Collaborative Data Analysis in a Crowdsourcing Environment Using Jupyter Notebook

Anastasiu Teodor Cristian
University of Zurich
Zurich, Switzerland
cristiananastasiu@gmail.com

ABSTRACT

The availability of data is growing faster than the availability of experts with the relevant skill set needed to interpret it. Finding competent experts for data analysis tasks is becoming increasingly challenging due to the variety of required skills. It is well known that data preparation and filtering steps take a considerable amount of processing time in ML problems [8]. Business and academic settings assume analysts to be proficient not only in the domain of their interest, but also in core analysis disciplines such as statistics, computing, software engineering, and algorithms. Data analysis routines in these domains span over multiple disciplines and individuals involved in their accomplishment are subject to many biases due to their personal traits/background, which may cause errors.

This paper proposes a collaborative data analysis framework based on Jupyter Notebook, allowing structured data analysis tasks to be distributed as a collaborative process to a group of people with a diverse set of abilities and knowledge. Our evaluations showed that data analysis tasks, especially the pre-processing part, can be distributed to non-expert workers, where it is assumed that every member possesses a tiny fragment of the required knowledge and, taken together, they can use their collective intelligence for successful data analytics. Specifically, the goal of this paper is to contribute to this field by discussing and implementing a framework to structure data analysis as a collaborative and distributed process accessible to a public with a diverse set of skills.

Author Keywords

jupyter; ipython; data-analysis; data-mining; collaboration; task distribution

ACM Classification Keywords

H.5.3 Group and Organization Interfaces: Collaborative computing; See http://acm.org/about/class/1998/ for the full list of ACM classifiers.

INTRODUCTION

In this paper, we introduce a collaborative crowd sourcing platform on which the pre-processing of data is divided into steps and each task is assigned to a crowd worker. All steps and notes are visible to all workers involved in the project, which allows each individual to draw on the strengths of the crowd in order to produce the desired results. The workers are non-experts on the project, with limited knowledge on the field of data analysis. Dividing the project into several simple tasks, as opposed to few complex ones, allows the project owner to make use of crowd abilities in order to speed up the project. In our experiments, we test whether the results of non-expert individuals working on multiple simple tasks are comparable to those produced by experts handling the same project. We explore whether drawing on the collaborative strength of crowds returns solid data pre-processing results. This work focuses on data pre-processing, as the tasks can be more minute and simple in this area, meaning we can draw on individuals with a more varied skill set. Data interpretation, encompassing data mining and evaluation, on the other hand, is a more expert level work, in which specific knowledge of the field is required.

This contribution of this study is twofold: technical and empirical. First, on the technical side, we have taken the popular Juypter Notebook (IPython) platform and extended it into a collaborative data analysis platform, with the possibility of working in a crowdsourcing environment. It allows data analysis experts to divide projects into multiple action items and assign them to individual workers. It is assumed that every worker possesses a tiny fragment of the required knowledge and, taken together, they can use their collective intelligence for successful data analytics. Assignments are shared with workers in form of Jupyter notebooks. Workers can work individually on their own task, add notes, but also communicate among each other and with the owner, and produce results for their tasks. In an iterative process, the owner merges the results of all of the tasks, runs the code, analyzes the output and provides feedback for the next interaction. Second, we are testing out the extended platform in several experiments, both in a crowdworking environment and with a group of students. The goal of the experiments is to test two hypotheses:

- The pre-processing part of a data analysis project can be decomposed in small enough tasks such that can be performed by non-expert data scientists.
- The proposed approach of teams with mixed level of expertise leads to results comparable with expert teams.

