5. Different machine learning algorithms or statistical models can be applied depending on the type of task (e.g. descriptive or predictive).

6. The results are then interpreted, visualized and communicated.

Data management tasks, such as data cleaning and data integration play a very important role in the data science process. Mostly because a cleaned, and properly integrated dataset provides a better basis for the exploration of the data and extraction of insights. However, depending on the type of data science task, it is not necessary that all of the steps above are present in a data science pipeline (Wang et al., 2019). For instance, predictive modelling is not necessary in order to get a simple descriptive statistic, but it would be a data science pipeline nonetheless.

2.1.1 Data Mining Tasks

Tan et al. (2016) identified two different types of data mining activities, which helps with the identification of the different types of data science notebooks prevalent in the industry and academia:

Descriptive Mining Tasks. The goal of descriptive mining tasks is to derive patterns from the data, such as correlations, trends, clusters, trajectories and anomalies, which summarize the underlying relationship between the variables (Tan et al., 2016). These type of tasks are mostly exploratory in nature (Tan et al., 2016). Furthermore, simple descriptive statistics which describe the data, such as the mean and standard deviation, have also been added under this definition.

Predictive Mining Tasks. The aim of predictive mining tasks is to predict the value of a dependent variable, through the value of an independent variable. These can either be a classification task for discrete target variables, or regression for continuous target variables (Tan et al., 2016).

2.2 Data Management Tasks

Data management is an umbrella term for a wide variety of tasks that ensure data is of high quality, consistent and retrievable throughout an organization (Cupoli et al., 2014). For the purpose of this thesis the analysis of data management tasks will be limited to data cleaning and data integration.

2.2.1 Data Cleaning

Data wrangling or to be more specific, data cleaning has been shown to require up to 80% of the time in a data science project (Furche et al., 2016). Data cleaning is necessary, because incorrect or inconsistent data can distort the results of an analysis (Hellerstein, 2008). Furthermore, data cleaning is not only limited to correcting inconsistent data, but also about preparing the data appropriately and making sure that the right data points are selected. Data scientists mention in interviews that they make decisions on, e.g., how to treat missing values through having domain-specific knowledge, knowing the patterns of the data and through conversations with the data (Muller et al., 2019). Muller et al. (2019) state that ground truth data also requires cleaning or ”grooming” in some special cases. Sutton et al. (2018) mention that analyses are usually repeated over multiple data samples, and subsequently proposed diff-like transformations for data cleaning. This thesis tries to expand the evidence within this context, and analyze the different data cleaning tasks present in the data science notebooks, while evaluating the percentage of code dedicated to data cleaning in different types of data science notebooks.

Krishnan et al. (2016) surveyed data scientists and discovered that data cleaning is often a non-linear and iterative process. This process can be ad-hoc, and lacks any proper methodology to evaluate if the data has been sufficiently cleaned. Moreover, the authors mention that cleaning data iteratively could lead to over-fitting, because the data is cleaned until a specific output is achieved. This thesis analyzes if data cleaning happens iteratively, and how data cleaning is in relation with the other data science steps. Dasu and Loh (2012) state that cleaning data might have an impact on the statistical properties of the underlying distribution of the data, which implies that cleaner data does not necessarily have to be more useful. The authors also tested cleaning strategies, using criteria like statistical distortion, glitch improvement and cost, and came to the conclusion that a simple imputation method performed better than a sophisticated algorithm that relied on assumptions not suited for the data. This thesis will not calculate the performance of different data cleaning algorithms, but it will count the occurrences of different data cleaning tasks.

2.2.2 Data Integration

Miller (2017) mentions that the value of data increases when two data sets are integrated. Hellerstein (2008) states that it is unusual for large and aged databases to contain data from a single source. Furthermore, Hellerstein (2008) also mentions that databases evolve by integrating pre-existing databases over time, which is a source of inconsistencies within the new data set. Moreover, Feinberg et al. (2017) demonstrate that data scientists often have to deal with complicated translations during dataset integrations. Feinberg et al. (2014) also state that there is a tendency to exclude non-conformant data while integrating data sets, which might lead to elisions. This thesis does not calculate the value created by the data integration, nor the evolution of a database over time, but it does focus on the different data integration tasks prevalent in the data science notebooks, and how many data sets are analyzed by data scientists simultaneously.

2.3 Analysis of Computational Notebooks

Computational notebooks, such as Jupyter Notebooks, Mathematica, and others are widely used by data scientists for their ability to combine both code and documentation together with visualizations in a single document (Rule et al., 2018). Whereby the goal is not only to perform analysis, but also to document and share the insights.

2.3.1 Exploring Computational Notebooks

Kery et al. (2018) discovered three typical use cases that are not necessarily disjoint from each other for literate programming tools: (1) preliminary and short-lived work, (2) code that ends up extracted for production pipeline, (3) and computational notebooks meant to be shared. Furthermore, the authors state that data scientists often create a narrative structure during their exploration to tell a story of their analysis. Moreover, Rule et al. (2018) analyzed over 1 million computational notebooks from GitHub, and the authors found a tension between data exploration and explanation process in computational notebooks. According to Rule et al. (2018), a quarter of the notebooks had no text, and the median notebook had barely more text than the abstract of a research paper. The authors also noted that computational notebooks in their corpus rarely stood alone, but were rather in repositories that contained other notebooks or a README file. They also suggest that a single narrative may occur through combining multiple notebooks together, whereby each notebook contains a part of the data science process. Moreover, Rule et al. (2018) state that nearly half (43.9%) of the notebooks that were analyzed had an iterative nature, which was explained by the code that had a non-linear execution order. No research has been published within the context of analyzing data management tasks in computational notebooks to the best of my knowledge.

2.3.2 Classifying Computational Notebook Cells

The work of the master thesis by Ramasamy (2019) automatically classifies cells and assigns the data science activity to each cell using machine learning methods. Multiple supervised classifiers were evaluated using Singlelabel and Multilabel classification methods. The results show that logistic regression methods are more suitable for classification of computational notebooks. The researcher in this thesis used a keyword-based approach to label the cells, and to classify the notebooks instead. This was done with the intention to maximize the number of data science notebooks from the entire dataset. Furthermore, the focus of this thesis lies on extracting insights on data management tasks, and not the method itself. At a cell-level, this thesis uses the same granularity as Ramasamy (2019).

3

Preliminaries

This chapter will illustrate the format and structure of a computational notebook.

3.1 Parsing Computational Notebooks

While there are many literary programming tools, this thesis exclusively focuses on data available in Jupyter Notebooks. Jupyter notebook documents are self-contained JSON documents with an ".ipynb" extension, which are capable of carrying both inputs and outputs of computations. These inputs include narrative text, source code, and metadata. The rich media output is generated by the source code, and it extends to other type of formats such as HTML, images, videos and plots (Jupyter, 2015). Since a jupyter notebook is nothing but a JSON, it can be read and handled programmatically by all programming languages (Jupyter, 2015). Furthermore, each notebook has a kernel that runs the code inside of the notebook, while the kernel is started by the notebook web application [1].

3.1.1 Structure of a Computational Notebook

A jupyter notebook is essentially a dictionary with four keys[2]. However, there exist some older notebook formats that are also present in the data set, which contain an extra key named *worksheets* [3]. This key is a list that can contain multiple cells. The full list of keys are listed below next to their corresponding types:

1. metadata (dict)
2. nbformat (int)
3. nbformat_minor (int).
4. cells (list)

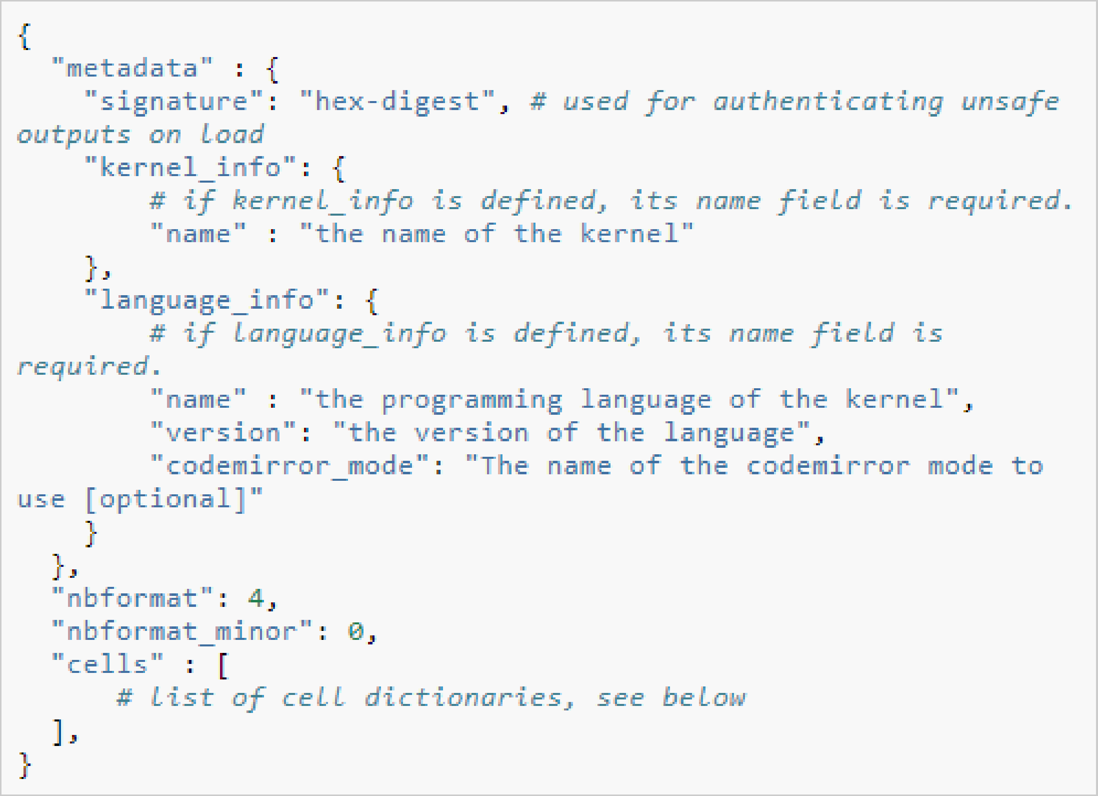
[1] *https://jupyter-notebook.readthedocs.io/en/latest/examples/Notebook/What%20is%20the%20Jupyter%20Notebook.html*

[2] *https://ipython.org/ipython-doc/dev/notebook/nbformat.html*

[3] *https://github.com/ipython/ipython/wiki/IPEP-17%3a-Notebook-Format-4*

5. worksheets (list) (*only in older versions*)

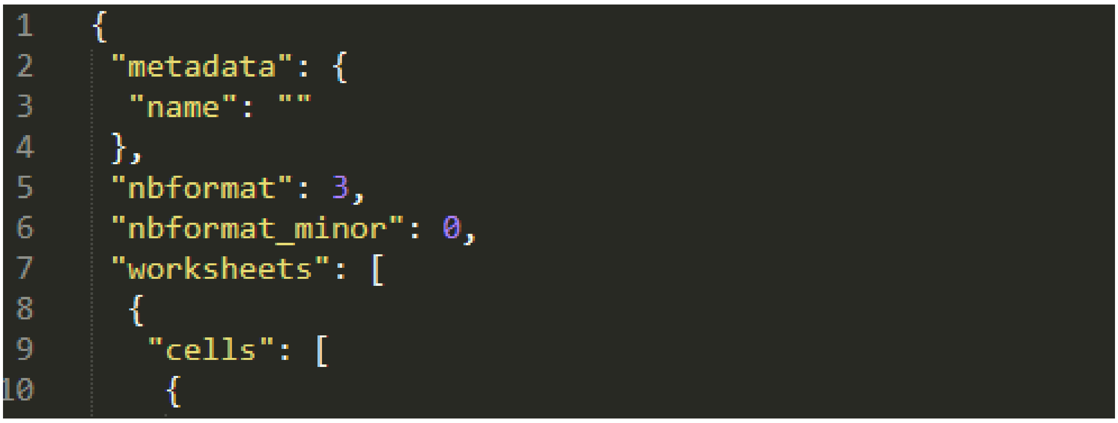
The first key is *metadata*, which contains: *signature*, *kernel_info* and *language_info*. *kernel_info* is a dictionary itself, that contains the name of the kernel, while *language_info* holds the name, version of the programming language and the name of the codemirror mode. The most interesting key is probably *cells*, because it is a list that can contain code, markdown or raw cells. Last but not least, both *nbformat* and *nbformat_minor* are keys that display the version of the notebook format. Figure 3.1 shows the structure of the Jupyter notebook JSON.



```
{
  "metadata" : {
    "signature": "hex-digest", # used for authenticating unsafe
outputs on load
    "kernel_info": {
        # if kernel_info is defined, its name field is required.
        "name" : "the name of the kernel"
    },
    "language_info": {
        # if language_info is defined, its name field is
required.
        "name" : "the programming language of the kernel",
        "version": "the version of the language",
        "codemirror_mode": "The name of the codemirror mode to
use [optional]"
    }
  },
  "nbformat": 4,
  "nbformat_minor": 0,
  "cells" : [
      # list of cell dictionaries, see below
  ],
}
```

Figure 3.1: Jupyter Notebook Structure. Source: https://ipython.org/ipython-doc/dev/notebook/nbformat.html

Figure 3.2 shows an example of an old notebook format containing the extra key called *worksheets* .



```
{
 "metadata": {
  "name": ""
 },
 "nbformat": 3,
 "nbformat_minor": 0,
 "worksheets": [
  {
   "cells": [
    {
```

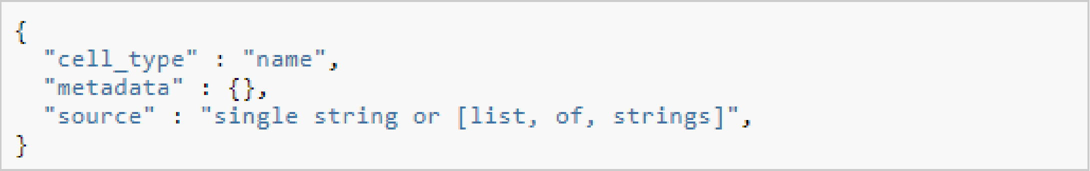
Figure 3.2: Example of an older Jupyter Notebook format present in the data set

3.1.2 Cell Types in Computational Notebooks

Jupyter notebooks have a linear sequence of cells that are stored inside of a list, which pertains to the key *cells*. There are different type of cells, some of which were made redundant in newer notebook formats:

1. Code cells: Input and output of source code
2. Markdown cells: Narrative text
3. Raw cells: Text that has not been formatted, which is included when notebooks are converted to different formats using nbconvert [4]
4. Strategy cells (*only in older versions*)
5. Heading cells (*only in older versions*)
6. Plaintext cells (*only in older versions*)

Each cell has the keys *cell_type*, *metadata* and *source* as shown in Figure 3.3.

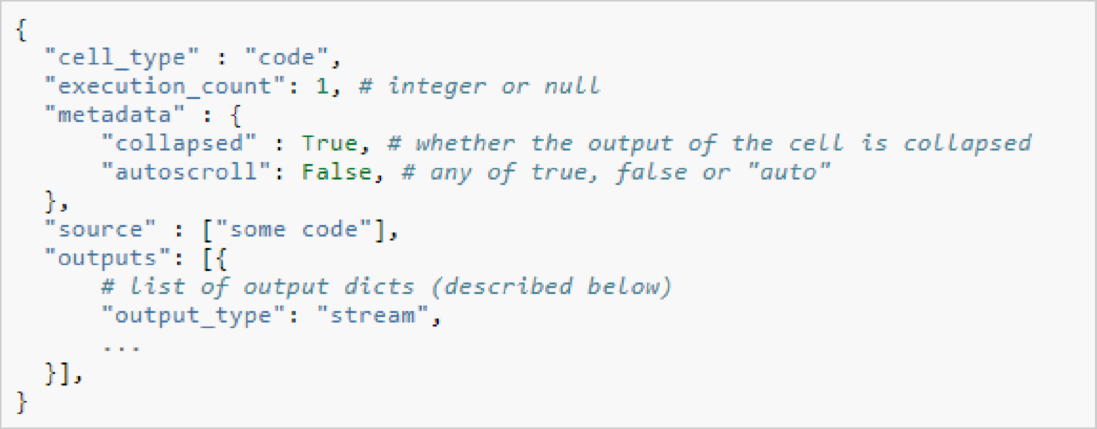


```
{
  "cell_type" : "name",
  "metadata" : {},
  "source" : "single string or [list, of, strings]",
}
```

Figure 3.3: Example of a basic cell structure. Source: https://ipython.org/ipython-doc/dev/notebook/nbformat.html

[4] *https://ipython.org/ipython-doc/dev/notebook/nbformat.html*

Code cells make up the main content of a notebook. The content of a code cell is source code, which is in the same programming language as the kernel of the notebook [5]. They also have a list of outputs, and an execution count. The structure of a code cell can be seen in Figure 3.4.



```
{
  "cell_type" : "code",
  "execution_count": 1, # integer or null
  "metadata" : {
      "collapsed" : True, # whether the output of the cell is collapsed
      "autoscroll": False, # any of true, false or "auto"
  },
  "source" : ["some code"],
  "outputs": [{
      # list of output dicts (described below)
      "output_type": "stream",
      ...
  }],
}
```

Figure 3.4: Structure of a code cell from https://ipython.org/ipython-doc/dev/notebook/nbformat.html

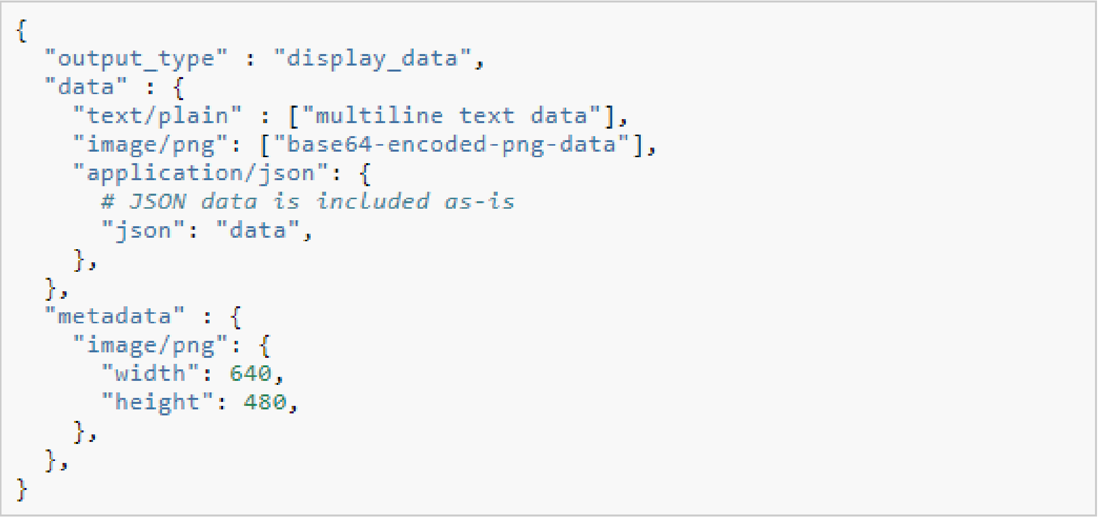
3.1.3 Types of Code Cell Output

Outputs are dictionaries inside of a list with the key *output_type*. There are multiple types of outputs for a code cell:

1. stream output: format is either stdout or stderr.
2. display_data: generated through display_data messages. Has data keyed by mime-type.
3. execute_result: contains results of a cell that was executed.
4. error: shows a traceback if an execution failed.

Figure 3.5 shows the structure of an output of the type display_data as an example.

[5] *https://ipython.org/ipython-doc/dev/notebook/nbformat.html*



```
{
  "output_type" : "display_data",
  "data" : {
    "text/plain" : ["multiline text data"],
    "image/png": ["base64-encoded-png-data"],
    "application/json": {
      # JSON data is included as-is
      "json": "data",
    },
  },
  "metadata" : {
    "image/png": {
      "width": 640,
      "height": 480,
    },
  },
}
```

Figure 3.5: Structure of a display_data output. Source: https://ipython.org/ipython-doc/dev/notebook/nbformat.html

4

Methodology

This chapter provides a detailed description of the dataset, labelling system and the mining process that was applied in order to answer the research questions mentioned in section 1.2.

4.1 Dataset

The GitHub dataset used in this thesis was prepared by Rule et al. (2018), who collected Jupyter notebooks that were available on Github. The dataset itself was published by UC San Diego[1]. It contains over 1,227,573 publicly available Jupyter notebooks from GitHub, which have not been forked from another repository. The metadata regarding the repository, and the corresponding README files are also available whenever those files were published by the authors in the GitHub source. The dataset consists mostly of Python notebooks, with a very small minority having R and Julia as their primary programming language. This dataset contains notebooks of diverse nature, including educational notebooks and homework submissions.

4.2 Data Preparation

A small subset of notebooks were selected from the GitHub dataset, which fulfilled the following criteria:

1. The programming language of the Jupyter notebook is in Python.
2. The notebook contains either simple descriptive, descriptive mining or predictive mining cells.
3. The notebook has at least one data cleaning and one data integration cell.
4. The notebook imports at least two files.
5. The notebook contains at least one function, and one variable.

These requirements were chosen in order to create a dataset, which is suited to answer the research questions mentioned in section 1.2.

[1] https://library.ucsd.edu/dc/object/bb2733859v

4.2.1 Notebook Labelling System

The aim of the keyword-based labelling system is to create a dataset, which adheres to the criteria mentioned earlier in the beginning of section 4.2. This is achieved with a set of labelling keywords, which are added or removed iteratively according to the evaluation of their precision, recall or accuracy. This set of keywords are explained further in section 4.3.1. Moreover, the keyword-based labelling system is based on an iterative bottom-up approach in order to maximize the amount of notebooks retrieved from the entire dataset, which fit the requirements. It is important to note that the labelling system has a different keyword group than the data mining keyword group. This is because the labelling keyword group was defined iteratively in order to get the largest amount of notebooks possible, which fit all five criteria. However, the data mining keyword group was not created iteratively, but rather based on taxonomies.