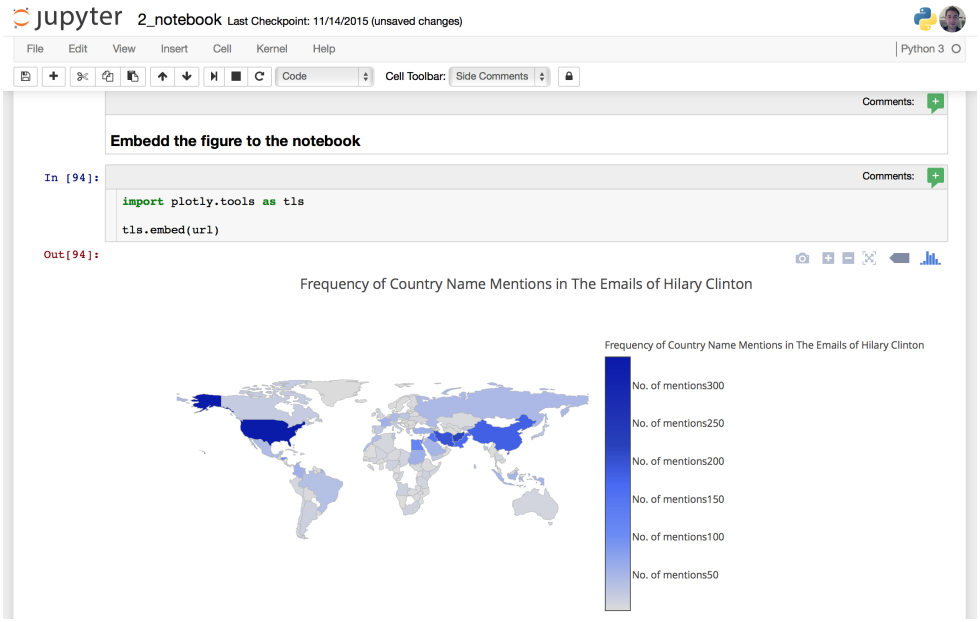

Figure 1. An example of a Jupyter Notebook showing a map plot using plotly. On the right side of each cell there is a comment section.

The exact procedure and structure of the evaluation is explained in more detail in the evaluation section.

RELATED WORK

This study draws on the theoretical framework of Coordination Theory proposed by Malone and Crowston (1994). The Coordination Theory aggregates work done on the coordination topic from a variety of fields and can be used to explore how software designed to support groups of people working together might improve their performance. This theory fits our study, as it differs from other organization studies by conceptualizing dependencies between tasks, rather than individuals or units, and focusing on a need to coordinate rather than on the desired outcome of coordination [**kevincrowston2006ten**, 9]. This approach has the advantage of making it easier to model the effects of reassignments of activities to different actors. Therefore, identifying dependencies and coordination mechanisms offers special leverage for redesigning processes [10]. For example, a common coordination problem is that certain tasks require specialized skills, therefore limiting the amount of workers that can efficiently process them. This dependency between a task and a worker suggests identifying and studying such common dependencies and their related coordination mechanisms across a wide variety of organizational settings. To overcome these coordination problems, workers must perform additional work, which Malone and Crowston (1994) called coordination mechanisms. For example, if a particular expertise is necessary to perform a particular task, then a worker with that expertise must be identified and the task assigned to them [**kevincrowston2006ten**].

Collaborative data analysis might be seen as adjacent to distributed software development. Despite the evolution of sophisticated collaboration and software engineering tools, coordination continues to be challenging in software development. Multiple studies mention lack of coordination as the reason behind delays, overspending and functional flaws in software projects [1, 3, 4]. Since software projects are large and globally distributed it is important to understand how to coordinate and support the effective communication among team members. Therefore, we address the theory of Distributed Cognition theory, which reflects on a phenomenon that emerges in social interactions as well as interactions between people and structure in their environments. This perspective highlights three fundamental questions about social interactions: (1) how are the cognitive processes we normally associate with individuals transformed in a group of individuals, (2) how do the cognitive properties of groups differ from the cognitive properties of the individual members of these groups, and (3) how are the cognitive properties of individual minds affected by participation in group activities.

The theory assumes three kinds of distributed cognitive processes; processes may be distributed across the members of a social group, cognitive processes may be distributed in the sense that the operation of the cognitive system involves coordination between internal and external (material or environmental) structures, and processes may be distributed through time in such a way that the products of earlier events can transform the nature of later events [6]. Distributed cognition theory has gained popularity in multiple communities. HCI scholars relied on it as a theoretical foundation for understanding interactions among people and technology and addressing a complex networked world of information and computer-mediated interactions [5]. Scientific communities have received special attention because the work of science, similarly to the assumed collaborative data analysis, is fundamentally cognitive and distributed. Among the studied aspects are how the distribution of cognitive activity within so-

cial networks of researchers and coauthors accounts for much of the work of science, and how scientific facts are created by communities in a process that simply could not fit into the mind of a single individual [7]. Additionally, distributed cognition has been viewed in the light of software design, arguing that social context and the artifacts present in the environment result in a collaborative cognitive system that goes beyond individual cognition [11].

DISTRIBUTED DATA ANALYSIS FRAMEWORK USING JUYPTER NOTEBOOK

After having analyzed different frameworks and tools for addressing the collaborative data analysis requirements, the decision was taken to use and test Jupyter Notebook. According to [**ipython-wiki**] Jupyter "is a command shell for interactive computing in multiple programming languages, originally developed for the Python programming language, that offers enhanced introspection, rich media, additional shell syntax, tab completion, and rich history". Using Jupyter, researchers can "capture data-driven workflows that combine code, equations, text and visualizations and share them with others" [**gitipy**].

The following features had a decisive impact on the decision to use this tool:

- A browser-based notebook (see Figure 1)with support for code, text, mathematical expressions, inline plots and other rich media.
- Although initially designed for Python, it is language agnostic and provides the ability to be extended with additional intepreters such as R, Ruby and others.
- Support for the interactive data visualization toolkits required in data analysis.

Jupyter-Drive

One of the extensions used to enable collaboration on Jupyter was a project called "Jupyter Drive" [**gitjup**]. "Jupyter Drive" is a Jupyter Notebook extension that allows Jupyter to use Google Drive for file management. When activated, users need to authenticate using OAuth2.0 to their Google Drive account. If the authentication is successful, they will have access to all their Google Drive contents in the Jupyter Notebook application over a web interface. This approach has multiple advantages. The first advantage is that the Jupyter server runs on a central location and can be accessed by all of the contributors/users using the web interface. Each user will see only their own Google Drive contents and will be able to create and execute notebooks on the central Jupyter server. The second main advantage is that we can use of the collaboration features provided by Google Drive, such as sharing notebooks, adding different sets of permissions to the Jupyter notebooks etc. The third advantage, although not directly visible for end users, is the ability to use Google's REST API's for managing content and orchestrating Jupyter projects on Google Drive. This is not provided by the "Jupyter Drive" extension and has to be developed as a separate component.

Collaboration extensions

As mentioned above, the "Jupyter Drive" extension provides the baseline of our Jupyter Notebook collaboration platform. However, additional features are required and have been developed as part of our collaborative data analysis framework:

- Workflow for creating a data analysis/mining project and distributing tasks to different workers (see Figure 2).
- Ability to manage projects.
- Ability for users / collaborators to annotate notebooks. For this, a commenting function is required.
- Ability to merge all notebooks of a project into a master notebook which can run all the different distributed steps in one run.
- Ability to have an iterative collaboration process.

These additional capabilities, including the workflow for the project definition, and the management page, were developed in a prototype as part of the project this paper refers to.

The project creation workflow is designed to allow a data analysis expert, who will act as the project owner, to define a project and distribute assignments to workers in a top-to-bottom approach.

In our framework, an *action* is the smallest unit into which a task can be split, and is described by its name, input and output. An example of an action would be *loadDFFromCSV*, which receives as input the path of the CSV file and returns a dataframe. The project owner can search or filter for *actions* from a default taxonomy and group them into assignments. For this prototype, the default taxonomy used was the "Catalogue of Methods in Data Pre-Processing" created by AixCAPE e.V. [2]. Splitting tasks into small actions, especially the pre-processing part, allows the project manager to group and distribute them to non-expert workers, where we assume that each individual has a small part of the knowledge required for the completion of the project.

We will now refer to the end result of this extension process as the "Collaborative Jupyter Notebook" environment. The prototype was then used in several experiments to evaluate our two hypothesis.

EVALUATION STUCTURE

In the context of this paper, we evaluate the platform on several levels, starting from two hypotheses.

The first hypothesis states that the pre-processing part of a data analysis project can be decomposed in small enough tasks such that they can be performed by non-expert workers. In order to test this, we ran experiments with crowd workers and compared the results to those of experts. A group of students also tested this hypothesis on our Collaborative Jupyter Notebook. We used the crowdsourcing platform Upwork to source the non-expert data scientists. As a first step, we defined a full "vision" of task decomposition criteria as follows: based on worker attributes, based on task attributes and based on external factors.