Labelling System

The keyword-based and iterative labelling system is shown in Figure 4.1. In a first step, the GitHub dataset is prepared, and a preliminary data exploration is performed. In a second step, the notebooks are being labelled with the keywords in an iterative manner. The results are evaluated manually, and checked for precision, recall and accuracy. After careful inspection, keywords are added or removed. This process is repeated multiple times. The result is an annotated data set, which is ready to be mined in order to provide results for the empirical analysis. The labelling system could be applied to different granularity levels: cell-level or line-level. However, this thesis will use the cell-level, because as Ramasamy (2019) mentioned in her master thesis, the cell-level is the unit of a Jupyter notebook, and because a data science activity needs more than one line to be implemented.

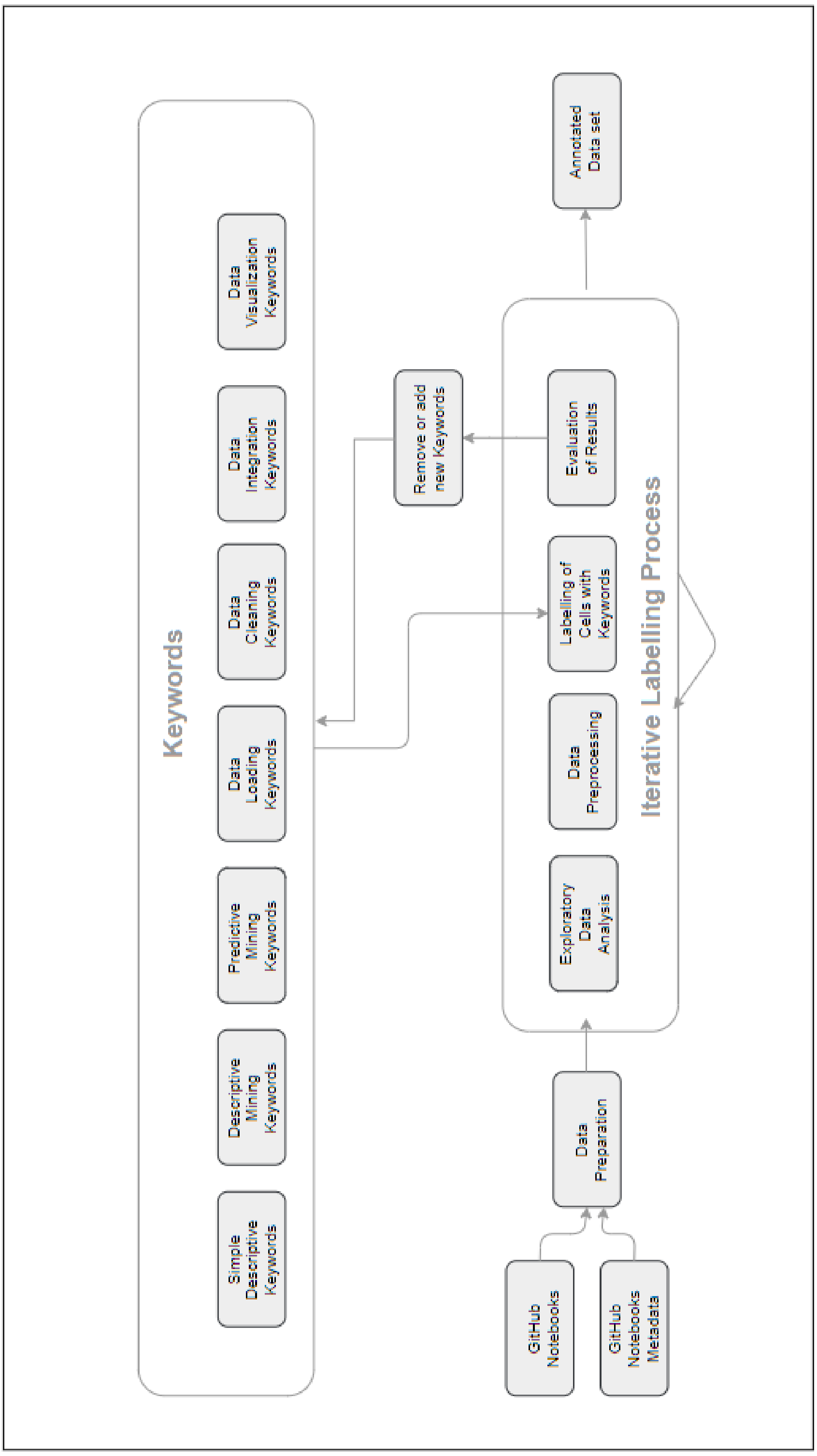

Keywords
Simple Descriptive Keywords
Descriptive Mining Keywords
Predictive Mining Keywords
Data Loading Keywords
Data Cleaning Keywords
Data Integration Keywords
Data Visualization Keywords
Remove or add new Keywords
GitHub Notebooks
GitHub Notebooks Metadata
Data Preparation
Iterative Labelling Process
Exploratory Data Analysis
Data Preprocessing
Labelling of Cells with Keywords
Evaluation of Results
Annotated Data set

Figure 4.1: Keyword-based labelling system

Notebook Types

Three different types of notebooks were derived from the definitions mentioned in section 2.1.1 for the categorization of data science notebooks. The three notebook types are listed below.

Simple Descriptive Notebooks. Simple descriptive notebooks contain exclusively simple statistics or visualizations, which describe the variables in the data. The set of simple statistics used to classify notebooks of this type are: mean, median, standard deviation, variance or similar. For instance, a notebook analysing a dataset for its min, max, mean and standard deviation using the *df.describe()* function in Python[2].

Descriptive Mining Notebooks. Descriptive mining notebooks are defined as notebooks that contain algorithms, which try to identify clusters, trajectories, anomalies, correlations or trends in the data. A simple descriptive statistic is not enough for a notebook to be considered as a descriptive mining notebook. For example, a notebook using the clustering algorithm *K-Means* in order to analyze text from school websites using Python[3].

Predictive Mining Notebooks. Predictive mining notebooks must contain a predictive model or an algorithm, which tries to classify or predict data. For instance, a notebook applying a *Decision Tree Classifier*, and a *Random Forest Classifier* on the data using Python[4].

Figure 4.2 shows the number of Python notebooks in the entire Github dataset, and the notebooks selected for each category.

[2] Example of a simple descriptive notebook: *https://github.com/mac475/15.07.01.01.kaggle.caterpillar/blob/f29ca6297e8b891fb9d8ddb678cfac6ab3f18e8f/15.07.12.01.dataset.processing.comp.verified.family.%ED%86%B5%ED%95%A9.ipynb*

[3] Example of a descriptive mining notebook: *https://github.com/sambarrows/school_websites/blob/c30b006b71708247ac2bacbbe16038c606dbe752/.ipynb_checkpoints/School_websites-checkpoint.ipynb*

[4] Example of a predictive mining notebook: *https://github.com/mitchellang/The-Cat-Demo/blob/93f8272dfb69d223a6c7a46c8baca249f8cf78d3/the-cat-yp-tree/Untitled.ipynb*

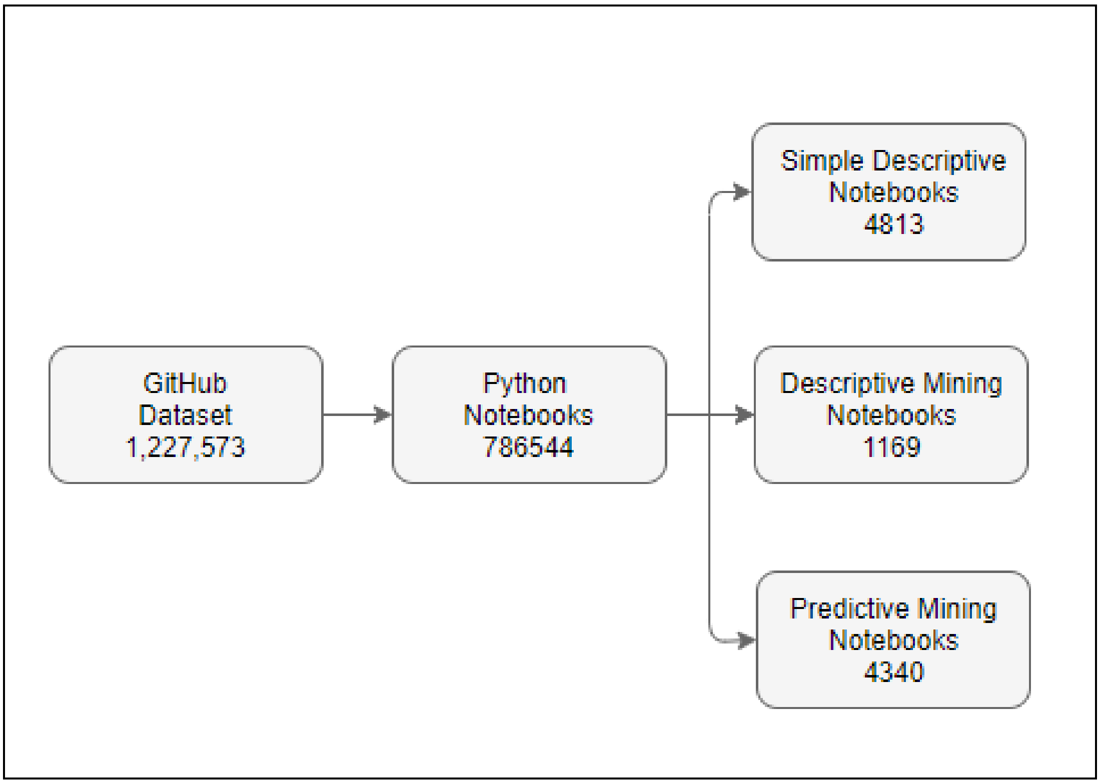


Figure 4.2: The three types of datasets with the number of notebooks per type.

Cell Types

It is necessary to define the different cell types, since the granularity of the labelling system is on a cell-level, and because they will be used to define the classification labels.

Data Loading Cells. Data loading cells contain keywords, which import files of different types: .csv, .mat, .pkl, .txt, .json, tables, .xls, SFrames and data retrieved by SQL queries.

Data Cleaning Cells. Data cleaning cells contain keywords, which actively clean the data. This ranges from removing duplicates to imputation. A data cleaning cell can contain multiple data cleaning tasks. More information is found in section 4.3.1.

Data Integration Cells. Data integration cells contain keywords, which indicate that at least two datasets are being merged together. A data integration cell can contain multiple data integration tasks. More information is found in section 4.3.1.

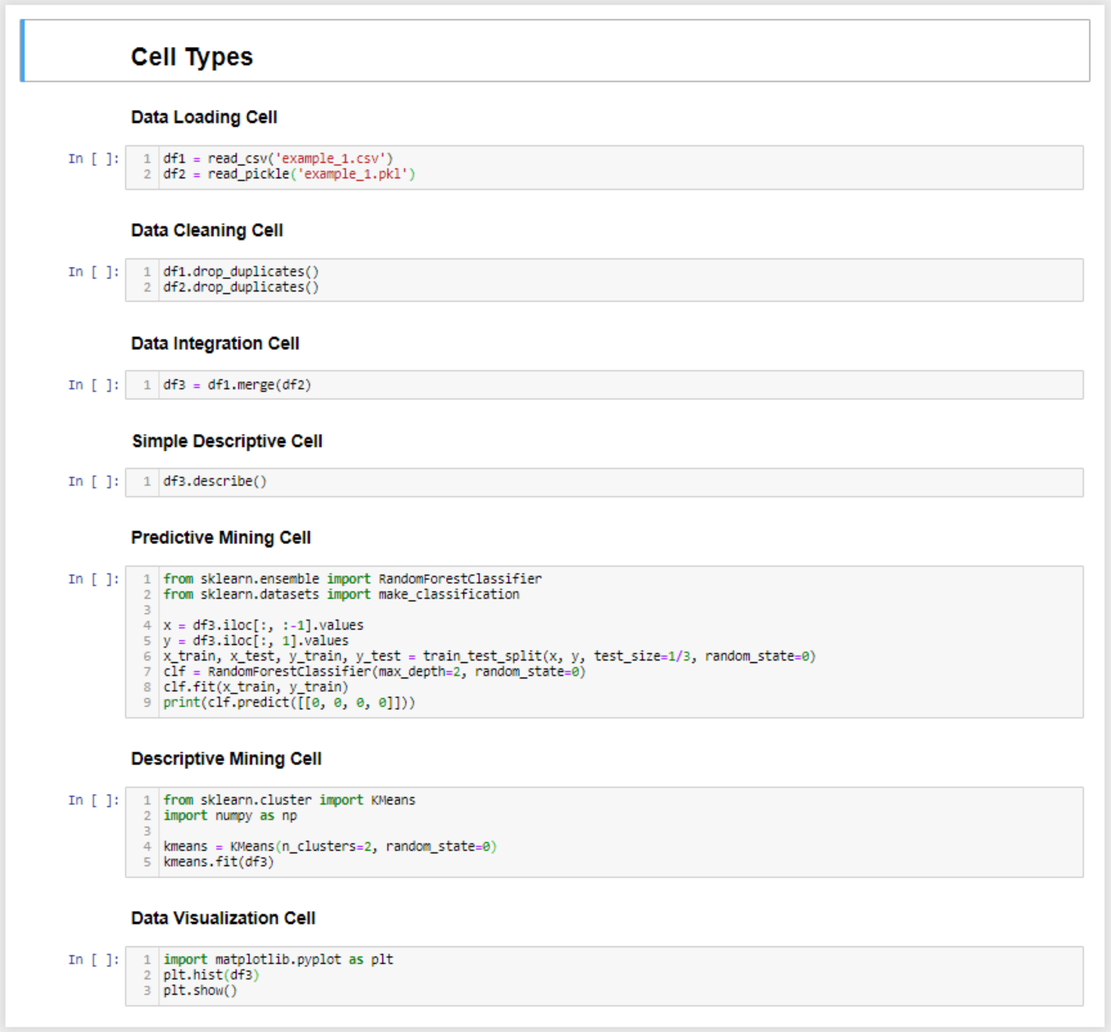
Data Visualization Cells. Data visualization cells contain keywords that plot different variables in the data.

Simple Descriptive Cell. Simple descriptive cells contain keywords with simple statistics or keywords that describe the data through visualization. However, the keywords for simple descriptive cells are disjoint from descriptive mining and predictive mining keywords.

Descriptive Mining Cell. Descriptive mining cells contain keywords of algorithms that are applied to the dataset in the notebook, such as clustering, outlier detection or similar. Simple descriptive statistics can appear in the descriptive mining cell, but are not enough to be labelled as a descriptive mining cell.

Predictive Mining Cell. Predictive mining cells contain at least one keyword with one predictive mining model.

Figure 4.3 shows some simple examples of each of the aforementioned cell types inside of a computational notebook.



Cell Types

Data Loading Cell

In [ ]:
```
df1 = read_csv('example_1.csv')
df2 = read_pickle('example_1.pkl')
```

Data Cleaning Cell

In [ ]:
```
df1.drop_duplicates()
df2.drop_duplicates()
```

Data Integration Cell

In [ ]:
```
df3 = df1.merge(df2)
```

Simple Descriptive Cell

In [ ]:
```
df3.describe()
```

Predictive Mining Cell

In [ ]:
```
from sklearn.ensemble import RandomForestClassifier
from sklearn.datasets import make_classification

x = df3.iloc[:, :-1].values
y = df3.iloc[:, 1].values
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=1/3, random_state=0)
clf = RandomForestClassifier(max_depth=2, random_state=0)
clf.fit(x_train, y_train)
print(clf.predict([[0, 0, 0, 0]]))
```

Descriptive Mining Cell

In [ ]:
```
from sklearn.cluster import KMeans
import numpy as np

kmeans = KMeans(n_clusters=2, random_state=0)
kmeans.fit(df3)
```

Data Visualization Cell

In [ ]:
```
import matplotlib.pyplot as plt
plt.hist(df3)
plt.show()
```

Figure 4.3: Example containing the different cell types.

Classification Labels

The purpose behind the classification labels is to be able to identify different types of notebooks, and which type of cells that it contains. Seven labels were chosen using the criteria in section 4.2. The labels themselves indicate if a specific activity is present in the notebook, while the entire content of the cell is stored in another variable.

isDataLoading. This label indicates that there is at least one data loading cell in the notebook.

isDataCleaning. There is at least one data cleaning cell in the notebook.

isDataIntegration. Two conditions have to be met: there are at least two datasets being imported, and there is one data integration cell in the notebook.

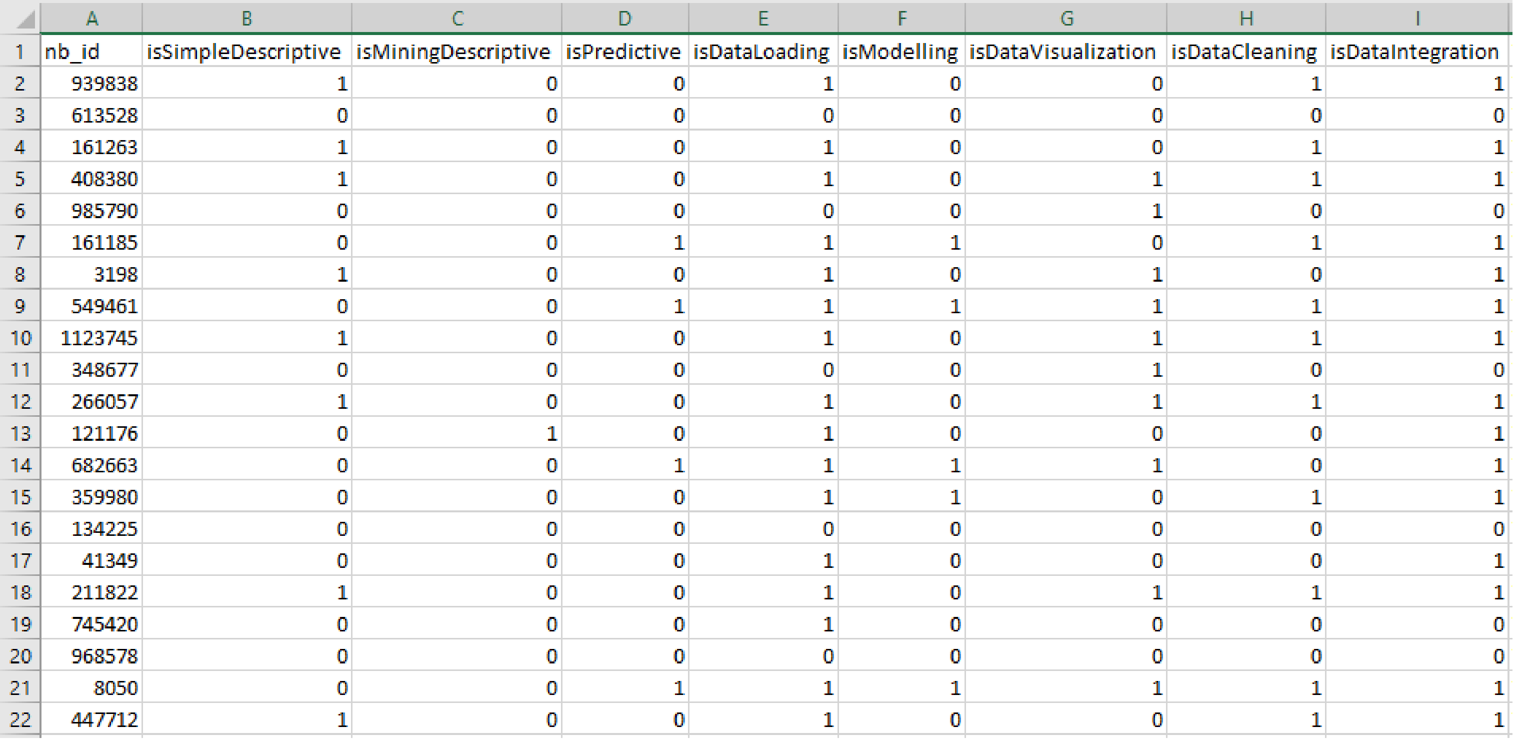
isDataVisualization. The notebook contains at least one data visualization cell.

isSimpleDescriptive. Three conditions have to be met: there is at least one simple descriptive cell, one data loading cell, and there are no descriptive mining or predicitive mining cells in the notebook.

isDescriptiveMining. Three conditions have to be met: there is at least one descriptive mining cell, one data loading cell, and there is no predictive mining cell in the notebook. Please note that it can contain simple descriptive cells, but it does not suffice to be labelled as a descriptive mining notebook.

isPredictive. This label has two requirements: there is at least one predictive mining cell, and one data loading cell. It can contain descriptive mining or simple descriptive cells, but it does not suffice to be labelled as a predicitve mining notebook.

The result of programmatically labelling the notebooks in the data set is documented in a file, which contains the value of all classification labels for each notebook. The classification labels can have a value of 0 (not true) or 1 (true). Figure 4.4 shows how a typical classification looks like.



	A	B	C	D	E	F	G	H	I
1	nb_id	isSimpleDescriptive	isMiningDescriptive	isPredictive	isDataLoading	isModelling	isDataVisualization	isDataCleaning	isDataIntegration
2	939838	1	0	0	1	0	0	1	1
3	613528	0	0	0	0	0	0	0	0
4	161263	1	0	0	1	0	0	1	1
5	408380	1	0	0	1	0	1	1	1
6	985790	0	0	0	0	0	1	0	0
7	161185	0	0	1	1	1	0	1	1
8	3198	1	0	0	1	0	1	0	1
9	549461	0	0	1	1	1	1	1	1
10	1123745	1	0	0	1	0	1	1	1
11	348677	0	0	0	0	0	1	0	0
12	266057	1	0	0	1	0	1	1	1
13	121176	0	1	0	1	0	0	0	1
14	682663	0	0	1	1	1	1	0	1
15	359980	0	0	0	1	1	0	1	1
16	134225	0	0	0	0	0	0	0	0
17	41349	0	0	0	1	0	0	0	1
18	211822	1	0	0	1	0	1	1	1
19	745420	0	0	0	1	0	0	0	0
20	968578	0	0	0	0	0	0	0	0
21	8050	0	0	1	1	1	1	1	1
22	447712	1	0	0	1	0	0	1	1

Figure 4.4: Labelling Sample

Labelling Keywords

The keywords were defined according to the definitions in section 4.2.1. A keyword was added or removed from a keyword group if it reduced the accuracy, recall or precision of the overall keyword group. It is important to note that the keywords are sensitive to casing as well as blank spaces, which is why different combinations might appear, e.g. *”merge(”* and *”merge (”*. Furthermore, the keywords usually contain opening parantheses to ensure that the keywords are functions. Each classification label has different keyword groups, but all keywords are in the Python programming language.

The full list of the labelling keywords can be found in the appendix in A.1, but table 4.1 shows a small sample for each keyword group. For instance, the keyword group isDataLoading displays four different ways to load data of different types. Whereas the four keywords listed in the isDataCleaning group try to clean the data by dropping duplicates *(”.drop_duplicates(”)*, dropping *(”.drop_na(”)* or filling NaN (Not a Number) values *(”fillna(”)* and by identifying null values *(”isnull(”)*. Moreover, the group isDataIntegration has keywords, such as *(”merge_ordered(”)*, which is designed for ordered datasets and *(”merge_asof(”)* that is similar to a left-join with the exception that it matches the nearest key. Furthermore, the group isDataVisualization has keywords, such as *(”.hist()”)* and *(”scatter_matrix(”)*. Their function is to plot a histogram or to draw a matrix of scatter plots. The isSimpleDescriptive group, e.g., has keywords that generate a table with descriptive statistics all at once *(”.describe()”)* or one by one: *(”std(”)*, *(”.min(”)* and *(”.max(”)*. Furthermore, the keywords for isPredictive and isDescriptiveMining are of common algorithms, such as *(”.IsolationForest(”)* and *(”LogisticRegression(”)*.

Table 4.1: Table with a sample of the keywords used for each keyword group.

isDataLoading	isDataCleaning	isDataIntegration	isDataVisualization
”read_csv(”, ”read_excel(”, ”read_table(”, ”read_pickle(”, ...	”dropna(”, ”isnull(”, ”fillna(”, ”drop_duplicates(”	”merge (”, ”merge(”, ”merge_ordered”, ”merge_asof”	”plt.show()”, ”scatter_matrix(”, ”.hist()”, ”plt.plot()”, ...

isSimpleDescriptive	isDescriptiveMining	isPredictive
”std(”, ”.max(”, ”.min(”, ”.describe(”, ...	”get_outliers_inliers(”, ”LocalOutlierFactor(”, ”EllipticEnvelope(”, ”IsolationForest(”, ...	”LogisticRegression(”, ”DecisionTreeClassifier(”, ”RandomForestClassifier(”, ”GradientBoostingClassifier(”, ...

Evaluation of the Labelling Results

Random samples of 50 notebooks were selected and verified manually for each keyword group to ensure that the results were in accordance with the criteria set in chapter 4.2. This was done 8 times in total, while adding and removing keywords, which increased or decreased the values of the metrics. The accuracy, recall and precision of the results are listed in table 4.2 for each label. These metrics were calculated as defined by Olson and Delen (2008). The corresponding formulas can be found below:

$$Precision = \frac{TP}{TP + FP} \tag{4.1}$$

$$Recall = \frac{TP}{TP + FN} \tag{4.2}$$

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{4.3}$$

where:

TP = Total amount of true positives in the sample

FP = Total amount of false positives in the sample

TN = Total amount of true negatives in the sample

FN = Total amount of false negatives in the sample

Table 4.2: Table with precision, recall and accuracy

Labels	Precision	Recall	Accuracy
isDataCleaning	96%	85%	90%
isDataIntegration	79%	100%	90%
isDataLoading	97%	81%	84%
isPredictive	100%	92%	96%
isDescriptiveMining	90%	96%	92%
isSimpleDescriptive	76%	86%	82%
isDataVisualization	95%	84%	90%

4.3 Analysis

This section gives a detail description of the taxonomy behind the mining keywords, and the methodology behind each research question.

4.3.1 Mining Process

The keyword-based labelling system explained in section 4.2.1. provides three annotated datasets of notebooks, which contain data management tasks, as well as notebooks with descriptive and predictive cells. The datasets are: simple descriptive, descriptive mining and predictive mining. In this section, the annotated datasets are mined in order to have concrete data and to be able to answer the research questions. The mining occurs through mining keywords, which are defined by the data cleaning and data integration taxonomies. These taxonomies are described further in the sections below.

Data Cleaning Taxonomy

The data cleaning taxonomy is the source for the data cleaning keywords, which are used to mine the annotated dataset. Some of the keyword groups are based on a data cleaning taxonomy by Corrales et al. (2018). The authors propose six data cleaning tasks: *Imputation*, *Outlier Detection*, *Dimensionality Reduction*, *Balanced Classes*, *Label Correction* and *Remove Duplicates.* The keyword groups *Data Transformation*, *Removing Redundancies* and *Unify Formatting* were defined while exploring the data. All data cleaning tasks used in this thesis are defined below:

Imputation. The aim of imputation is to fully remove missing data points in a dataset by filling them with computed values or removing all instances (Corrales et al., 2018).

Outlier Detection. Identifies outliers by detecting them through algorithms like Angle-Based Outlier Degree or Local Outlier Factor in order to remove them (Corrales et al., 2018).

Dimensionality Reduction. Identifies relevant attributes, which represent the dataset through dimensionality reduction algorithms, such as Principal Component Analysis (Corrales et al., 2018).

Balanced Classes. Uses under- or oversampling to eliminate instances from majority or minority classes (Corrales et al., 2018).

Label Correction. Removes instances or corrects the label if the instances with the same value have different classes (Corrales et al., 2018).

Removing Duplicates. Deletes duplicate values in the dataset (Corrales et al., 2018).

Removing Redundancies. Removes data points, which are not relevant within the context of the analysis.

Data Transformation. Transforms a variable of a dataset, e.g., datetime.

Unify Formatting. Synchronizes formatting between two datasets, which are about to be integrated.

The data cleaning task *Label Correction* was defined to be out-of-scope for this thesis, because there was no set of keywords that matched the complexity of this specific data cleaning task one-to-one without having to add even more complexity to the keyword-based mining system.

Data Integration Taxonomy

Specific keyword groups were defined to identify the different data integration cells within the annotated dataset. The data integration keyword groups were defined while exploring the data. All data integration tasks used in this thesis are defined below:

Merging. Merging occurs when at least two data sets are joined together.

Dataset Comparison. Dataset comparison occurs when to datasets are evaluated against each other before they are integrated.

Mining Keywords

The data cleaning and data integration keywords used to mine the annotated datasets in order to answer the research questions are available in the appendix in A.2. Table 4.3 displays a sample of the data cleaning keywords. The keyword group Imputation shows three different ways to impute the data. For instance, *”SimpleImputer(”* is used to complete missing values through the use of the mean or median, while *”IterativeImputer(”* uses functions of other features and *”KNNImputer(”* makes use of k-Nearest Neighbours algorithm to do the same (Pedregosa et al., 2011). The three keywords listed in the Outlier Detection group contain algorithms designed to detect outliers through different methods with the purpose of removing them from the dataset. *”LocalOutlierFactor(”*, e.g., detects data points by measuring local deviations of data points with regards to its neighbours (Breunig et al., 2000). Furthermore, the keyword group Dimensionality Reduction has three keywords, which apply different variations of the Principal Component Analysis in order to reduce the dimensions of the dataset: *”PCA(”*, *”IterativePCA(”* and *”KernelPCA(”*. Additionally, the over-sampling technique is represented in the Balanced Classes group through three different keywords listed in the table: *”RandomOverSampler(”*, *”over_sampling(”* and *”SMOTE(”*. The Removing Duplicates group has only one example, which shows how the data is being removed of duplicates through the use of the keyword *”drop_duplicates(”*. The keyword group Removing Redundancies illustrates three keywords, which are commonly found within the notebooks, which are applied to remove single (*”remove_column(”*) or multiple columns (*”remove_columns(”*). The keyword *”drop(”* can be used to remove both columns and rows from a dataset. The Data Transformation group contains keywords, which are intended to change the formatting of an object to datetime (*”to_datetime(”*) or numeric (*”to_numeric(”*). Last but not least, the Unify Formatting group contains keywords, which are able to rename axes of labels (*”rename(”*) or capable of resetting (*”reset_index(”*) and setting indexes (*”set_index(”*).

Table 4.3: Table with a small sample of the keywords for each keyword group

Imputation	Outlier Detection	Dimensionality Reduction	Balanced Classes
”SimpleImputer(”, ”IterativeImputer(”, ”KNNImputer(”, ...	”LocalOutlierFactor(”, ”ABOD(”, ”EllipticEnvelope(”, ...	”PCA(”, ”IncrementalPCA(”, ”KernelPCA(”, ...	”RandomOverSampler(”, ”over_sampling”, ”SMOTE(”, ...
Removing Duplicates	Removing Redundancies	Data Transformation	Unify Formatting
”drop_duplicates”	”remove_column(”, ”remove_columns(”, ”drop(”	”to_datetime(”, ”to_numeric(”	”rename(”, ”reset_index(”, ”set_index(”, ...

The table 4.4 contains a sample of the data integration keywords for each keyword group. The merging group contains common keywords found in notebooks, which try to merge (*"merge("*), join (*"join("*) or concatenate (*"concatenate("*) datasets. Additionally, the Dataset Comparison group has keywords, such as *"intersection("*, which extracts the items present within both datasets, as well as the keyword *"diff("* that does the opposite.

Table 4.4: Table with a sample of the data integration keywords.

Merging	Dataset Comparison
"merge(", "join(", "concat(", ...	"intersection(", "diff("

4.3.2 Research Questions

RQ1: *How many data sets do data scientists analyze simultaneously?* In order to answer this question, every occurrence of data loading keyword in a notebook was counted, and stored in the variable files_imported. The variable was evaluated for precision, recall and accuracy. Furthermore, the variable was calculated for every notebook in all three annotated datasets. The files_imported variable was grouped by notebook type and renamed to: filesImportedSimple, filesImportedMining, and filesImportedPredictive. The mean and standard deviation was calculated for each of the three variables. Furthermore, a kernel density estimation was created for all three variables, and scaled logarithmically.

RQ2: *What are the least and most frequent data cleaning tasks?* All three annotated datasets were mined using the keywords derived from the data cleaning tasks. Each data cleaning task has its own set of keywords. The frequency of each keyword per data cleaning task was counted for every notebook. The amount of data cleaning tasks were then stored in the following variables: imputation_count, outliersDetection_count, balancedClasses_count, dimensionalityReduction_count, removingDuplicates_count, dataTransformation_count, removingRedundancies_count and unifyFormatting_count. Subsequently, all occurrences of each data cleaning task were added up per notebook type. For instance, all counts of imputation in the notebooks of the type simple descriptive were summed up, and stored in a variable called simple_imputation. The naming convention is the same for the other tasks and types of notebooks, e.g., mining_imputation and predictive_imputation. Following this, all variables were visualized in a bar plot.

Additionally, two groups of data cleaning task were identified for each of the three notebook types. These were formed in order to be able to make in-group comparisons within a notebook type, and provide more empirical evidence in this regard. The groups identified are explained below:

1. Clustering versus outlier detection algorithms in descriptive mining notebooks.
2. Regression versus classification algorithms in predictive mining notebooks.
3. Statistical hypothesis tests versus simple statistics (mean and standard deviation) in predictive mining notebooks.

RQ2a: *What are the least and most frequent data cleaning tasks in the outlier detection versus clustering group?* Two groups were recognized within descriptive mining notebooks, which allow for in-group comparisons across all data cleaning tasks. The groups are *Outlier Detection* and *Clustering.* Both groups consists of keywords, which relate to the function of the group. Thus, the group outlier detection consists of descriptive mining notebooks, which have the flag variable outlier_flag set to 1 (true). The flag variable is set to true if the keywords inside of the notebook coincide with the set of keywords of the outlier detection group. The keyword set is a narrower selection of the mining keywords shown in 4.3.1, but with a focus on outlier detection. The same is done for the clustering group through the flag variable clustering_flag. The keyword sets can be found in the appendix in A.3. Furthermore, the occurrences of the keywords for each group were counted and summed up, and displayed in a bar plot.

RQ2b: *What are the least and most frequent data cleaning tasks in the simple statistics versus statistical tests group?* In order to answer this research question, two groups were identified within the simple descriptive notebooks, which allow for in-group comparisons across all data cleaning tasks. The groups are *Simple Statistics* and *Statistical Tests*. Both groups consists of keywords, which relate to the function of the group. The simple statistics group consists of keywords, which apply statistics like the mean and standard deviation. The keywords for statistical tests are keywords, which apply hypothesis tests or similar. Furthermore, the group simple statistics consists of simple descriptive notebooks, which have the flag variable meanstd_flag set to 1 (true). The flag variable is set to true if the keywords inside of the notebook coincide with the set of keywords of the simple descriptive group. The keyword set is a narrower selection of the mining keywords shown in 4.3.1, but with a focus on simple statistics. The same is done for the statistical tests group with the flag variable hypothesis_flag . The keyword sets can be found in the appendix in A.3. Furthermore, the occurrences of the keywords for each group were counted and summed up, and displayed in a bar plot.

RQ2c: *What are the least and most frequent data cleaning tasks in the regression versus classification group?* This particular research question was answered by defining two groups within the predictive mining dataset, which allow for in-group comparisons across all data cleaning tasks. The groups are *Regression* and *Classification*. Both groups consists of keywords, which relate to the function of the group. Additionally, the regression group consists of predictive mining notebooks, which have the flag variable regression_flag set to 1 (true). The flag variable is set to true when the notebooks contain keywords from the regression keyword group. The keyword set is a narrower selection of the mining keywords shown in 4.3.1, but with a focus on regression algorithms. The same applies for the classification group with the flag variable classification_flag. The keyword sets can be found in the appendix under A.3. Moreover, the occurrences of the keywords for each group were counted and summed up, and displayed in a bar plot.

RQ3: *What are the least and most frequent data integration tasks?* All three annotated datasets were mined using the keywords from each data integration task. Each occurrence for every keyword was computed in every notebook. All instances across all notebooks of the same type were grouped by its notebook type, and stored in the following variables: simple_merging, simple_datasetComparison, mining_merging, mining_datasetComparison, predictive_merging and predictive_datasetComparison. Following this, all variables were visualized in a bar plot.

Additionally, the same six groups as in RQ2 were defined using the same keywords. The groups are: clustering, outlier detection, regression, classification, statistical hypothesis tests and simple statistics, such as the mean and standard deviation. These were defined in order to be able to make in-group comparisons within the notebook types for data integration. The same procedure was applied, the instances of the keywords were added up across all notebooks of the group for each notebook type, and shown in a bar plot.

RQ3a: *What are the least and most frequent data integration tasks in the outlier detection versus clustering group?* Same procedure as in RQ2a with the same flag variables: outlier_flag, and clustering_flag. The data integration tasks were grouped by outlier detection, and clustering keywords. The counts of each data integration task within each group was then displayed in a bar plot.

RQ3b: *What are the least and most frequent data integration tasks in the simple statistics versus statistical tests group?* The same procedure applies as in RQ2b by using the same flag variables: meanstd_flag, and hypothesis_flag. The occurrences of the data integration tasks were grouped by simple statistics and statistical tests. In a second step, they were summed up according to the data integration task. The results were then shown in a bar plot.

RQ3c: *What are the least and most frequent data integration tasks in the regression versus classification group?* This research question was answered by appling the same method as in RQ2c. The same flag variables were used: regression_flag, and classification_flag. The occurrences of the data integration tasks within the predictive mining dataset were grouped according to the function of the classification and regression group. In a second step, they were summed up according to the respective data integration task. The results were then shown in a bar plot.

RQ4a: *Is the mean length of predictive notebooks equal to the mean length of simple and descriptive mining notebooks?* Each notebook has a variable called total_cells that contains the total number of cells inside the notebook. The total cells for each notebook were grouped by their notebook type and stored in the variables: length_simple, length_mining, and length_predictive. Subsequently, the distribution of each of these variables were tested for normality with the Shapiro-Wilk test (Shapiro and Wilk, 1965). The equality of variances was also assessed using Levene's test on all three distributions as defined by Levene (1960). Furthermore, the mean and standard deviation of the three distribution were calculated, and a kernel density estimation was plotted for each of the distributions with a logarithmic scale. The hypotheses were defined as the following:

Hypothesis 1. *The mean length of predictive notebooks is unequal to the mean length of simple descriptive or descriptive mining notebooks.*

Null Hypothesis 1. *The mean length of predictive notebooks is equal to the mean length of simple descriptive or descriptive mining notebooks.*

These hypotheses were tested using the Kruskal-Wallis test (Kruskal and Wallis, 1952). This was performed on the three samples in order to test if the samples originate from the same distribution. Furthermore, this method was applied instead of ANOVA, because all three distributions are non-parametric. Moreover, the Kruskal-Wallis test was rejected, and thus a post-hoc Dunn test was performed (Dinno, 2015).

RQ4b: *Is the mean of data cleaning tasks in predictive notebooks equal to the mean in simple descriptive and descriptive mining notebooks?* The procedure is very similar to the process in RQ4a. In order to answer this research question, all cleaning tasks were totaled by notebook type. This was done by using the variables: imputation_count, outliersDetection_count, balancedClasses_count, dimensionalityReduction_count, removingDuplicates_count, dataTransformation_count, removingRedundancies_count and unifyFormatting_count. The result are three distributions, which are stored in the variables: cleaning_tasks_simple_total, cleaning_tasks_mining_total and cleaning_tasks_predictive_total. This was followed by testing the distribution for normality with the Shapiro-Wilk test (Shapiro and Wilk, 1965). The equality of variances was also assessed using Levene's test on all three distributions as defined by Levene (1960). Furthermore, the mean and standard deviation of the three distributions were calculated, and a kernel density estimation was plotted for each of the distributions with a logarithmic scale. The hypotheses were defined as the following:

Hypothesis 2. *The mean of data cleaning tasks in predictive notebooks is unequal to the mean of data cleaning tasks in simple descriptive or descriptive mining notebooks.*

Null Hypothesis 2. *The mean of data cleaning tasks in predictive notebooks is equal to the mean of simple descriptive or descriptive mining notebooks.*

These hypotheses were tested with the Kruskal-Wallis test (Kruskal and Wallis, 1952). Furthermore, a Kruskal-Wallis test was performed on the three samples in order to test if the samples originate from the same distribution, because the samples were non-parametric (Kruskal and Wallis, 1952). Consequently, the Kruskal-Wallis test was rejected, and thus a post-hoc Dunn test was performed (Dinno, 2015).

RQ4c: *Is the mean of data integration tasks in predictive notebooks equal to the mean in simple descriptive and descriptive mining notebooks?* In order to answer this research question, all data integration tasks were totaled by notebook type by using the variables: merging_count and datasetComparison_count. The result are three distributions, which are stored in the following variables: integration_tasks_simple_total, integration_tasks_mining_total and integration_tasks_predictive_total. The three distributions were tested both for normality with the Shapiro and Wilk (1965) test and for equality of variances by using Levene's (1960) test. Furthermore, the mean and standard deviation of the three distributions were calculated, and a kernel density estimation was plotted for each of the distributions with a logarithmic scale. The hypotheses were defined as the following:

Hypothesis 3. *The mean of data integration tasks in predictive notebooks is unequal to the mean of data integration tasks in simple descriptive or descriptive mining notebooks.*

Null Hypothesis 3. *The mean of data integration tasks in predictive notebooks is equal to the mean of data integration tasks in simple descriptive or descriptive mining notebooks.*

Subsequently, a Kruskal-Wallis test was performed on the three samples in order to test if the samples originate from the same distribution (Kruskal and Wallis, 1952). Consequently, the Kruskal-Wallis test was rejected, and thus a post-hoc Dunn test was performed (Dinno, 2015).

RQ4d: *Is the mean of all data cleaning cells across all notebook types equal to the mean of all data integration cells across the three notebook types?* The answer to this question will make use of the distributions defined in RQ4c and RQ4d. The three variables cleaning_tasks_simple_total, cleaning_tasks_mining_total and cleaning_tasks_predictive_total were added together to form a new distribution called cleaning_tasks_total, which now contains all data cleaning tasks across all notebook types. The same is done for the three variables: integration_tasks_simple_total, integration_tasks_mining_total and integration_tasks_predictive_total. The end result is a distribution, which is contained in the variable integration_tasks_total. Both distributions are tested for normality with Shapiro and Wilk's (1965) test and for equality of variances by using Levene's (1960) test. Furthermore, the mean and standard deviation of the two distributions were calculated, and a kernel density estimation was plotted for each of the distributions with a logarithmic scale. The hypotheses were defined as the following:

Hypothesis 4. *The mean of data cleaning tasks is unequal to the mean of data cleaning tasks across all three notebook types.*

Null Hypothesis 4. *The mean of data cleaning tasks is equal to the mean of data cleaning tasks across all three notebook types.*

Subsequently, Kruskal and Wallis test was performed on the two samples in order to test if the samples originate from the same distribution, since the samples were non-parametric (Kruskal and Wallis, 1952). The Kruskal-Wallis test was rejected, and thus a post-hoc Dunn test was performed (Dinno, 2015).

RQ5: *What is the percentage of code dedicated to data cleaning in different types of data science notebooks?* This research question was answered by calculating the percentage of data cleaning in all notebooks and saving the results in the variable cleaning_percentage, which was defined as the total amount of data cleaning cells over the total amount of cells. The cleaning percentage was then grouped by notebook type in the variables: simple_cleaning_percentage, mining_cleaning_percentage, and predictive_cleaning_percentage. Subsequently, the mean and standard deviation of all three distributions were calculated, and the distributions were plotted in a kernel density estimation with a logarithmic scale.

RQ6: *What is the percentage of code dedicated to data integration in different types of data science notebooks?* This research question was answered in a similar fashion to RQ5. Each notebook contains a variable calculating the percentage of data integration in each notebook, which was defined as the total amount of data integration cells over the total amount of cells. The integration percentage was then grouped by notebook type inside the variables: simple_integration_percentage, mining_integration_percentage, and predictive_integration_percentage. Subsequently, the mean and standard deviation of all three distributions were calculated, and the distributions were plotted in a kernel density estimation with a logarithmic scale.

RQ7a: *Does data cleaning happen iteratively?* The cells inside of a Jupyter notebook are ordered, which facilitates the identification of the placement of a specific cell. Therefore, the placement of each data cleaning cell within a notebook was captured inside of the following variables: imputation_steps, outliersDetection_steps, balancedClasses_steps, removingDuplicates_steps, dataTransformation_steps, removingRedundancies_steps, and unifyFormatting_steps. In a second step, these variables were grouped by notebook type, and all data cleaning tasks were summed up together. The end result are the variables simple_cleaning_steps, mining_cleaning_steps and predictive_cleaning_steps, which contain the placement of every data cleaning task inside of a notebook. Furthermore, a notebook applies data cleaning iteratively when two data cleaning cells are at least more than one cell apart. Subsequently, this iteration has been calculated through an algorithm, which calculates the distance between two data cleaning cells inside of a notebook. This has been performed for all three variables.