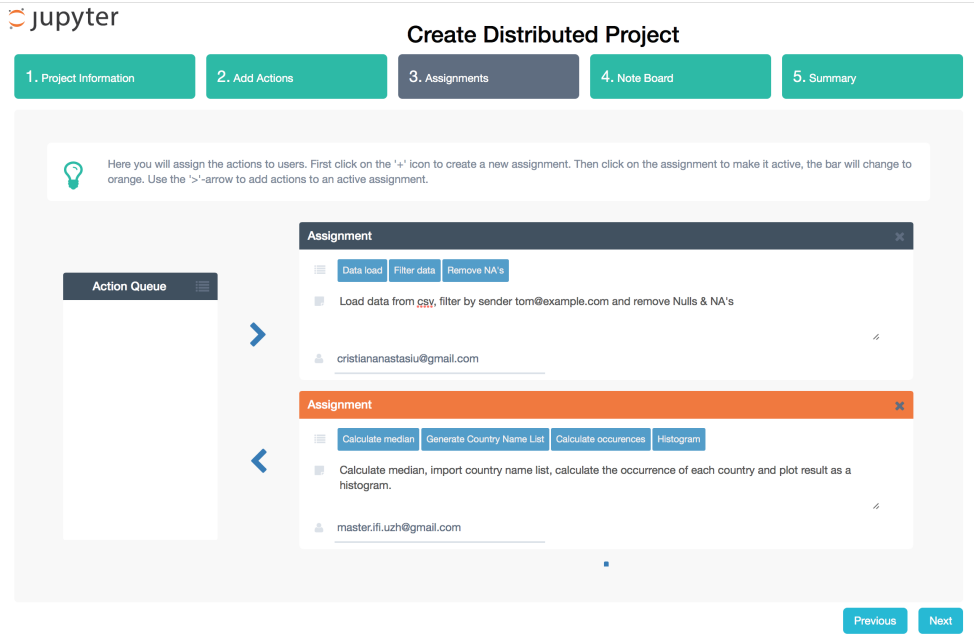

Figure 2. Project creation workflow

The method of splitting the project by worker attributes can further be divided into four areas: expertise (skills pertaining to the subject matter, skills the individuals have listed on their Upwork profile, hours worked, work history, test scores for specific skills on Upwork), ranking (experience: beginner/intermediate/expert, success rate on past projects, recommendations), abilities (cognitive abilities, communication, etc.), preferences (time availability, type of project: scraping vs. modelling, own statements of preferences). For a summary of the splitting methods see Table 1.

When dividing the project based on task attributes, we take into consideration two categories: dependency (sequential tasks vs. parallel) and resources (i.e. concurrent access of tasks on same data).

The third decomposition method uses external factors as criteria, for example availability (timezone dependency, availability of workers at any given time).

In our experiments, we performed the tasks distribution based on the subjects expertise and preferences, and on the tasks dependencies.

The second hypothesis states that the proposed team with a mixed level of expertise performs as well as standard expert-based projects.

In order to test this hypothesis, we analyzed two elements of this project.

First, we evaluated our collaboration tool and workflow. The tool was used by a group of crowdsourced non-experts on data analysis projects, in order to see whether the same quality of results can be achieved using our tool, as using traditional development tools that experts used. We look at

Task decomposition methods		
Factor	**Criteria**	**Sub Criteria**
Subject	Expertise	Skills Hours worked Project History Work History Upwork tests
	Ranking	Entry / Intermediate / Expert Success rate
	Abilities	Cognitive abilities Communication Personal
	Preferences	Time availability Type of projects Own statements
Tasks	Process	Sequential Parallel
	Resource dependency	
External factors	Availability of workers	

Table 1. Table shows the different criteria which can be used to perform task decomposition.

whether the tasks can be decomposed with relative ease, whether the projects can run from beginning to end and we compare feedback on the workflow from the participants.

Second, we looked at the results of the projects. We check the quality of the results by comparing the two data sets, then running a correlation. We also run a t-test to evaluate the statistical significance of the difference in results for one of the experiments.

For the project selection, we used the data science platform Kaggle. As described on Wikipedia, "Kaggle is a platform for data prediction competitions on which companies and researchers post their data and statisticians and data miners from all over the world compete to produce the best models". We assume that these projects were solved by data science experts, and use their results and methodology for comparing the results of the crowd workers. The following criteria where used to select the projects:

- Project should be written in R or Python, as these are major languages for data analysis [12].
- Project should contain a relatively large pre-processing part.
- Project should not be too complex, tasks should be delivered in 3-5 days and be equivalent to 6-10 hours of work. Therefore, we selected projects with less than 10 versions.
- Ability to split the project in 3 to 6 parts.
- Due to our hardware limitations, the data set required by the project should be less than 2gb.

EXPERIMENTS

In the context of this paper, we performed experiments with three projects selected from Kaggle, and one project chosen by the supervisor of a group of students at the University of Zurich as part of a seminar for Masters' students. The projects are composed of existing code and data sets taken from Kaggle, an online community of data scientists. On Kaggle, members post solutions to data science problems in the framework of competitions on selected topics. They interact with each other, are able to rate their solutions, make suggestions and create new projects that build upon existing ones. When choosing Kaggle, we assume that the members are experts and the project results are assumed to have expert-level quality.

We chose three projects:

- Hillary Clinton's Emails

  https://www.kaggle.com/ampaho/hillary-clinton-emails/foreign-policy-map-through-hrc-s-emails/code

- Earnings Chart by Occupation and Sex

  https://www.kaggle.com/wikunia/2013-american-community-survey/earnings-by-occupation-sex

- Reddit Sentiment Analysis

  https://www.kaggle.com/lplewa/reddit-comments-may-2015/communication-styles-vs-ranks/code

The three projects were chosen to meet the criteria previously defined, and such that we have different complexity levels for the tasks. Earnings Chart is the least complex, Hillary Clinton's Emails is of medium complexity, and Reddit Sentiment Analysis is slightly more complex. We also made sure to choose projects which can be split into several tasks, specific to the needs of each project. These assignments contain combined elements of loading, cleaning and transforming data, as well as data mining and visualisation.

While the project manager provides some guidelines on how they imagine that the tasks should be solved, it is up to the workers to decide on the actual implementation and on the packages and libraries that need to be used. Also, workers do not have to strictly follow the actions and the sequence defined by the manager. Actions can be added or skipped, depending on the implementation chosen by the worker.

Earnings Chart by Occupation and Sex

The aim of this first project is to create a chart showing the earnings of the population by occupation and gender, using a subset of the latest US census data set from the year 2014. The original project on Kaggle uses a data set from 2013. Also, the original Kaggle project analyzes 24 categories, while in our project we reduced the categories to 11. The chart focuses on the following categories: Management, Business, IT, Engineering, Science, CommunityService, Legal, Education, Arts Sports, Healthcare, Military.

We were running multiple parallel projects on our system and were constrained by the storage and compute capacity. Thus, we decided to reduce the data set and created a subset of the US 2014 census data based on random data sampling. The US census 2014 data set can be downloaded from http://factfinder.census.gov/faces/nav/jsf/pages/searchresults.xhtml?refresh=t

This project was split in three assignments. The first task centered around data loading and cleaning with the primary goal of identifying the right industry code ranges and sub-setting the data. It consists of five *actions* - "Data Load", "Identify Occupation Industry Codes", "Subset data", "Remove NAś", "Save temporary csv". The output of this task should be a csv file containing the information about the population working in the 11 industries relevant for our chart.

The second task focuses more on the data transformation aspect and has only two simple *actions* - "Mean" and "Save temporary csv". The output of this task is expected to be an aggregated data set containing the mean earnings of men and women per industry of interest.

In the last assignment the user has to plot the data as a bar chart diagram in descending order, showing the distribution of men and women per industry and their average earnings. It consists only of one action - "Bar Chart" - and should produce as output a bar chart similar to the one in the Kaggle project.