RQ7b: *What is the ratio between data cleaning and predictive cells?* Only the predictive mining dataset was used to answer this question, since only this dataset contains predictive mining cells. Furthermore, this research question was answered by dividing the total amount of data cleaning cells present in the predictive mining notebooks by the total amount of predictive cells in the notebooks. The corresponding variables, which contain the total amount of data cleaning and predictive mining cells are: cleaning_cells_predictive_sum and predicive_tasks_sum.

RQ7c: *What is the ratio between data cleaning and descriptive mining cells?* The only dataset considered for this research question was the descriptive mining dataset, because it is the only one that contains descriptive mining cells. Subsequently, this research question was answered by dividing the total amount of data cleaning cells present in the descriptive mining notebooks by the total amount of descriptive mining cells in the notebooks. The corresponding variables, which contain the total amount of data cleaning and descriptive mining cells are: cleaning_cells_mining_sum and mining_tasks_sum.

R7d: *What is the ratio between data cleaning and simple descriptive cells?* Only the simple descriptive dataset was used to answer this question, because it is the sole dataset with simple descriptive cells. The ratio was calculated by dividing the total amount of data cleaning cells present in the simple descriptive notebooks by the total amount of simple descriptive cells. The corresponding variables, which contain the total amount of data cleaning and simple descriptive mining cells are: cleaning_cells_simple_sum and simple_tasks_sum.

RQ7e: *What is the ratio between data cleaning and data visualization cells?* All three datasets were used to answer this question, since they all contain both data cleaning and data visualization cells. The ratio was calculated by dividing the total amount of data cleaning cells present in the notebooks by the total amount of data visualization cells across all notebook types. The corresponding variables, which contain the total amount of data cleaning and data visualization cells are: sum_total_cleaning_cells and sum_total_visualization_cells.

RQ8a: *Does data integration happen iteratively?* Due to the cells in a jupyter notebook being ordered, it is possible to identify the position of every data integration cell within a notebook. Therefore, the placement of each data integration cell within a notebook was captured inside of the following variables: merging_steps, datasetComparison_steps. In a second step, the variables were grouped by notebook type, and all data integration tasks were summed up together. The end result is summarized in the variables simple_integration_steps, mining_integration_steps and predictive_integration_step. Moreover, a notebook applies data integration iteratively when two data integration cells are at least more than one cell apart. Consequently, a piece of code was devised, which calculates the distance between two data integration cells inside of a notebook. This has been performed for all variables.

RQ8b: *What is the ratio between data integration and predictive cells?* Solely the predictive mining dataset was used to answer this question. This is because it is the only dataset, which contains predictive mining cells. Additionally, the ratio was calculated by dividing the total amount of data integration cells present in the predictive mining notebooks by the total amount of predictive cells in the notebooks. The corresponding variables, which contain the total amount of data integration and predictive mining cells are: integration_cells_predictive and predictive_tasks_sum.

RQ8c: *What is the ratio between data integration and descriptive mining cells?* The descriptive mining dataset was taken exclusively into consideration for this particular research question. The ratio was calculated by dividing the total amount of data integration cells present in the descriptive mining notebooks by the total amount of descriptive mining cells in the same notebooks. The corresponding variables, which contain the total amount of data integration and descriptive mining cells are: integration_cells_mining and mining_tasks_sum.

RQ8d: *What is the ratio between data integration and simple descriptive cells?* This research question only considers the simple descriptive dataset. The ratio between the data integration and simple descriptive cells was calculated by dividing the total amount of data integration cells by the total amount of simple descriptive cells present across the simple descriptive notebooks. The variables used to calculate this particular ratio were: integration_cells_simple and simple_tasks_sum.

RQ8e: *What is the ratio between data integration and data visualization cells?* In contrast to RQ8a until RQ8d, this research question takes all datasets into consideration in order to calculate the ratio between the data integration and data visualization cells. The ratio itself was calculated by dividing the total amount of data integration cells by the total amount of data visualization cells prevalent in the notebooks across all notebook types. The variables used to calculate the ratio were the following: total_integration_cells and total_visualization_cells.

5

Empirical Results

This chapter discusses the results for each research question, which were obtained after labelling and applying statistical methods.

5.1 Exploratory Data Analysis

This section discusses the results of the exploratory data analysis done on the Github dataset, as well as on the three annotated datasets in order to understand the characteristics of the notebooks.

5.1.1 Programming Languages

The GitHub dataset has notebooks with different programming languages, unlike the three annotated datasets. Table 5.1 shows the distribution of the programming languages within the entire GitHub dataset. It is apparent from the table that Python is the most common programming language followed by Julia, R and Scala.

Table 5.1: Distribution of programming language in the GitHub dataset

Programming Language	Number of Notebooks
Python	786544
Julia	11174
R	9394
Scala	1094
Ruby	915
Matlab	576
Haskell	511
Javascript	462
C++	263
Lisp	67
Clojure	25
Stata	12
Elm	11
Elixir	10

5.1.2 Total Cells

The total number of cells were calculated for each notebook and grouped by the type of dataset. The results are four distributions containing the total number of cells per dataset type, whose mean and standard deviation are shown in table 5.2. The predictive mining dataset has the highest mean for the number of total cells per notebook, and a standard deviation of 70.07. The Github dataset, however, has the smallest mean with 27.93 total cells per notebook, and a standard deviation of 32.15.

Table 5.2: Mean and standard deviation of the distributions of the total number of cells per notebook

Dataset	Mean	Standard Deviation
Simple Descriptive	62.69	58.43
Descriptive Mining	70.32	72.21
Predictive Mining	74.36	70.07
Github	27.93	32.15

Figure 5.1 shows the four distributions in a kernel density estimation plot. The x-axis shows the number of total cells per notebook scaled logarithmically, and the y-axis displays the density.

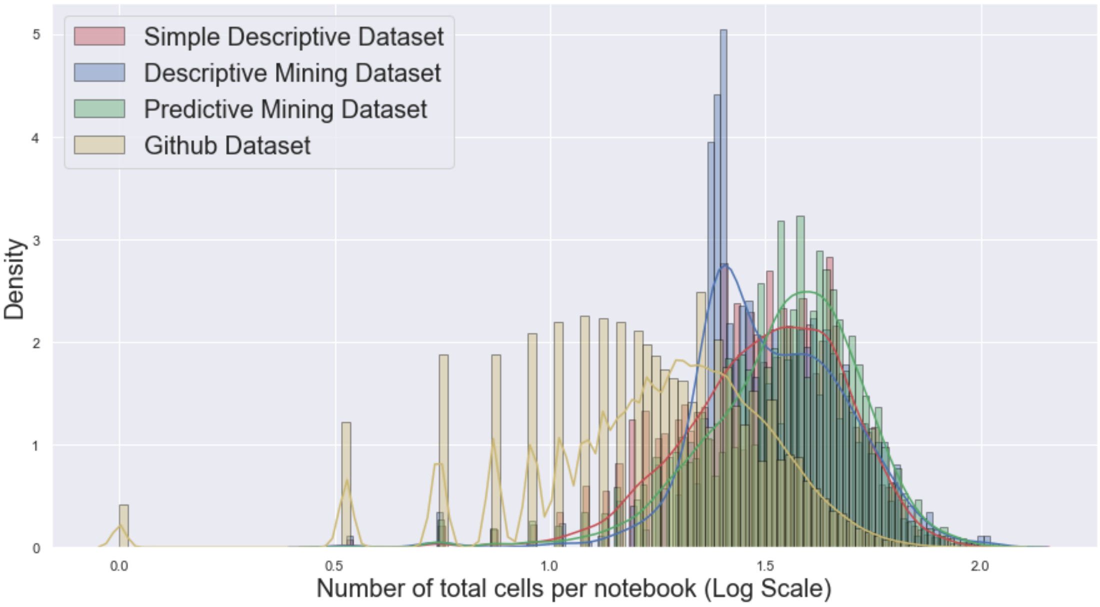


Figure 5.1: Log-scaled distribution of the number of total cells per notebook for each dataset

5.1.3 Code Cells

The total number of code cells were calculated for each notebook inside the four datasets. The results are four distributions, which contain the total number of code cells per notebook for each of the four datasets. The mean and standard deviation of each distribution are shown in table 5.3. The predictive mining dataset has the highest mean with a value of 57.70. The GitHub dataset has the lowest mean of all four distributions with 18.84 total number of cells per notebook.

Table 5.3: Mean and standard deviation of the distributions of the number of code cells per notebook

Dataset	Mean	Standard Deviation
Simple Descriptive	47.39	44.98
Descriptive Mining	51.05	60.14
Predictive Mining	57.70	56.80
Github	18.84	22.58

Figure 5.4 shows the four distributions in a kernel density estimation plot. The x-axis displays the number of code cells per notebook scaled logarithmically, while the y-axis shows the density.

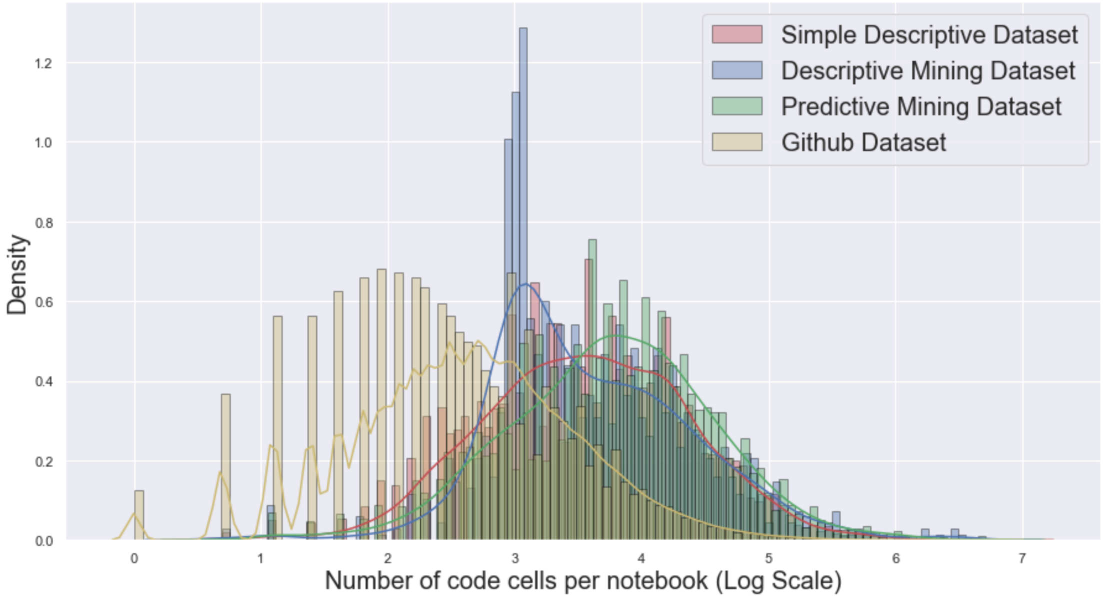


Figure 5.2: Log-scaled distribution of the number of code cells per notebook for each dataset

5.1.4 Markdown Cells

The total number of markdown cells were calculated for each notebook inside of each dataset. The target variable was then grouped for each dataset, creating four distributions of the total number of markdown cells. The mean and standard deviation of each of the four distributions are shown in table 5.4. The descriptive mining dataset has the highest mean with a value of 18.79. The GitHub dataset has the lowest mean of all four distributions with 8.71 markdown cells per notebook.

Table 5.4: Mean and standard deviation of the distributions of the number of markdown cells per notebook

Dataset	Mean	Standard Deviation
Simple Descriptive	14.71	22.22
Descriptive Mining	18.79	21.97
Predictive Mining	16.05	21.69
Github	8.71	15.96

Figure 5.3 shows the four distributions in a kernel density estimation plot. The x-axis displays the number of markdown cells per notebook scaled logarithmically, and the y-axis shows the density.

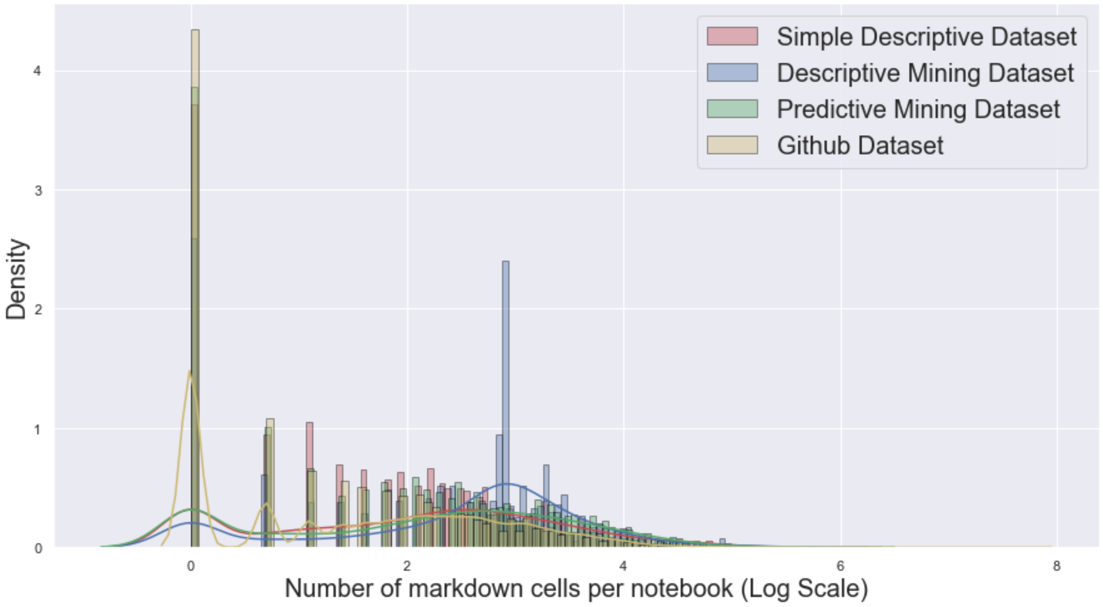


Figure 5.3: Log-scaled distribution of the number of markdown cells per notebook for each dataset

5.1.5 Functions

Four distributions were created by extracting the number of functions for each notebook. One distribution for each of the four datasets. The mean and standard deviation of each of the four distributions are shown in table 5.5. The descriptive mining dataset has the highest mean with a value of 8.98. The GitHub dataset has the lowest mean of all four distributions with 3.48 functions per notebook.

Table 5.5: Mean and standard deviation of the distributions of the number of functions per notebook

Dataset	Mean	Standard Deviation
Simple Descriptive	7.88	8.87
Descriptive Mining	8.98	8.24
Predictive Mining	8.84	10.82
Github	3.48	9.26

Figure 5.4 shows the four distributions in a kernel density estimation plot. The x-axis displays the number of functions per notebook scaled logarithmically, and the y-axis shows the density.

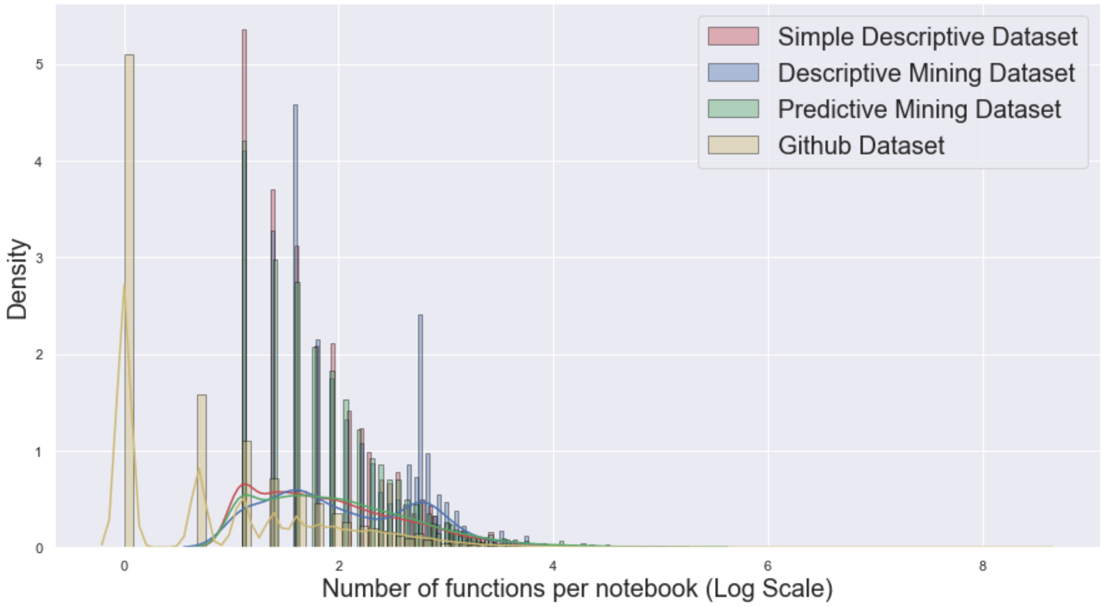


Figure 5.4: Log-scaled distribution of the number of functions per notebook for each dataset

5.1.6 Variables

The number of variables inside of each notebook were calculated for each of the four datasets. The result of this are four distributions containing the number of variables per notebook for each dataset. The mean and standard deviation of each of the four distributions are shown in table 5.6. The predictive mining dataset has the highest mean with a value of 126.95 variables per notebook. The GitHub dataset has the lowest mean of all four distributions with 31.13 variables per notebook.

Table 5.6: Mean and standard deviation of the distributions of the number of variables per notebook

Dataset	Mean	Standard Deviation
Simple Descriptive	89.51	92.54
Descriptive Mining	91.37	86.41
Predictive Mining	126.95	116.22
Github	31.13	54.16

Figure 5.5 displays the four distributions in a kernel density estimation. The x-axis displays the number of variables per notebook scaled logarithmically, and the y-axis shows the density.

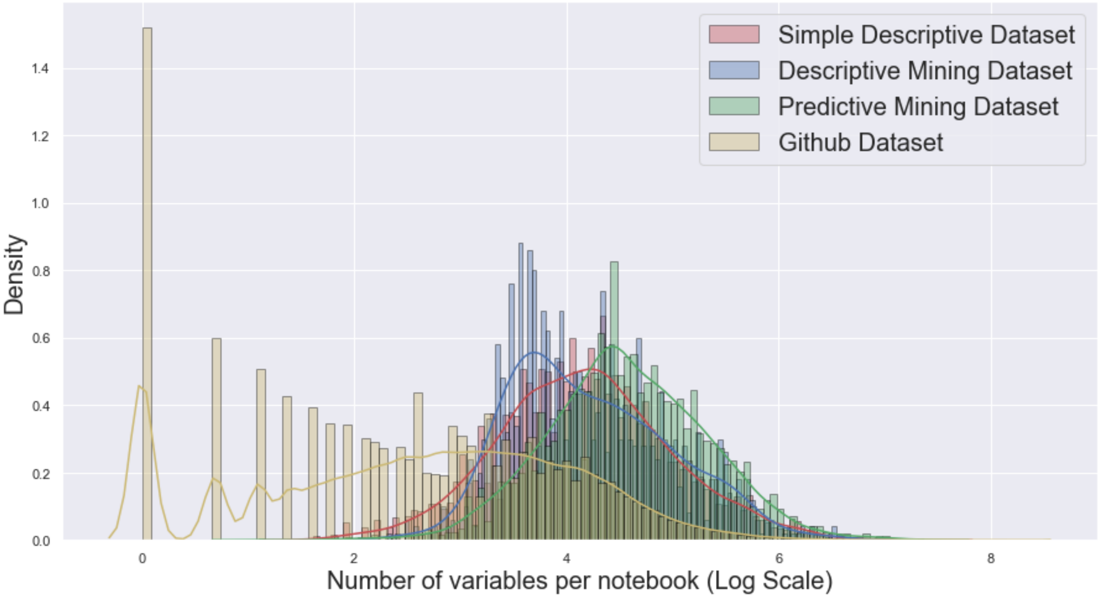


Figure 5.5: Log-scaled distribution of the number of variables per notebook for each dataset

5.2 Answers to the research questions

5.2.1 How many datasets do data scientists analyze simultaneously?

The number of datasets imported were calculated for each notebook and grouped by notebook type. The results are three distributions, whose mean and standard deviation are shown in Table 5.7. Predictive mining notebooks have the highest mean, and standard deviation of all three types with an average of 6.9 imported files per notebook and a standard deviation of 13.05.

Table 5.7: Table containing mean and standard deviation of the distribution of imported datasets

	Simple Descriptive	Mining Descriptive	Predictive Mining
Mean	6.55	5.70	6.90
Standard Deviation	8.22	6.06	13.05

Three kernel density estimations (KDE) were created in order to display the three distributions. All kernel density estimations have the number of imported datasets per notebook in logarithmic scale in the x-axis, and the corresponding density in the y-axis. Figure 5.6 shows the kernel density estimation for notebooks of type simple descriptive, while figure 5.7 does the same for descriptive mining and figure 5.8 for predictive mining notebooks. Furthermore, the results of the three distributions indicate that the majority of the notebooks import a small number of datasets, while only a minority of the notebooks import a large quantity of datasets. This is true for all three distributions. Moreover, the unusual high number of datasets imported in a notebook can be attributed in some cases to the measurement method. For instance, the keyword *"open("* can indicate that there is a file being imported, but there are some cases in which it appears in conjunction with the keyword *"load("*. This would count as two files being imported, although it is only effectively one file. However, none of the keywords can be removed from the set, because it would increase the amount of false negatives. Furthermore, having both keywords in the set is still a good compromise with regards to the precision, recall and accuracy of the entire keyword set.