Hillary Clinton's Emails

This project takes a look at the content of Hillary Clintons emails, which were released by Hillary Clinton herself in response to a Freedom of Information Act (FOIA) request, and produces a heat map of the countries which appear most often in the emails sent by Ms. Clinton. The data set for this project is available as a sqlite database or as .csv files and can be downloaded from the Kaggle web site https://www.kaggle.com/forums/f/15/kaggle-forum/t/16444/hillary-clinton-email-dataset.

The project was also split into three tasks each task was then assigned to a different worker. The first task focuses on data loading and cleaning, and consist of three *actions* - "Data load", "Filter Data" and "Generate Country Name List". The output of this task should be a cleansed subset containing only the emails sent by Hillary Clinton and a list of all the countries in the world and their alternative spellings and abbreviations.

The second task focuses on identifying countries in the email data set and contains two *actions* - "Subset" and "Calculate occurrences". The output of this task should be a country occurrence list, containing the number of times each country is mentioned.

The last task focuses on the visualisation part and consists of two *actions* - "Histogram" and "Heatmap". The output should be a sorted histogram and a heat map in form of a world map, similar to the output of the Kaggle project.

Reddit Sentiment Analysis

The purpose of this project is to create a chart showing which Reddit (www.reddit.com) comments receive the highest scores, based on the sentiment of the comment. Reddit is a large social network where users can submit content. The dynamic of the website is solely dependent on the number of votes that the content receives. The content or comment with the highest number of votes is shown at the top. Known for its general skepticism and sarcasm, it is interesting to look at how the comment sentiment and the votes are related, if at all. The split is made according to the three sentiment categories - objective, negative, positive. For this, we use the Sentiwordnet nltk package.

The initial data set for this project containing all the Reddit comments from the month of May in the current year (2015) was downloaded from the Kaggle web site (https://www.kaggle.com/c/reddit-comments-may-2015/download/database.7z) in an sqlite database dump format. Because of our storage capacity limitations on the server, we sampled 1% of the data (500'000 comments), exported it as csv file and made it available to the workers.

This project was split in three parts. The first assignment focuses on data loading, data cleaning and calculating the sentiment score for each comment. It consists of the following *actions* - "Data load", "Remove NA's", "Subset data", "Create NLTK helper functions", "Calculate sentiment score for each comment". The output of this task should be a data frame with the 3 sentiment scores (objective, positive, negative) for each comment.

In the second task, the worker has to subset the data (extract only the representative comments for each sentiment category) and calculate the average ranking for each category. The tasks consist of three *actions* - "Average Ranking", "Filter data", "Aggregate Data". As output of this task, we expect an aggregated data set which can be used to plot the information in a chart.

The third task consists of plotting the results in a bar chart. It contains only one *action* - "Bar chart".

Student group project

Additionally, we also put the prototype at the disposal of a group of four students from the Marketing Research department within the University of Zurich. The project is done as part of a seminar and is supervised by a PhD student, which we assume is an expert in data analysis. The goal is to identify influential users on social media platforms and respond to the following research question: Which are the few users that could shape the opinions of many others?. The relevance and implications of this project are two-fold. First, in practice, the project is meant to develop a toolbox that helps firmly identify influential users. Second, in theory, the aim is to better understand the assumption and scenarios where different methods of identifying influencers can be applied.

The project covers the complete data analysis spectrum, from data loading and preparation, to data analysis, interpretation and project evaluation. The information that composes the data set is taken from multiple social media platforms and includes the number of posts per user per thread, the distribution of the number of posts per user, the distribution of the number of posts per thread per user, and the distribution of each measure. All of the pre-processing of this data, as well as the analysis and interpretation, will be done using our platform, all the while splitting the tasks into several fragments, each assigned to a non-expert student.

Although the student project will finish after the submission deadline of this paper, we have include the feedback of the supervisor and that of one of the students in the validation of our hypotheses and in the recommendations for future work.

Worker demographics

In order to complete the experiments, we had a team of three people work on each project. We sourced the crowd workers on the platform Upwork (www.upwork.com), which connects companies with a global community of freelance workers with various skill sets. The projects used in our empirical analysis require basic to intermediate knowledge of Python, data pre-processing and some data visualization, so when picking the workers we made sure that they fulfilled this minimum requirement. We selected three people for each of the three projects and payed each of them $40. At the end of the project, they were asked to complete a survey. The survey consists of three parts: the first is information on demographics and their prior experience, the second is an evaluation of their respective projects, and the third contained an assessment of the tool. The exact results of this survey can be found in the Appendix.