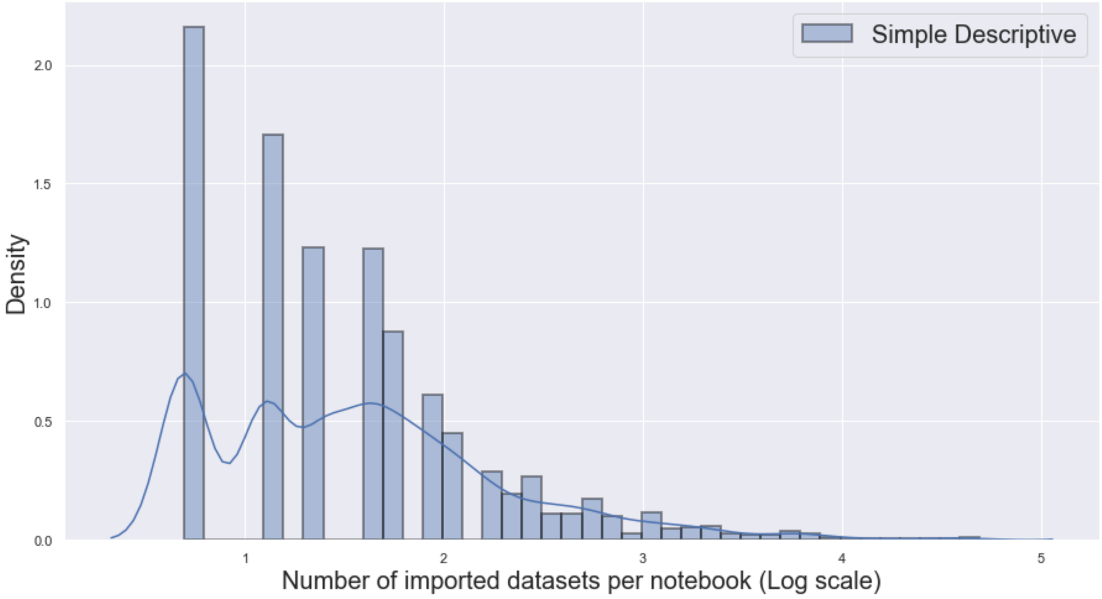


Figure 5.6: Log-scaled distribution of the number of datasets imported per simple descriptive notebook

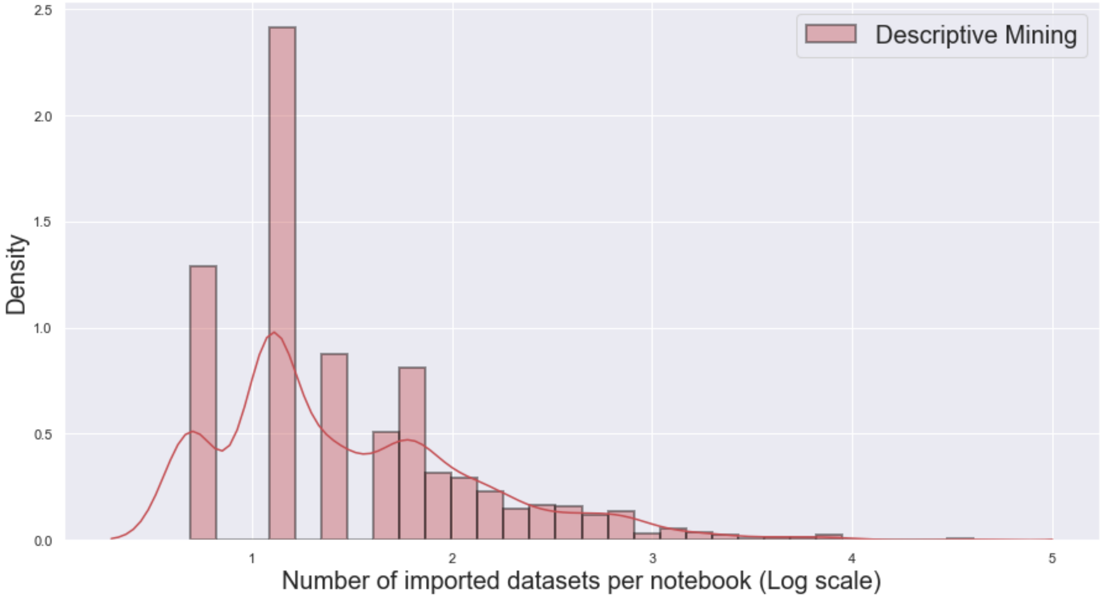


Figure 5.7: Log-scaled distribution of the number of datasets imported per descriptive mining notebook

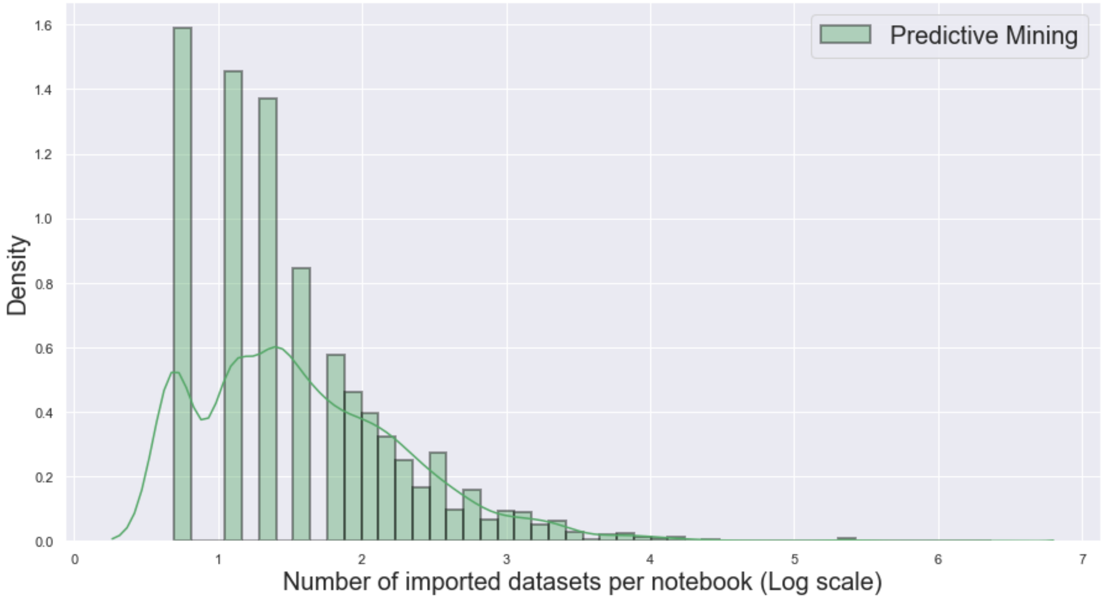


Figure 5.8: Log-scaled distribution of the number of datasets imported per predictive mining notebook

5.2.2 What are the least and most frequent data cleaning tasks?

Figure 5.9 displays a bar plot with the different data cleaning tasks in the x-axis and the frequency in the y-axis. The least and most frequent data cleaning task are shown for all notebook types. The most common data cleaning task for all three notebook types is *Unify Formatting*. The second most common data cleaning tasks varies between the three types. While *Imputation* is the second biggest data cleaning task for both simple descriptive and descriptive mining notebooks, it is the task *Removing Redundancies* for predicitve mining notebooks. Furthermore, the number of imputation tasks is greater in predictive mining notebooks in comparison to simple descriptive notebooks. The least frequent task is *Outlier Detection* for all three notebook types.

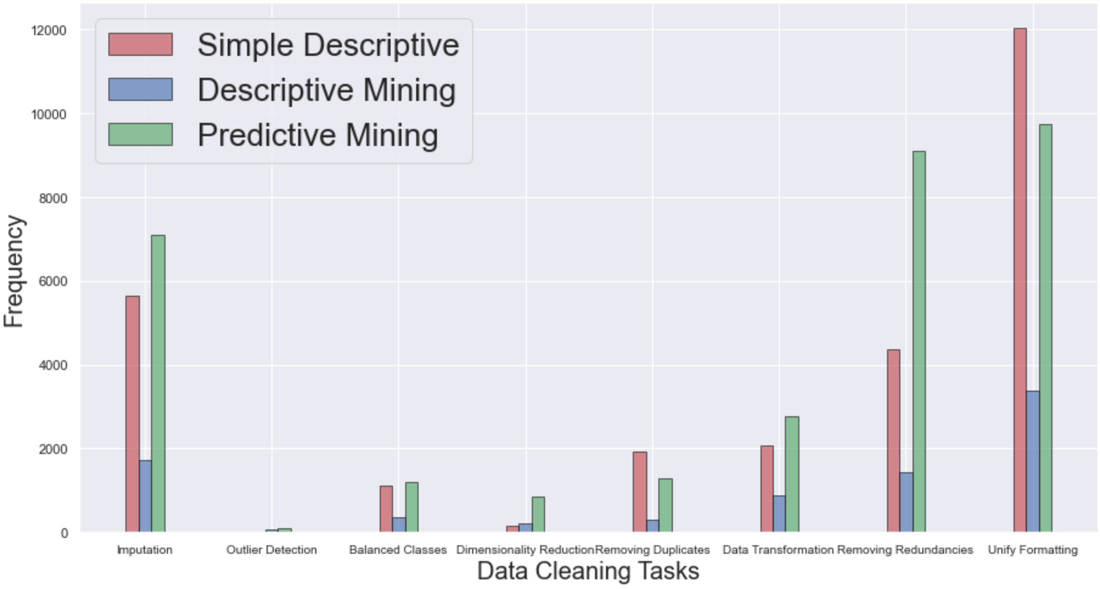


Figure 5.9: Frequency of data cleaning tasks by notebook type

Furthermore, the table 5.8 displays the percentage of each data cleaning task over the total cells for each notebook type. The most and least frequent data cleaning tasks coincide with the results shown in figure 5.9. The highest percentage is for the data cleaning task *Unify Formatting* with 3.98% for simple descriptive, 4.15% for descriptive mining, and 3.06% for predictive mining notebooks. The task with the lowest percentage across all notebook cells per notebook type is *Outlier Detection* with 0.0% for simple descriptive, 0.07% for descriptive mining and 0.03% for predictive mining notebooks.

Table 5.8: Percentage of the respective data cleaning task over the total cells for each notebook type

	Simple Descriptive	Descriptive Mining	Predictive Mining
Imputation	1.86%	2.10%	2.22%
Outlier Detection	0.00%	0.07%	0.03%
Balanced Classes	0.36%	0.42%	0.38%
Dimensionality Reduction	0.04%	0.27%	0.27%
Removing Duplicates	0.63%	0.35%	0.40%
Data Transformation	0.68%	1.06%	0.87%
Removing Redundancies	1.44%	1.75%	2.85%
Unfiy Formatting	3.98%	4.15%	3.06%

5.2.3 What are the least and most frequent data cleaning tasks in the outlier detection versus clustering group?

Figure 5.10 is a barplot, which shows the data cleaning tasks of the outlier detection and clustering group with the different types of data cleaning tasks in the x-axis, and the frequency in the y-axis. The outlier group has a total of six data cleaning occurrences, while the clustering group has a total of 1901. This could be attributed to the low amount of notebooks with outlier detection algorithms. Furthermore, the clustering group has all types of data cleaning tasks present with *Unify Formatting* being the most prominent task of all.

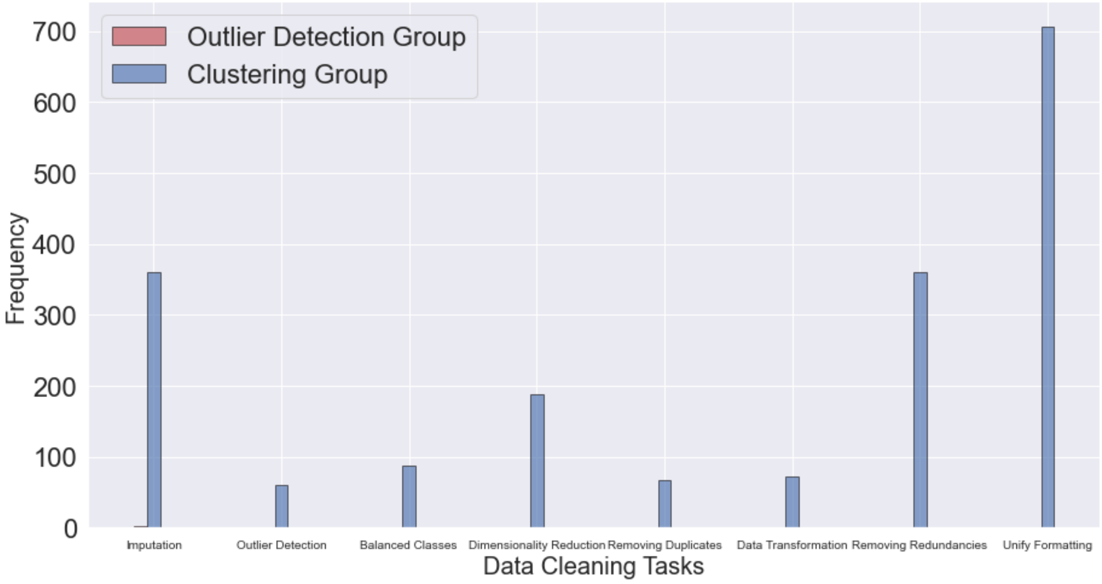


Figure 5.10: Frequency of data cleaning tasks by notebook group (outlier detection versus clustering)

5.2.4 What are the least and most frequent data cleaning tasks in the simple statistics versus statistical tests group?

Figure 5.11 is a barplot, which shows the data cleaning tasks of the simple statistics group versus the statistical tests group with the different types of data cleaning tasks in the x-axis, and the frequency in the y-axis. The simple statistics group has a total of 17343 data cleaning occurrences across all data cleaning tasks, while the statistical tests groups has only 2232. The figure also demonstrates that the task with the highest amount of occurrences is *Unify Formatting* for both groups. The least frequent data cleaning task is *Outlier Detection* for both groups.

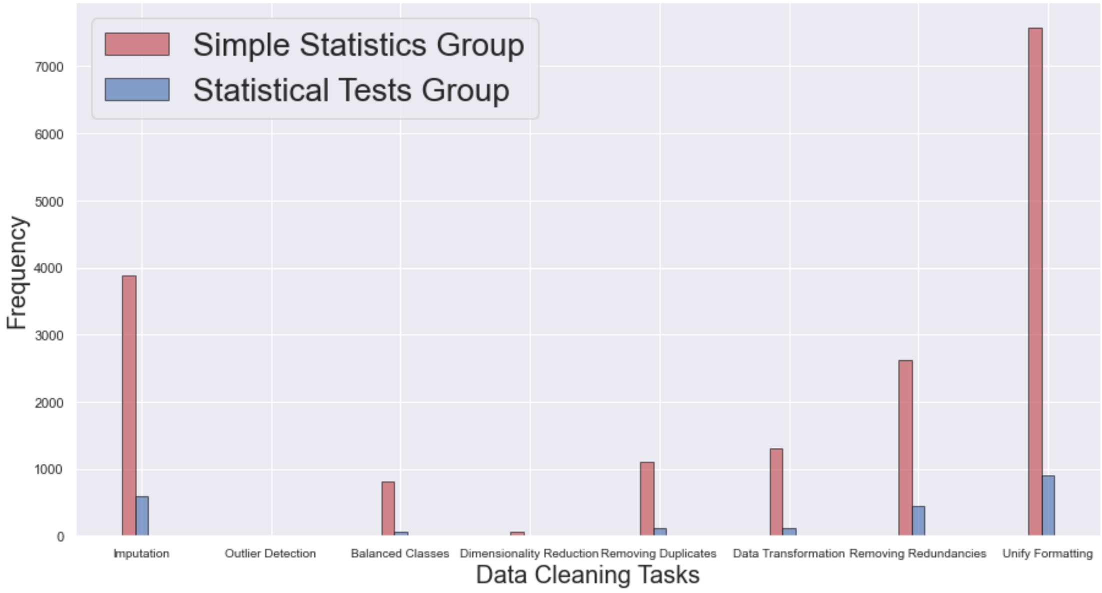


Figure 5.11: Frequency of data cleaning tasks by notebook group (simple statistics versus statistical tests)

5.2.5 What are the least and most frequent data cleaning tasks in the regression versus classification group?

Figure 5.12 is a barplot, which shows the data cleaning tasks of the regression group versus the classification group with the different types of data cleaning tasks in the x-axis, and the frequency in the y-axis. The regression group has a total of 21370 data cleaning occurrences across all data cleaning tasks, while the classification group has 13277. The data cleaning task with the highest occurrences is *Unify Formatting* for the regression group, and *Removing Redundancies* for the classification group. The least frequent data cleaning task is *Outlier Detection* for both groups.

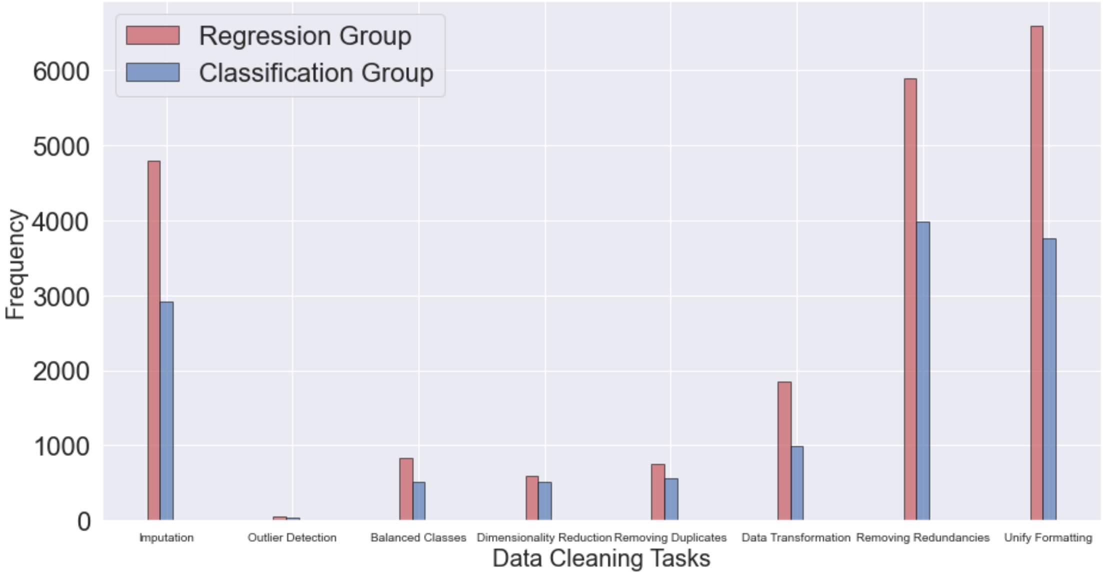


Figure 5.12: Frequency of data cleaning tasks by notebook group (regression versus classification)

5.2.6 What are the least and most frequent data integration tasks?

Figure 5.13 displays a bar plot with the different data integration tasks in the x-axis and the frequency in the y-axis. The least and most frequent data integration task are shown for all notebook types. The figure clearly shows that there are significantly more *Merging* tasks then *Dataset Comparison* tasks for all three notebook types.

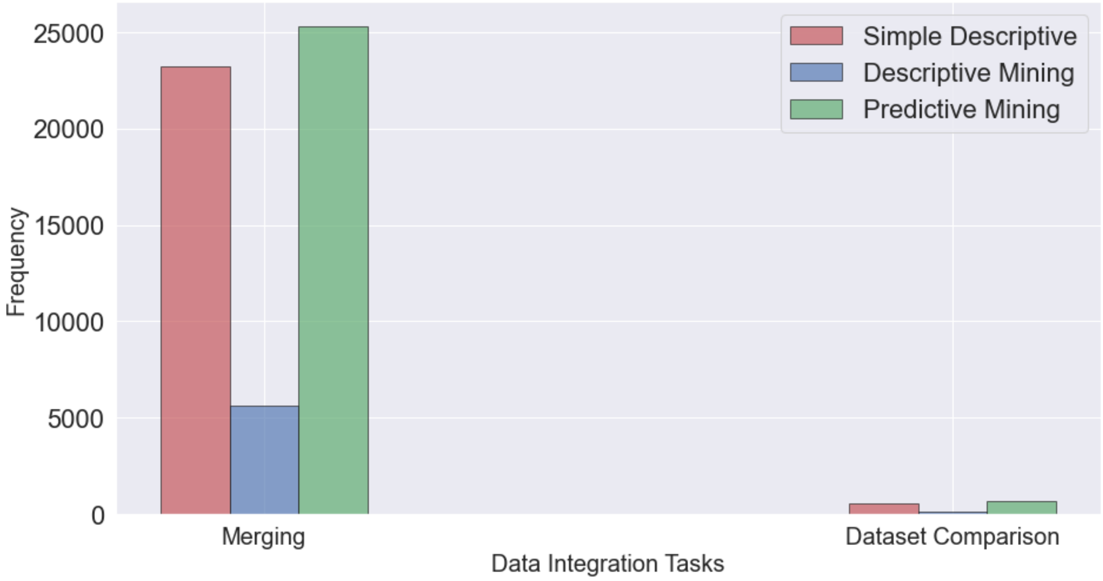


Figure 5.13: Frequency of data integration tasks by notebook type

Furthermore, the table 5.9 displays the percentage of each data integration task over the total number of cells for each notebook type. The most and least frequent data integration tasks coincide with the results shown in figure 5.13. The highest percentage is for the data integration task *Merging* with 7.69% for simple descriptive, 6.90% for descriptive mining, and 7.90% for predictive mining notebooks. The task with the lowest percentage across all notebook cells per notebook type is *Dataset Comparison* with 0.19% for simple descriptive, 0.18% for descriptive mining and 0.21% for predictive mining notebooks.

Table 5.9: Percentage of the respective integration task over the total cells per notebook

	Simple Descriptive	Descriptive Mining	Predictive Mining
Merging	7.69%	6.90%	7.90%
Dataset Comparison	0.19%	0.18%	0.21%

5.2.7 What are the least and most frequent data integration tasks in the outlier detection versus clustering group?

Figure 5.14 displays a bar plot for the outlier detection and the clustering group with the different data integration tasks in the x-axis and the frequency in the y-axis. The outlier group has a total of four data integration occurrences, while the clustering group has a total of 1909 instances across all notebook types. This could be attributed to the low amount of notebooks with outlier detection algorithms. The figure also shows that there are significantly more *Merging* instances than instances of *Dataset Comparison*.