Age, Gender and Language

Of these 9 people, 2 are female, and all but one participant are between 25-32 years old. Only one of the projects contained two people with the same native language (Russian). The native languages of the workers are: Arabic, English (Malaysian-born person), Italian, Macedonian, Russian (two people), German and Ukrainian (two people).

Education
66% of the participants have a Masters degree, the rest have a Bachelor's or no university degree. There was only one individual who did not have studies in the Science, Technology, Engineering or Mathematics (STEM) field. 44% have studied Computer Science.

Occupation
Currently, 5 of the 9 people are working in computer science, one in data science, one in optics and photonics, and two are looking for a job. 5 of the 9 workers had only been active as online freelancers for 6 months or less, only one worker had more then 1 year experience as a freelancer. The crowd workers spend an average 11.2 hours a week working as freelancers.

Technical skills
The participants were asked to rate their Python skills on a scale from one to five. One of them rated their experience a 1, all of the rest chose a 3 or a 4, amounting to an average of 3.2 out of 5.

They also rated their data analysis abilities. All workers chose a 3 or a 4, compounding an average of 3.4 out of 5, which was in alignment with our criteria selection, as we generally wanted to have non-experts assigned for the tasks.

Also, 7 of the 9 participants have stated they have worked with IPython/Jupyter before.

Data analysis experience
One of the most interesting findings for this project was their experience in the data analysis field. When asked about their experience, the answers varied across the board:

- One persons background was in data analysis in R and Stata during her studies in Economics, as well as writing a Masters Thesis on Statistics.
- The second person has similar experience with data analysis, but has also been working with Python for about a year.
- The third person is experienced in Python with the following packages: numpy, pandas, nltk, scikit-team, ggplot.
- Our fourth participant has some experience on Python and with the data science platform Kaggle. By looking at his Kaggle profile we noticed that he does not have any own scripts published, but participated as part of a team in a public competition which achieved being in the top 25% of one project.
- The fifth worked on a Masters project on machine learning and used Python, including the numpy and scipy packages.
- Worker number six was completely self taught through home projects with a family budget.
- The seventh participant has worked in Visual Basic for data analysis scripts on Microsoft Excel, has knowledge of R, having worked with medical data, and produced a data visualisation project in IPython for genetic research.
- Our eight crowd sourced worker seems to be the most experienced one in data analysis. He has ten years work experience with experimental and computational data. At first, he worked with MatLab and C, but he now prefers Python.
- The ninth participant has experience doing post-processing of Finite Element results in Python and Excel in her engineering work. She also wrote a data mining tool to visualize algorithms for creating subsets.

Overall, most of the workers have some experience in the data analysis field, either by experimenting at home or through classes in the University. Only one of them could be regarded as a truly experienced data analyst, with 10+ years of experience. Also, in terms of tools, most of them worked with Python and Excel for data analysis, only two mentioned having experience with R. The result of this is also reflected by the question regarding their data analysis skills, where the average was 3.4 out of 5.

RESULTS

In this section, we take a look at the results of the three experiments performed in the context of this paper. We test the two hypotheses and produce an evaluation of our tool.

Hypothesis evaluation

H1: *The pre-processing part of a data analysis project can be decomposed in small enough tasks such that can be performed by non-expert workers.*

To test the first hypothesis, we will first look at the first part of the statement and check whether it is possible to decompose the tasks into small enough sub-tasks. Based on the results of the experiments and the student group project, we can affirm that this can be achieved. It was possible to split all the projects into *actions*, even by the author, who is a non-experts in the field of data analysis. Also, all of the Upwork workers were able to successfully complete their assignments. They rated the complexity of their assignment an average of 2.1 out of 5. Another interesting fact is that the project complexity was rated slightly higher than the assignment complexity, with 2.3 out of 5. This means that we were able to divide a slightly complex project into several less complex tasks.

Regarding the distribution of the assignment, the tool allows only a top-to-bottom approach. The tasks were distributed mainly based on the subjects' expertise (e.g. some of them did not have any data visualisation expertise) and/or on the subjects' preference.