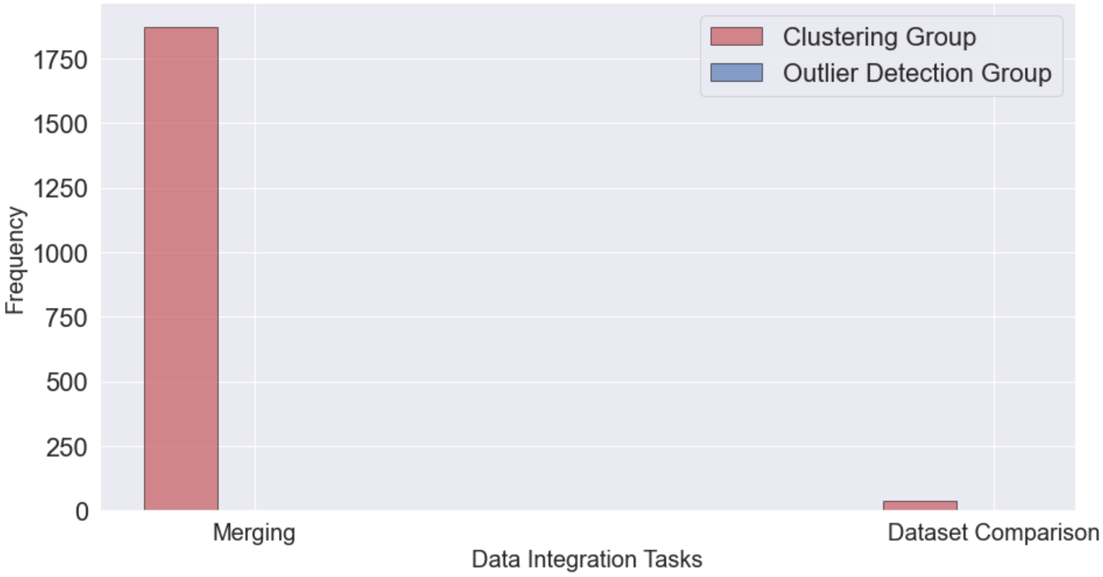


Figure 5.14: Frequency of data integration tasks by notebook group (outlier detection versus clustering)

5.2.8 What are the least and most frequent data integration tasks in the simple statistics versus statistical tests group?

The figure 5.15 shows a bar plot for the simple statistics and the statistical test groups with the different data integration tasks in the x-axis and the frequency in the y-axis. The simple statistics group has a total of 14960 data integration occurrences, while the statistical tests group has a total of 1803 instances across all notebook types. Figure 5.15 also shows that the data integration task with the highest occurrences is *Merging* for both groups.

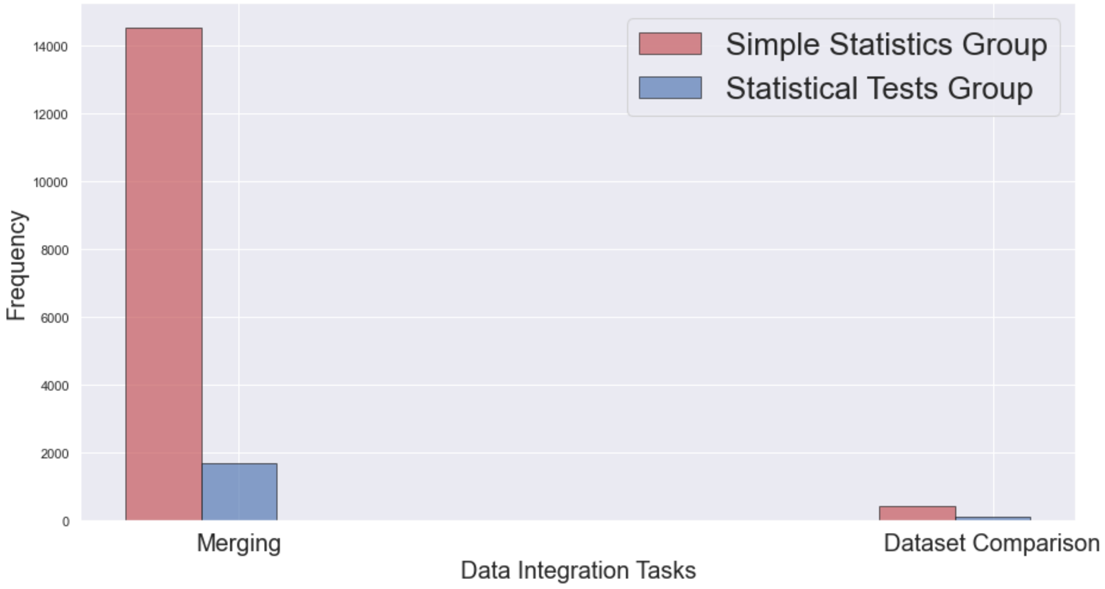


Figure 5.15: Frequency of data integration tasks by notebook group (simple statistics versus statistical tests)

5.2.9 What are the least and most frequent data integration tasks in the regression versus classification group?

The figure 5.16 shows a bar plot for the regression and the classification groups with the different data integration tasks in the x-axis and the frequency in the y-axis. The classification group has a total of 10803 data integration occurrences, while the regression group has a total of 16613 instances across all notebook types. The data integration task with the highest occurrences is *Merging* for both groups.

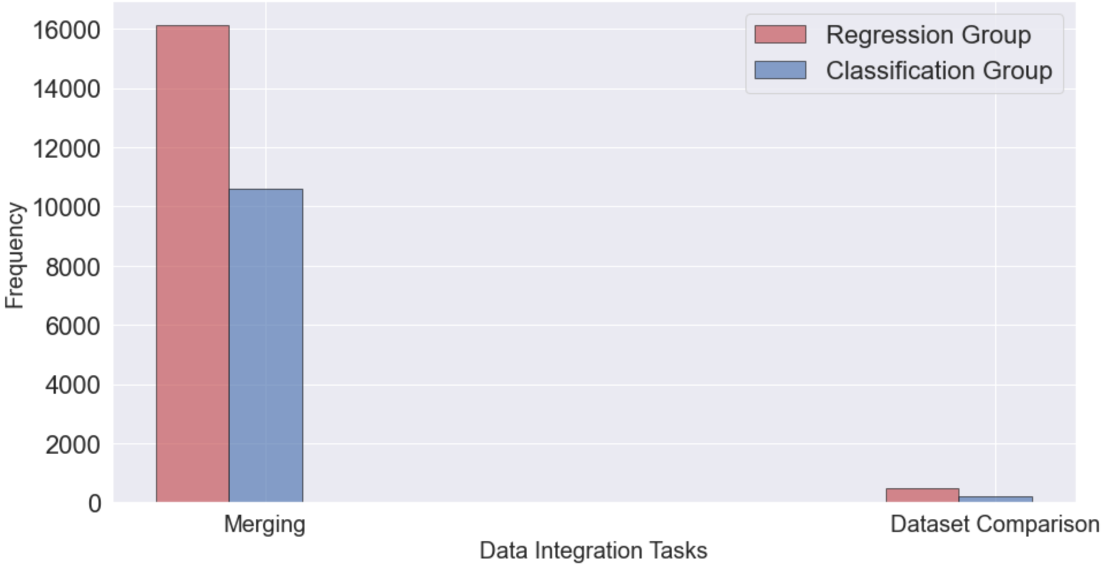


Figure 5.16: Frequency of data integration tasks by notebook group (regression versus classification)

5.2.10 Is the mean length of predictive notebooks equal to the mean length of simple and descriptive mining notebooks?

Figure 5.17 shows the distribution of the total cells for each notebook for all three notebook types in a kernel density estimation. The x-axis displays the total number of cells per notebook in a logarithmic scale, and the y-axis shows the density. Table 5.10 displays the mean and standard deviation of the total notebook cells distribution for each notebook type. The predictive mining dataset has the highest mean with 73.29, followed by descriptive mining with a mean of 69.81, and simple descriptive with 62.73.

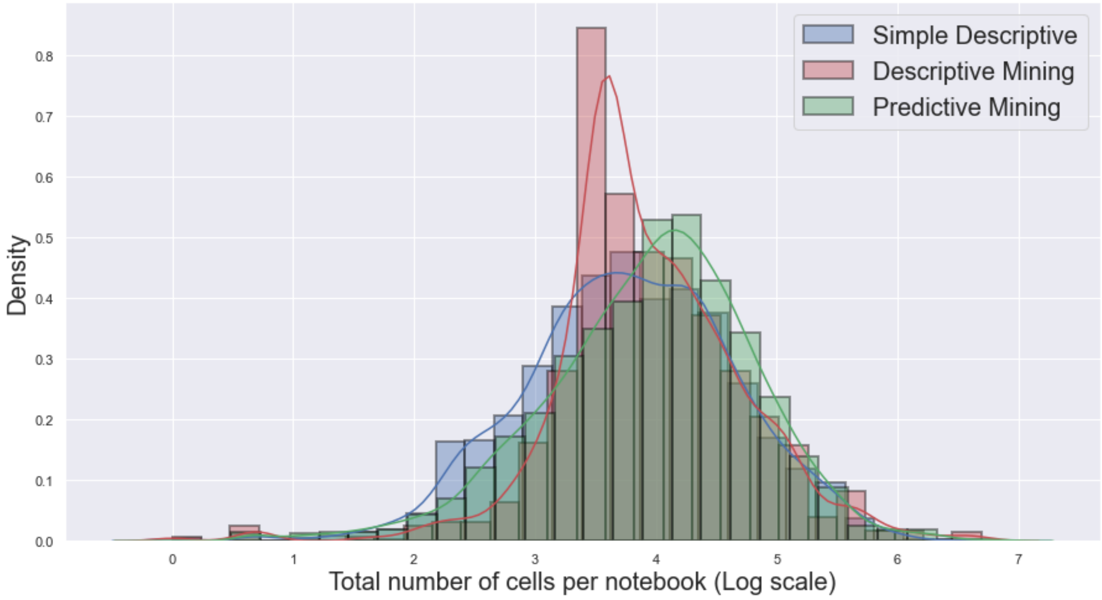


Figure 5.17: Log-scaled distribution of total cells per notebook type

Table 5.10: Mean and standard deviation of the notebook length distribution for each notebook type

	Simple Descriptive	Mining Descriptive	Predictive Mining
Mean	62.73	69.81	73.29
Standard Deviation	59.16	72.66	68.75

All three distributions tested negatively for normality, and for homogeneity of variances. This means that ANOVA has to be discarded as an option for the hypothesis test. Instead, the Kruskal-Wallis test was applied, since it is suited for non-parametric distributions. The hypotheses for this research question are reiterated below:

Hypothesis 1. *The mean length of predictive notebooks is unequal to the mean length of simple descriptive or descriptive mining notebooks.*

Null Hypothesis 1. *The mean length of predictive notebooks is equal to the mean length of simple descriptive or descriptive mining notebooks.*

The p-value of the Kruskal-Wallis test was less than 0.05, therefore the null hypothesis can be rejected. The post-hoc Dunn test was performed due to the rejection of the null hypothesis. The table 5.11 shows the results of the Dunn test. The highest p-value is between the samples simple descriptive and predictive mining with 0.029. Therefore, the null hypothesis can be rejected for all three samples. Hence, the data favours the alternative hypothesis that the data stems from different distributions.

Table 5.11: P-values of the post-hoc Dunn test

	Simple Descriptive	Descriptive Mining	Predictive Mining
Simple Descriptive	-	0.000004	8.919823e-27
Descriptive Mining	4.25664e-06	-	2.941777e-02
Predictive Mining	8.919823e-27	0.029418	-

5.2.11 Is the mean of data cleaning tasks in predictive notebooks equal to the mean in simple descriptive and descriptive mining notebooks?

Figure 5.18 shows the distribution of the number of data cleaning tasks per each notebook for all three notebook types in a kernel density estimation. The x-axis displays the number of data cleaning tasks per notebook in a logarithmic scale, and the y-axis shows the density. Furthermore, table 5.12 displays the mean and standard deviation for the distribution of the data cleaning tasks for each notebook type. The predictive mining dataset has the highest mean with 7.41 and the highest standard deviation with 8.78. It is followed by the descriptive mining distribution with a mean of 7.12, albeit with the lowest standard deviation of 8.02. Last but not least, the simple discriptive dataset has a mean of 5.66, and a standard deviation of 8.35.

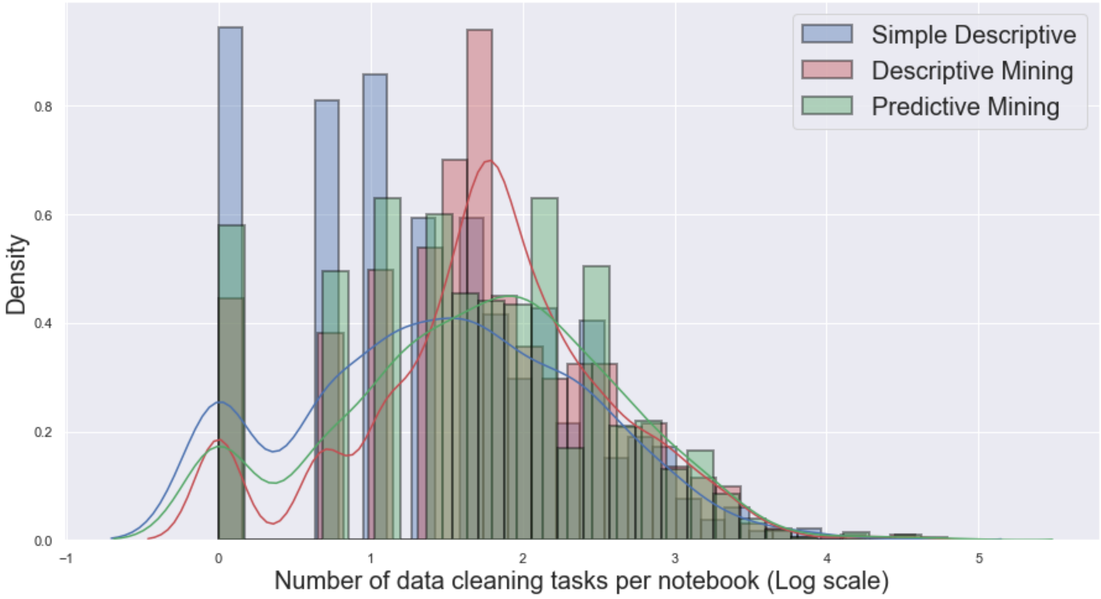


Figure 5.18: Log-scaled distribution of total data cleaning tasks per notebook type

Table 5.12: Mean and standard deviation of the data cleaning tasks distribution for each notebook type

	Simple Descriptive	Mining Descriptive	Predictive Mining
Mean	5.66	7.12	7.41
Standard Deviation	8.35	8.02	8.78

All three distributions tested negatively for normality, and for homogeneity of variances. This means that ANOVA has to be discarded as an option for the hypothesis test. Instead, the Kruskal-Wallis test can be used, since it is suited for non-parametric distributions. The hypotheses for this research question are reiterated below:

Hypothesis 2. *The mean of data cleaning tasks in predictive notebooks is unequal to the mean of data cleaning tasks in simple descriptive or descriptive mining notebooks.*

Null Hypothesis 2. *The mean of data cleaning tasks in predictive notebooks is equal to the mean of simple descriptive or descriptive mining notebooks.*

The p-value of the Kruskal-Wallis test was less than 0.05, therefore the null hypothesis can be rejected. The post-hoc Dunn test was performed due to the rejection of the null hypothesis. The table 5.13 shows the results of the Dunn test. The hypothesis cannot be rejected between the samples descriptive mining and predictive mining. However, the hypothesis can be rejected between the samples predictive mining and simple descriptive. Additionally, the hypothesis can also be rejected between the samples simple descriptive and descriptive mining. Hence, the data favours the null hypothesis that the mean of data cleaning tasks stems from the same distributions only between the samples predictive mining and descriptive mining.

Table 5.13: P-values of the post-hoc Dunn test

	Simple Descriptive	Descriptive Mining	Predictive Mining
Simple Descriptive	-	1.086537e-23	3.281756e-45
Descriptive Mining	1.086537e-23	-	3.254378e-01
Predictive Mining	3.281756e-45	3.254378e-01	-

5.2.12 Is the mean of data integration tasks in predictive notebooks equal to the mean in simple descriptive and descriptive mining notebooks?

Figure 5.19 shows the distribution of the number of data integration tasks per each notebook for all three notebook types in a kernel density estimation. The x-axis displays the number of data integration tasks per notebook in a logarithmic scale, and the y-axis shows the density. Table 5.14 displays the mean and standard deviation for the distribution of the data integration tasks for each notebook type. The predictive mining dataset has yet again the highest mean with 5.98 and the highest standard deviation with 8.21. It is followed by the descriptive mining distribution, which has a mean of 4.95, albeit with a lower standard deviation than all three distributions with 5.33. Last but not least, the distribution for the simple descriptive notebooks has a mean of 4.94, with a standard deviation of 5.34

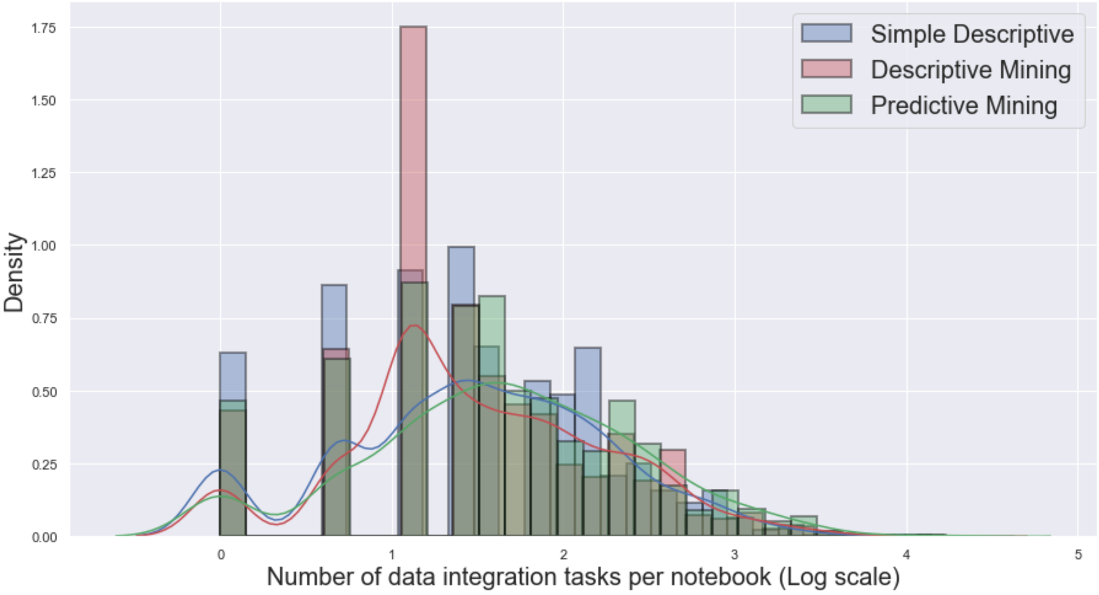


Figure 5.19: Log-scaled distribution of total data integration tasks per notebook type

Table 5.14: Mean and standard deviation of the distribution of data integration tasks per notebook.

	Simple Descriptive	Mining Descriptive	Predictive Mining
Mean	4.94	4.95	5.98
Standard Deviation	5.34	5.33	8.21

All three distributions tested negatively for normality, and for homogeneity of variances. This means that ANOVA has to be discarded as an option for the hypothesis test. Instead, the Kruskal-Wallis test can be used, since it is suited for non-parametric distributions. The hypotheses for this research question are reiterated below:

Hypothesis 3. *The mean of data integration tasks in predictive notebooks is unequal to the mean of data integration tasks in simple descriptive or descriptive mining notebooks.*

Null Hypothesis 3. *The mean of data integration tasks in predictive notebooks is equal to the mean of data integration tasks in simple descriptive or descriptive mining notebooks.*

The p-value of the Kruskal-Wallis test was less than 0.05, therefore the null hypothesis can be rejected. The post-hoc Dunn test was performed due to the rejection of the null hypothesis. The table 5.15 shows the results for the Dunn test. The only null hypothesis, which cannot be rejected is between the descriptive mining and simple descriptive samples. It can be rejected however between the distributions simple descriptive and predictive mining. Furthermore, it can also be rejected between the distributions predictive mining and descriptive mining. Hence, the data favours the null hypothesis that the mean of data integration tasks stems from the same distributions only between the samples descriptive mining and simple descriptive.

Table 5.15: P-values of the post-hoc Dunn test

	Simple Descriptive	Descriptive Mining	Predictive Mining
Simple Descriptive	-	4.907318e-01	9.366997e-16
Descriptive Mining	4.907318e-01	-	8.679414e-09
Predictive Mining	9.366997e-16	8.679414e-09	-

5.2.13 Is the mean of data cleaning cells across all notebook types equal to the mean of all data integration cells across the three notebook types?

Figure 5.20 shows two distributions in a kernel density estimation: number of data integration tasks and data cleaning tasks per notebook across all notebook types. The x-axis displays both tasks in a logarithmic scale, and the y-axis shows the density. Table 5.16 displays the mean and standard deviation for both distributions. The mean and standard deviation of the total data cleaning tasks distribution is higher than that of the data integration tasks distribution with a mean of 6.56 compared to 5.38.

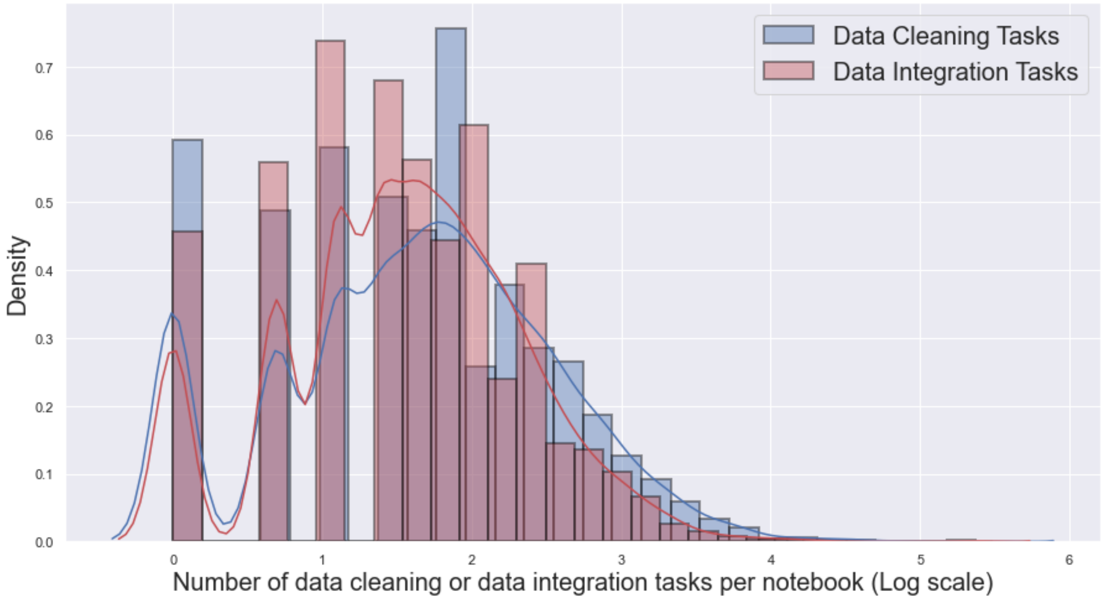


Figure 5.20: Log-scaled distributions of total data integration and data cleaning tasks per notebook type

Table 5.16: Mean and standard deviation of the data cleaning and data integration distributions across all notebook types

	Total Data Cleaning Tasks	Total Data Integration Tasks
Mean	6.56	5.38
Standard Deviation	8.54	6.72

Both distributions tested negatively for normality, and for homogeneity of variances. This means that ANOVA has to be discarded as an option for the hypothesis test.

Instead, the Kruskal-Wallis test can be used, since it is suited for non-parametric distributions. The hypotheses for this research question are listed below:

Hypothesis 4. *The mean of data cleaning tasks is unequal to the mean of data cleaning tasks across all three notebook types.*

Null Hypothesis 4. *The mean of data cleaning tasks is equal to the mean of data cleaning tasks across all three notebook types.*

The p-value of the Kruskal-Wallis test was less than 0.05, therefore the null hypothesis can be rejected. Hence, the data favours the alternative hypothesis that the mean of data integration tasks and the mean of data cleaning tasks do not stem from the same distributions.

5.2.14 What is the percentage of code dedicated to data cleaning in different types of data science notebooks?

Three plots were created in order to display the three distributions containing the percentage of code dedicated to data cleaning. All figures display the percentage of code dedicated to data cleaning on the x-axis, while the y-axis shows the density. While figure 5.24 shows the kernel density estimation for notebooks of type simple descriptive, figure 5.25 shows the distribution for descriptive mining and figure 5.26 for predictive mining notebooks. Table 5.17 shows the mean and standard deviation of all three distributions. The descriptive mining dataset has the highest mean with 10.98% of data cleaning percentage per notebook. The dataset containing the notebooks classified as simple descriptive, however, has the highest standard deviation of 10.77%.

Table 5.17: Table with the mean and standard deviation of the three distributions

	Simple Descriptive	Mining Descriptive	Predictive Mining
Mean	10.18%	10.98%	10.81%
Standard Deviation	10.77%	10.14%	9.95%

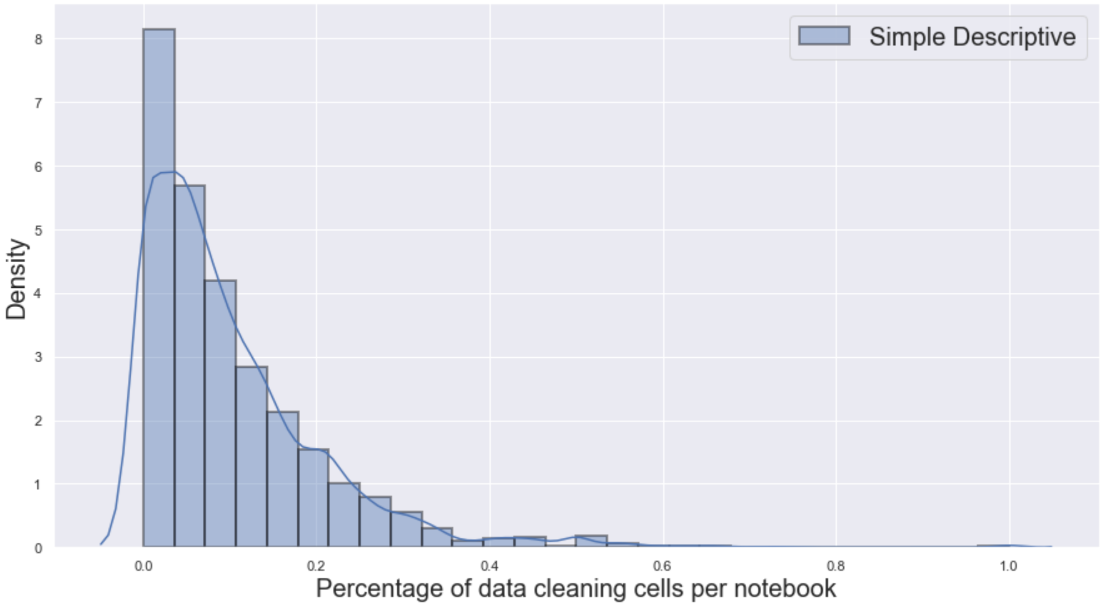


Figure 5.21: Distribution of the percentage of data cleaning cells in simple descriptive notebooks

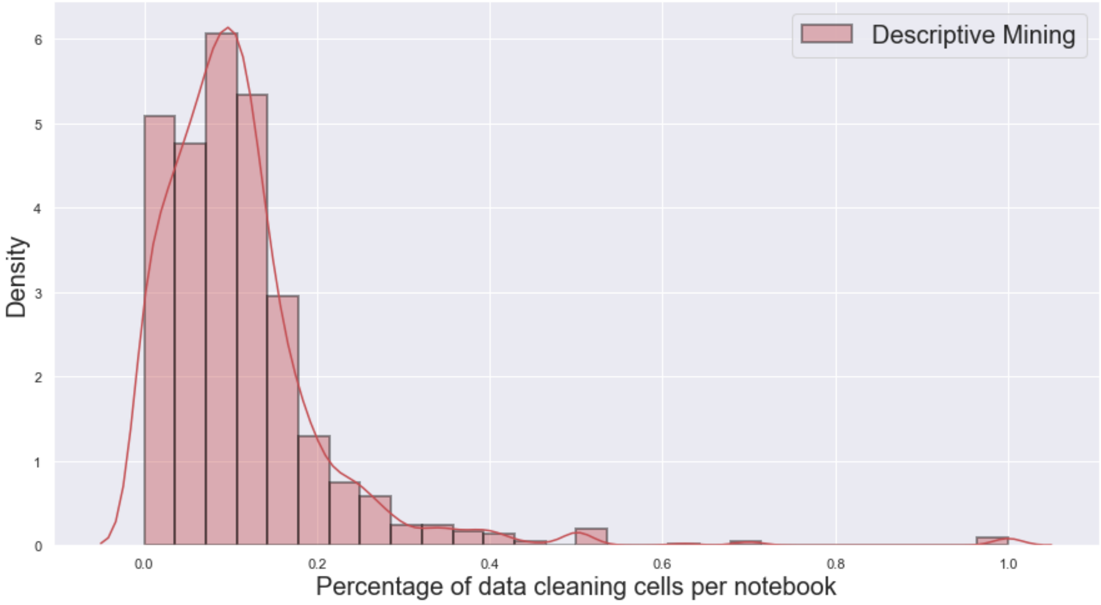


Figure 5.22: Distribution of the percentage of data cleaning cells in descriptive mining notebooks

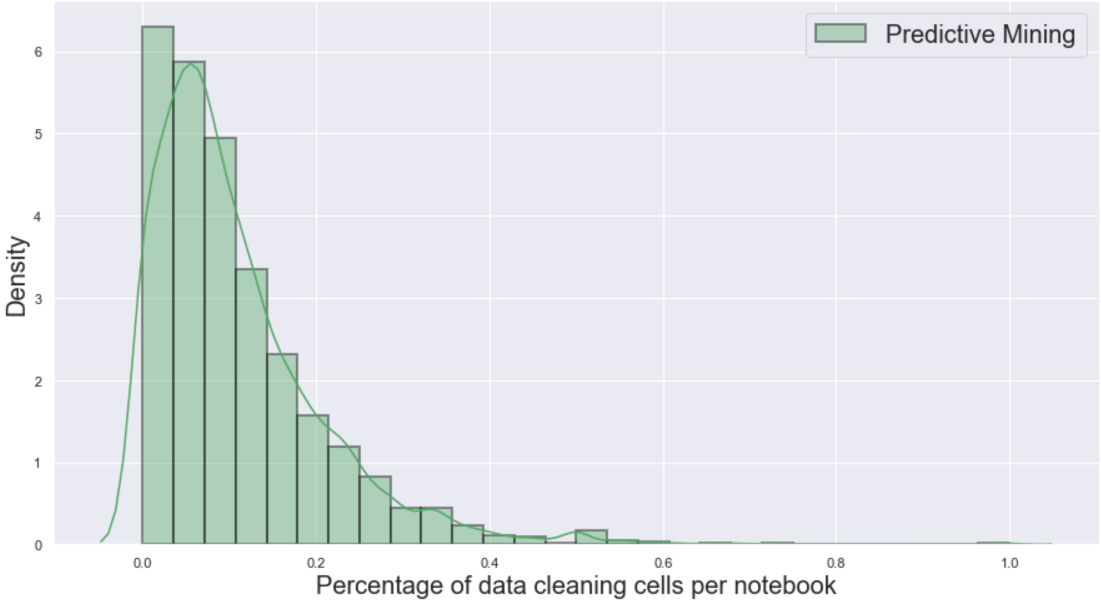


Figure 5.23: Distribution of the percentage of data cleaning cells in predictive mining notebooks

5.2.15 What is the percentage of code dedicated to data integration in different types of data science notebooks?

Three plots were created in order to display the three distributions containing the percentage of code dedicated to data integration. All figures display the percentage of code dedicated to data integration on the x-axis, while the y-axis shows the density. Figure 5.21 shows the kernel density estimation for notebooks of type simple descriptive, while figure 5.22 does the same for descriptive mining and figure 5.23 for predictive mining notebooks. Table 5.18 shows the mean and standard deviation of all three distributions. The simple descriptive dataset has the highest mean with 11.31% of data cleaning percentage per notebook and the highest standard deviation with 11.09%.

Table 5.18: Table with the mean and standard deviation of the three distributions

	Simple Descriptive	Mining Descriptive	Predictive Mining
Mean	11.31%	9.55%	10.91%
Standard Deviation	11.09%	10.05%	10.43%

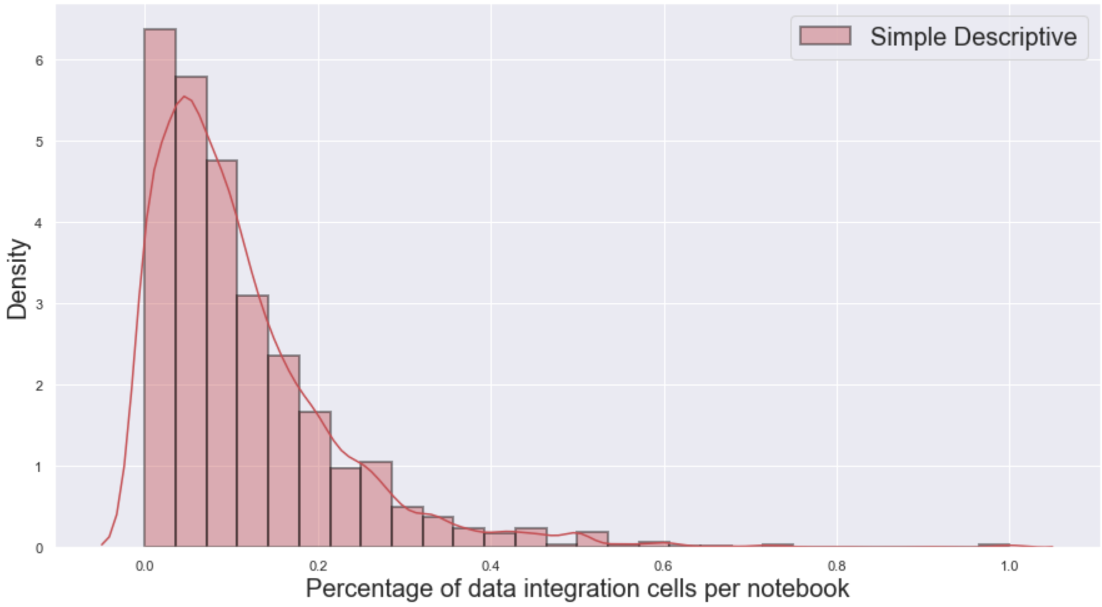


Figure 5.24: Distribution of the percentage of data integration cells in simple descriptive notebooks

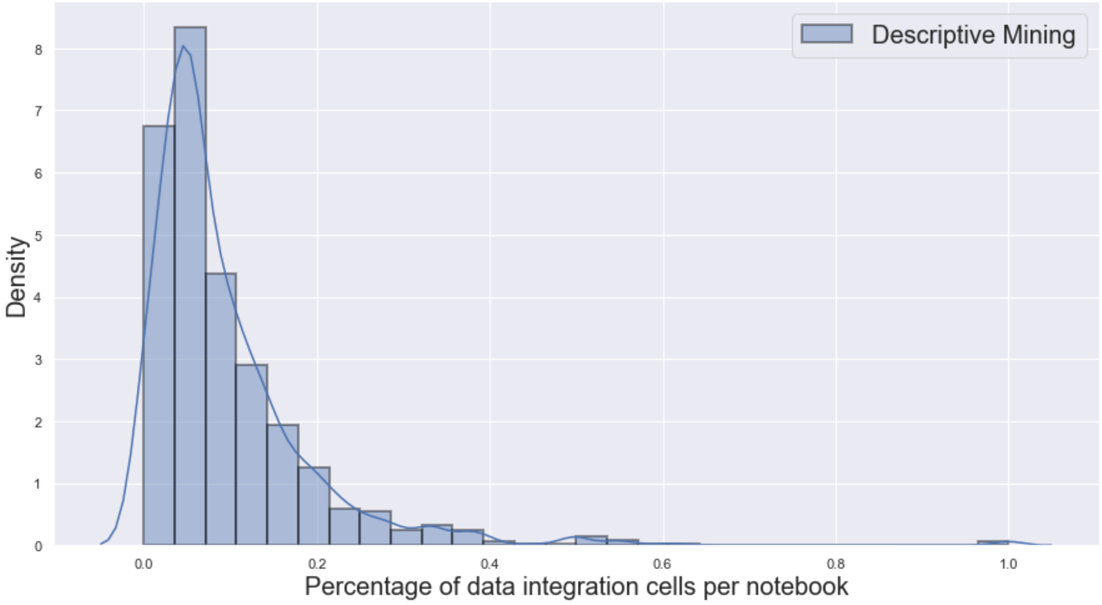


Figure 5.25: Distribution of the percentage of data integration cells in descriptive mining notebooks

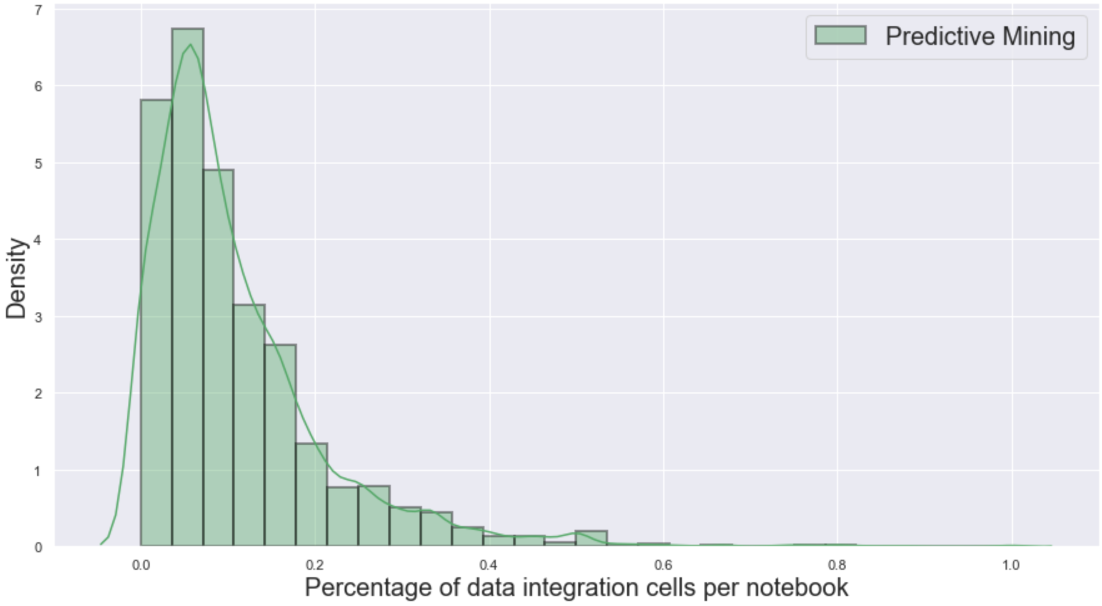


Figure 5.26: Distribution of the percentage of data integration cells in predictive mining notebooks

5.3 What is the relation between data cleaning and other data science steps?

5.3.1 Does data cleaning happen iteratively?

A notebook applies data cleaning iteratively when two data cleaning cells are at least more than one cell apart. The descriptive mining dataset has the highest percentage of notebooks with iterative data cleaning tasks at 84.00 percent. This is followed by the predictive mining dataset, which contains 79.90 percent of iterative data cleaning cells. The dataset with the least percentage of iterative data cleaning cells is the dataset that contains cells classified as simple descriptive with a percentage of 69.60.

5.3.2 What is the ratio between data cleaning and simple, predictive, descriptive and visualization cells?

Figure 5.27 displays the four different ratios. The first one is the ratio between the total data cleaning cells, and the simple descriptive cells. This was calculated solely for the notebooks of type simple descriptive. This is followed by the ratio displayed in the color blue, which shows the ratio between the total data cleaning cells and the descriptive mining cells. This was also only calculated for the notebooks of type descriptive mining. Furthermore, the ratio shown in the color green is the number of total data cleaning cells divided by the number of predictive mining cells in the predictive mining dataset. Last but not least, the ratio shown in yellow is the total number of data cleaning cells divided by the total number of data visualization cells across all notebook types. The results demonstrate, that the ratio between the total simple descriptive cells and the data cleaning cells is the highest with 1.44. This means that for every data cleaning cell there are 1.44 simple descriptive cells. The lowest ratio is shown for the visualization ratio with a value of 0.76.

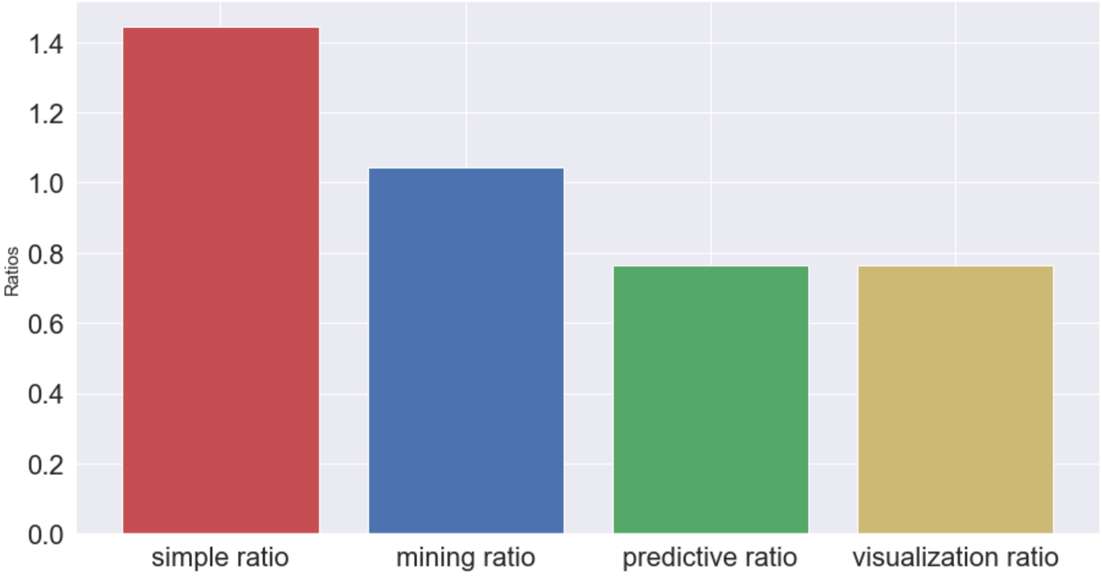


Figure 5.27: Ratios between total data cleaning and simple, predictive, descriptive and visualization cells

5.4 What is the relation between data integration and other data science steps?

5.4.1 Does data integration happen iteratively?

A notebook applies data integration iteratively when two data integration cells are at least more than one cell apart. The descriptive mining dataset has the highest percentage of notebooks with iterative data integration tasks at 83.20 percent. This is followed by the predictive mining dataset, which contains 82.20 percent. The dataset with the least percentage of iterative data integration cells is the dataset that contains cells classified as simple descriptive with a percentage of 77.80.

5.4.2 What is the ratio between data integration and simple, predictive, descriptive and visualization cells?

Figure 5.28 displays the four different ratios. The first one is the ratio between the total data integration cells, and the simple descriptive cells. This was calculated solely for the notebooks of type simple descriptive. This is followed by the ratio displayed in the color blue, which shows the ratio between the total data integration cells and the descriptive mining cells. This was also only calculated for the notebooks of type descriptive mining. Furthermore, the ratio shown in the color green is the number of total data integration cells divided by the number of predictive mining cells in the predictive mining dataset. Last but not least, the ratio shown in yellow is the total number of data integration cells divided by the total number of data visualization cells across all notebook types. The results demonstrate, that the ratio between the total simple descriptive cells and the data integration cells is the highest with 1.49. This means that for every data integration cell there are 1.49 simple descriptive cells. The lowest ratio is shown for the visualization ratio with a value of 0.74.

Figure 5.28 shows the ratios between the total data integration cells, and the simple, predictive, descriptive and visualization cells. The ratio between the total simple descriptive cells and the data integration cells is the highest with 1.49. This means that for every data integration cell there are 1.49 simple descriptive cells.

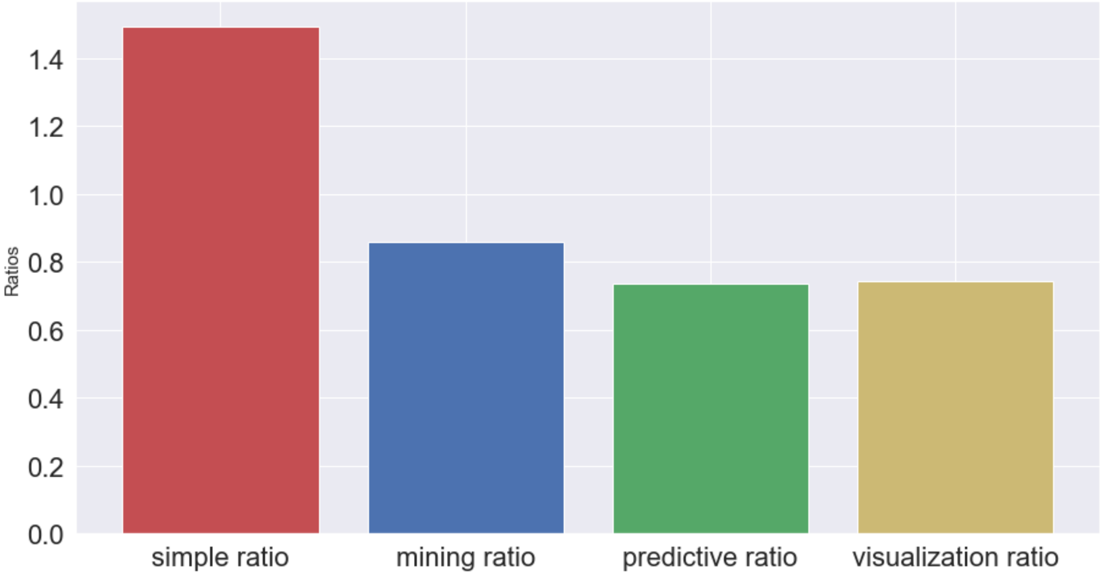


Figure 5.28: Ratios between total data integration and simple, predictive, descriptive and visualization cells.

6

Discussions

The aim of the research questions was to empirically analyze data science notebooks with regards to data cleaning and data integration, and thereby possibly extracting new insights on how data scientists work. The results clearly state that on average there are between 5.7 to 6.9 data cleaning cells inside of a data science notebook, depending on the notebook type. The results also indicate the data cleaning amounts to be between 10.18 and 10.98 percent of the entire notebook, depending on the type of notebook. This is especially interesting, because the related work mentions that data cleaning has been shown to require up to 80% of the time in a data science project (Furche et al., 2016), while the results of this thesis indicate that it still only makes between 10.18% and 10.98% percent on average of a data science notebook, depending on the notebook type. Moreover, these results back Krishnan et al. (2016) claim that data cleaning is a non-linear and an iterative process. Furthermore, the data also suggests that for every predictive mining cell there are on average 0.76 data cleaning cells. This could imply that predictive mining notebooks place a higher emphasis on prediction itself than on data cleaning. Additionally, for every visualization cell across all notebook types there are on average only 0.76 data cleaning cells. This is an interesting insight, considering that this implies that there is a greater emphasis on visualization rather than on data cleaning across all notebook types. Maybe this is due to the fact that data scientists want to first visualize data before deciding to go for data cleaning. Futhermore, a closer examination of the data cleaning tasks shows that *Outlier Detection* was almost non-existent across all notebook types and groups, while *Unify Formatting* stands out as one of the tasks most prevalent throughout all notebook types.

Furthermore, the results also suggest with regards to data integration that *Merging* is more prevalent in data science notebooks of all types and groups rather than *Dataset Comparison*. Interestingly enough, the percentage of data integration cells inside of a data science notebook is higher than the percentage of data cleaning cells in simple descriptive notebooks, but lower in descriptive mining notebooks. The data also suggests that data integration is a non-linear and iterative process similar to data cleaning. Moreover, there are 0.73 data integration cells for every predictive mining and 0.74 data integration cells for every visualization cell. Implying that prediction and visualization might have a higher emphasis in data science over both data cleaning and data integration cells, at least with regards to the amount of total cells in a notebook.

The data also shows that the hypothesis number one stating that the mean length of the three notebook types were equal and could not be rejected, which is interesting considering that each of the three notebook types has a different focus. This thesis also provides new evidence through hypothesis number four , whose intention was to test the mean of data cleaning tasks with the mean of data integration tasks. Interestingly enough, this hypothesis also could not be rejected, stating that both data management tasks have an equal mean across all notebook types.

Overall, the keyword-based labelling system is limited by the set of keywords, which were not identified during the evaluation process. Nonetheless, the upside to this approach are three large annotated datasets, which have been already manually evaluated for precision, recall and accuracy, and could be used in further studies. Furthermore, another limitation might lie in the measurement method counting the number of files imported. This is because there are some instances were a file imported is being counted twice due to some keywords that can appear alone or in conjunction with other keywords, e.g., *"open("* and *"load("*.

Further research could investigate where in the process data cleaning occurs, as well as the similarity between the datasets used in a notebook. Additionally, it would be interesting to further research data visualization within the context of data science given the emphasis it was given in the notebooks. It might be that the reason we see so much more data visualization happening is because this is a crucial step in understanding the data better and understanding where the data needs to be cleaned.

7

Conclusions

This thesis aimed to provide new insights with regards to data cleaning and data integration tasks within data science notebooks, and how these tasks vary depending on the type of data science notebook. Based on a quantitative analysis, it can be concluded that the percentage of data cleaning and data integration present in the data science notebooks, and across all notebook types, is similar. The results also indicate that while data cleaning has been shown to require up to 80% of the time in a data science project Furche et al. (2016), it only amounts to 10.98% of an entire notebook. The research also backs Krishnan et al. (2016) claim that data cleaning is a non-linear and iterative process. Moreover, this thesis has shown that data integration as well, is a non-linear and iterative process.

However, the research also raises the question of the importance of visualization within data science notebooks. The data suggests that data visualization appears more frequent than both data cleaning and data integration, considering the representation of data visualization cells with regards to both data cleaning and data integration cells. Furthermore, while the keyword-based labelling system is limited by the keywords, which weren't identified during the evaluation process, the approach provides three annotated datasets of different types of data science.

Based on these conclusions, further research should consider investigating where data cleaning happens inside of a notebook, and how similar the datasets are, which are being imported. Furthermore, future studies are needed to establish if data visualization plays a bigger role than data cleaning and data integration, and why this could be the case.

References

Breunig, M. M., Kriegel, H.-P., Ng, R. T., and Sander, J. (2000). Lof: identifying density-based local outliers. In *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, pages 93–104.

Corrales, D. C., Ledezma, A., and Corrales, J. C. (2018). From theory to practice: A data quality framework for classification tasks. *Symmetry*, 10(7):248.

CrowdFlower (2016). Data science report. *https://visit.figure-eight.com/rs/416-ZBE-142/images/CrowdFlower_DataScienceReport_2016.pdf*, last accessed on 2020-08-21.

Cupoli, P., Earley, S., and Henderson, D. (2014). Dama-dmbok2 framework. *DAMA International*.

Dasu, T. and Loh, J. M. (2012). Statistical distortion: Consequences of data cleaning. *arXiv preprint arXiv:1208.1932*.

Dinno, A. (2015). Nonparametric pairwise multiple comparisons in independent groups using dunn's test. *The Stata Journal*, 15(1):292–300.

Feinberg, M., Carter, D., and Bullard, J. (2014). A story without end: writing the residual into descriptive infrastructure. In *Proceedings of the 2014 conference on Designing interactive systems*, pages 385–394.

Feinberg, M., Carter, D., Bullard, J., and Gursoy, A. (2017). Translating texture: Design as integration. In *Proceedings of the 2017 Conference on Designing Interactive Systems*, pages 297–307.

Figure-Eight (2018). Data science report. *https://visit.figure-eight.com/rs/416-ZBE-142/images/Data-Scientist-Report.pdf*, last accessed on 2020-08-21.

Furche, T., Gottlob, G., Neumayr, B., and Sallinger, E. (2016). Data wrangling for big data: Towards a lingua franca for data wrangling. CEUR Workshop Proceedings.

Hellerstein, J. M. (2008). Quantitative data cleaning for large databases. *United Nations Economic Commission for Europe (UNECE)*, 25.

Jupyter, T. (2015). What is the jupyter notebook? *https://jupyter-notebook.readthedocs.io/en/latest/examples/Notebook/What%20is%20the%20Jupyter%20Notebook.html*, last accessed on 2020-08-21.

Kery, M. B., Radensky, M., Arya, M., John, B. E., and Myers, B. A. (2018). The story in the notebook: Exploratory data science using a literate programming tool. New York, NY, USA. Association for Computing Machinery.

Krishnan, S., Haas, D., Franklin, M. J., and Wu, E. (2016). Towards reliable interactive data cleaning: A user survey and recommendations. In *Proceedings of the Workshop on Human-In-the-Loop Data Analytics*, pages 1–5.

Kruskal, W. H. and Wallis, W. A. (1952). Use of ranks in one-criterion variance analysis. *Journal of the American statistical Association*, 47(260):583–621.

Levene, H. (1960). Robust tests for equality of variances. contributions to probability and statistics in olkin i, ed.

Miller, R. J. (2017). The future of data integration. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '17, page 3, New York, NY, USA. Association for Computing Machinery.

Mooney, P. (2018). Kaggle machine learning and data science survey. *https://www.kaggle.com/paultimothymooney/2018-kaggle-machine-learning-data-science-survey/notebook*, last accessed on 2020-08-21.

Muller, M., Lange, I., Wang, D., Piorkowski, D., Tsay, J., Liao, Q. V., Dugan, C., and Erickson, T. (2019). How data science workers work with data: Discovery, capture, curation, design, creation. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, pages 1–15.

Olson, D. L. and Delen, D. (2008). *Advanced data mining techniques.* Springer Science & Business Media.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Ramasamy, D. (2019). Automatic annotation of data science notebooks. Master's thesis.

Review, H. B. (2018). What data scientists really do, according to 35 data scientists. *https://hbr.org/2018/08/what-data-scientists-really-do-according-to-35-data-scientists*.

Rule, A., Tabard, A., and Hollan, J. D. (2018). Exploration and explanation in computational notebooks. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, pages 1–12.

Schutt, R. and O'Neil, C. (2013). *Doing Data Science: Straight Talk from the Frontline.* O'Reilly Media, Inc.

Shapiro, S. S. and Wilk, M. B. (1965). An analysis of variance test for normality (complete samples). *Biometrika*, 52(3/4):591–611.

Sutton, C., Hobson, T., Geddes, J., and Caruana, R. (2018). Data diff: Interpretable, executable summaries of changes in distributions for data wrangling. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2279–2288.

Tan, P.-N., Steinbach, M., and Kumar, V. (2016). *Introduction to data mining.* Pearson Education India.

Wang, A. Y., Mittal, A., Brooks, C., and Oney, S. (2019). How data scientists use computational notebooks for real-time collaboration. *Proceedings of the ACM on Human-Computer Interaction*, 3(CSCW):1–30.

A

Appendix

A.1 Labelling Keywords

Table A.1: Table with the full list of keywords for the label isSimpleDescriptive

isSimpleDescriptive
”pivot_table(”,”crosstab(”,”harmonic_mean(”,”geometric_mean(”, ”hmean(”,”mean(”,”fmean(”,”median_low(”,”median_high(”, ”median_grouped(”,”std(”,”.max(”,”.min(”,”.describe(”,”.mean(”, ”.median(”,”.variance(”,”.mode(”,”.percentile(”,”.quantile(”, ”.quantiles(”,”pstdev(”,”pvariance(”,”.var(”,”.stdev(”,”.skew(”, ”.skewness(”,”.kurt(”,”.kurtosis(”,”.cov(”,”shapiro(”,”normaltest(”, ”anderson(”, ”pearsonr(”, ”spearmanr(”, ”kendalltau(”, ”chi2_contingency(”, ”adfuller(”,”kpss(”,”ttest_ind(”,”ttest_rel(”, ”f_oneway(”,”mannwhitneyu(”,”wilcoxon(”,”kruskal(”, ”friedmanchisquare(”,”binom_test(”,”chisquare(”,”ztest(”, ”bartlett(”,”cumfreq(”,”relfreq(”,”ttest_1samp(”,”ttest_ind(”, ”ttest_ind_from_stats(”,”ttest_rel(”,”kstest(”, ”normaltest(”,”skewtest(”,”kurtosistest(”

Table A.2: Table with the full list of keywords for the label isDataLoading

isDataLoading
”load_CIFAR10(”,”load_image(”,”read_data_sets(”,”read_mat(”, ”get_CIFAR10_data()”,”DataReader(”,”loadmat(”,”.genfromtxt(”, ”SFrame(”,”read_file(”,”read_json(”,”fetch_open_data”,”load_builtin(”, ”load_from_df(”,”load_data(”,”load_boston”,”load_iris”, ”load_diabetes”, ”load_digits”, ”load_wine”, ”load_breast_cancer”,”read_sql(”,”loadtxt(”, ”open (”,”open(”, ”load(”, ”popen(”,”read_csv(”, ”read_excel(”, ”read_table(”,”read_pickle(”,”imread(”,”load_img(”

Table A.3: Table with the full list of keywords for the label isDescriptiveMining

isDescriptiveMining
”autocorr”,”MarkovChain(”,”get_outliers_inliers(”, ”LocalOutlierFactor(”,”EllipticEnvelope(”,”IsolationForest(”, ”.rolling(”,”pacf(”,”acf(”,”rolling_std(”,”rolling_mean(”, ”seasonal_decompose(”,”apriori(”,”association_rules(”, ”frequent_itemsets(”,”.corrcoef(”,”.corr(”,”AffinityPropagation(”, ”AgglomerativeClustering(”, ”SpectralClustering(”, ”DBSCAN(”, ”KMeans(”, ”MeanShift(”,”FeatureAgglomeration(”, ”OPTICS(”,”Birch(”, ”MiniBatchKMeans(”

Table A.4: Table with the full list of keywords for the label isDataVisualization

isDataVisualization
"plt.show()","scatter_matrix(",".hist()","plt.plot()", "plt.subplots()","plot(","plt.figure()","sns.distplot(", "sns.kdeplot(","sns.jointplot(","sns.rugplot(", "sns.pairplot(","scatter(","plt.hist(","vs.cluster_results("

Table A.5: Table with the full list of keywords for the label isPredictiveMining

isPredictiveMining
”quantileRegression(”,”quantreg(”,”Logit(”, ”NeuralNetwork(”,”logistic_regression_with_L2(”, ”xgb.XGBRegressor(”,”xgb.Classifier(”, ”xgb.DMatrix(”,”OLS(”,”BackpropTrainer(”, ”ExponentialSmoothing(”,”SimpleExpSmoothing(”, ”VARMAX(”,”VAR(”,”SARIMAX(”,”ARIMA(”, ”ARMA(”,”AutoReg(”,”tf.estimator”, ”tf.keras.Sequential()”,”predict_classes(”, ”predict_proba(”,”.ols(”,”roc_curve”, ”cross_val_predict”,”.accuracy_score(”, ”linregress(”,”recommend(”,”item_similarity_recommender” ,”LogisticRegression(”,”DecisionTreeClassifier(”, ”RandomForestClassifier(”,”GradientBoostingClassifier(”, ”GradientBoostingRegressor”,”KNeighborsRegressor(”, ”KNeighborsClassifier(”, ”LinearSVC()” ,”ExtraTreesClassifier(” ,”XGBClassifier(” ,”LGBMClassifier(”,” GridSearchCV(”, ” DNNClassifier(”,” RandomForestRegressor(”,” tf.Session()”, ”tf.Variable(”, ”NearestNeighbors(”, ”cross_val_score(”, ”model = sequential()”,”LinearRegression(”, ”Ridge(”, ”RidgeCV(”, ”Lasso(”, ”MultiTaskLasso(”, ”ElasticNet(”, ”MultiTaskElasticNet(”, ”Lars(”, ”.LassoLars(”, ”BayesianRidge(”, ”SVC(”, ”NuSVC(”, ”SVR(”, ”LinearSVR(”,”NuSVR(”, ”SGDRegressor(”, ”SGDClassifier(”, ”MLPClassifier(”, ”GaussianNB(”, ”MultinomialNB(”, ”ComplementNB(”, ”BernoulliNB(”, ”CategoricalNB(”, ”DecisionTreeRegressor(”, ”MLPClassifier(”,”models.resnet18(”,”models.alexnet(”, ”models.vgg16()”, ”models.squeezenet1_0()” ,”models.densenet161()”, ”models.inception_v3()”, ”models.googlenet()”, ”models.shufflenet_v2_x1_0()”, ”models.mobilenet_v2()”, ”models.resnext50_32x4d()”, ”models.wide_resnet50_2()”, ”models.mnasnet1_0()”, ”classifier.classify(”

Table A.6: Table with the full list of keywords for the label isDataCleaning

isDataCleaning
”OneSidedSelection(”,”PCA(”, ”IncrementalPCA(”,”KernelPCA(”, ”SparsePCA(”, ”TruncatedSVD(”, ”NMF(”, ”RandomOverSampler(”, ”SMOTE(”, ”SMOTENC(”, ”RandomUnderSampler(”, ”CondensedNearestNeighbour(”, ”EditedNearestNeighbours(”, ”NeighbourhoodCleaningRule(”,”TomekLinks(”, ”SimpleImputer(”, ”IterativeImputer(”, ”KNNImputer(”, ”MissingIndicator(”, ”ffill(”, ”mice(”,”locf(”, ”option = ’locf’”, ”option = ’nocb’”, ”method=’ffill’”,”method=’bfill’”, ”method=’backfill’”, ”to_numeric(”,”to_datetime(”,”.replace(0,nan)”, ”remove_columns(”,”remove_column(”, ”.replace(’[’,”)”,”.replace(’]’,”)”,”.replace(’\\”,”)”, ”.replace(’ ’,’_’)”,”.replace(’-’,’ ’)”,”.duplicated(”, ”.notnull()”,”drop(”,”np.delete(”,”np.nan()”, ”.nunique()”, ”.unique()”,”set_index(”, ”rename(”, ”dropna(”,”isnull(”,”fillna(”, ”drop_duplicates(”,”reindex_axis”

Table A.7: Table with the full list of keywords for the label isDataIntegration

isDataIntegration
”merge (”, ”merge(”,”merge_ordered”,”merge_asof”

A.2 Mining Keywords

A.2.1 Data Cleaning Keywords

Table A.8: Table with the full list of data cleaning keywords

Imputation	Outlier Detection	Dimensionality Reduction	Balanced Classes
’replace(nan,0)’, ’dropna(’, ’fillna(’, ’SimpleImputer(’, ’IterativeImputer(’, ’KNNImputer(’, ’MissingIndicator(’, ’ffill’, ’bfill’, ’backfill’, ’MICE(’, ’set_imputer(’, ’locf’, ’nocb’	’DBSCAN(’, ’LocalOutlierFactor(’, ’ABOD(’, ’EllipticEnvelope(’, ’IsolationForest(’	’PCA(’, ’IncrementalPCA(’, ’KernelPCA(’, ’SparsePCA(’, ’TruncatedSVD(’, ’SparseCoder(’, ’NMF(’	’sample(’, ’RandomOverSampler(’, ’over_sampling(’, ’under_sampling(’, ’SMOTE(’, ’SMOTENC(’, ’RandomUnderSampler(’, ’CondensedNearestNeighbour(’, ’EditedNearestNeighbours(’, ’NeighbourhoodCleaningRule(’, ’TomekLinks(’
Removing Duplicates	Removing Redundancies	Data Transformation	Unify Formatting
’drop_duplicates(’	’remove_column(’, ’remove_columns(’, ’drop(’	’to_datetime(’, ’to_numeric(’	”.replace(’[’,”)”, ”.replace(’]’,”)”, ”.replace(’\\”,”)”, ”.replace(’_’,”)”, ”.replace(’-’,’ ’)”, ”replace(’[’,”)”, ’rename(’, ’reset_index(’, ’set_index(’

A.2.2 Data Integration Keywords

Table A.9: Table with the full list of data integration keywords

Merging	Dataset Comparison
”merge(”, ”join(”, ”concat(”, ”merge_asof(”, ”merge_ordered(”, ”merge_asof(”, ”append(”	’intersection(’, ’diff(’

A.3 Group Keywords

A.3.1 Regression versus Classification Group

Table A.10: Table with the full list of the regression and classification keywords

Regression Group	Classification Group
”quantileRegression(”,	”SVC(”,
”quantreg(”,	”SGDClassifier(”, ”KNeighborsClassifier(”,
”logistic_regression_with_L2(”,	”DecisionTreeClassifier(”,
”xgb.XGBRegressor(”,	”MLPClassifier(”,
”OLS(”, ”.ols(”,	”xgb.Classifier(”,
”LinearRegression(”,	”RandomForestClassifier(”,
”LogisticRegression(”,	”GradientBoostingClassifier(”,
”SVR(”, ”SGDRegressor(”,	”ExtraTreesClassifier(” ,
”KNeighborsRegressor(”,	”XGBClassifier(”,
”DecisionTreeRegressor(”,	”LGBMClassifier(”,
”Ridge(”, ”linregress(”,	” DNNClassifier(”,
”GradientBoostingRegressor”	”MLPClassifier(”

A.3.2 Outlier Detection versus Clustering Group

Table A.11: Table with the full list of the outlier detection and clustering keywords

Outlier Group	Clustering Group
"EllipticEnvelope(", "IsolationForest(", 'LocalOutlierFactor(', 'ABOD('	AffinityPropagation(", "AgglomerativeClustering(", "SpectralClustering(", "DBSCAN(", "KMeans(", "MeanShift(", "FeatureAgglomeration(", "OPTICS(", "Birch(", "MiniBatchKMeans("

A.3.3 Statistical Tests versus Simple Statistics Group

Table A.12: Table with the full list of the statistical tests and simple statistics keywords

Statistical Tests Group	Simple Statistics Group
"shapiro(","normaltest(", "anderson(", "pearsonr(", "spearmanr(", "kendalltau(", "chi2_contingency(", "adfuller(", "kpss(","ttest_ind(","ttest_rel(", "f_oneway(","mannwhitneyu(", "wilcoxon(","kruskal(", "friedmanchisquare(","binom_test(", "chisquare(","ztest(", "bartlett(","ttest_1samp(", "ttest_ind(","ttest_ind_from_stats(" ,"ttest_rel(","kstest(","normaltest(", "skewtest(","kurtosistest("	"AffinityPropagation(", "AgglomerativeClustering(", "SpectralClustering(", "DBSCAN(", "KMeans(", "MeanShift(", "FeatureAgglomeration(", "OPTICS(", "Birch(", "MiniBatchKMeans("

List of Figures

List of Tables