Also, the feedback received from the expert supervising the student project concludes that tasks can be split into small enough assignments that can be performed by students. He was able to distribute the project among the students, stating

Workers' feedback on their skills and experience				
Project	**Worker**	**Python**	**Data analysis**	**Experience**
Earnings Chart by Occupation and Sex	#1	4	3	Machine-learning project for Master's in Computer Science, Python
	#2	3	4	Experimental and computational data permanently for about 10 years
	#3	3	3	Post-processing of Finite Element results with Python and Excel
Hillary Clintons Emails	#1	1	3	University exams, some Stata and R
	#2	3	4	Python for about 1year, practical knowledge through projects
	#3	4	4	Python experience, numpy, pandas, nltm, scikit, ggplot
Reddit Sentiment Analysis	#1	3	4	One year in office, some kaggling, Python
	#2	4	3	Home projects with family budget
	#3	4	3	VBA and Excel, intermediate expertise in R, geo-visualization project in IPython for genetic research

Table 2. Table shows the feedback received from the Kaggle freelancers on their expertise with Python and Data Analysis. The rating used a scale from 1 to 5, with 5 meaning expert.

that it was easy to decompose the tasks using the *action* taxonomy or by creating custom *actions*. Managing the project was also easy according the expert, because the tool allowed him to check the status of each assignment in real time and intervene if needed.

The second part of the hypothesis looks at the workers performing the sub-tasks, where we will try to determine whether non-experts should be able to perform them. Based on the feedback form received at the end of the experiments and their Upwork profile, we can state that out of the 9 participants from the freelancer platform, only two of them can be considered experienced data analysts (see Table 2). The rest are either beginners or intermediate the field of data analysis. The participants also stated that there is no prior expert knowledge required for the projects. The required knowledge and Python expertise was rated an average of 2.8 out of 5, which tell us that they also did not find it a requirement to be an expert in Python.

Based on all of these results, we can conclude that the hypothesis H1 is true.

H2: *The proposed approach of teams with mixed level of expertise leads to results comparable with expert teams.*

The first thing we are going to look at is the skill level in data analysis and Python, and the experience of the participants in each project. The information was assembled using the feedback forms at the end of the experiments and can be viewed in table 2. As shown in the table, the workers have a mixed level of expertise and skills.

To verify if the teams with mixed level of expertise perform as well as standard expert-based projects we compared the results of each empirical test.

Earnings Chart by Occupation and Sex

This was rated as the easiest project of the three. The goal is to create a chart showing the earnings of the population by occupation and gender. The main focus is on finding the right occupation categories and sub-setting the data accordingly. As mentioned in the Experiments section, the project used a random 1% sample of the US census data from the year 2014. In order to compare the two results, we ran both implementations (Kaggle results and our team's result) on the same data sample. The team of non-experts managed to successfully finish the project and the result of the team was similar to the one on Kaggle. The correlation coefficient between the two results was 0.8 a high correlation.

The differences in the results can be traced back on the method the two implementations perform the data subsetting. Each occupation in the data set is identified by a code. The 11 categories used in the project are quite generic, so it is the user's task to find the occupations which belong to the respective category. The implementation on Kaggle identifies only one occupation for each category, while the Upwork team's implementation aggregates multiple occupation codes under the same category. As an example, for the category "Management" the Kaggle project uses the occupation code 430 which represents "MGR-MISCELLANEOUS MANAGERS, INCLUDING FUNERAL SERVICE MANAGERS AND POSTMASTERS AND MAIL SUPERINTENDENTS", while the Upwork team's implementation uses the codes 0 to 499 which includes all management occupations like "MGR-CHIEF EXECUTIVES AND LEGISLATORS", "MGR-GENERAL AND OPERATIONS MANAGERS" etc. This leads to differing results on some of the categories.

Hillary's Clinton emails

The goal of the project is to create a heat map based on the frequency the countries are mentioned in the emails sent by Hillary Clinton. The non-expert team managed to successfully finish the project. The output of their result is similar to that of the Kaggle project (see figure 3).In both implementations, the heap map is based on a country occurrence list. We compared the two results by calculating the correlation coefficient between the two occurrence lists. The correlation coefficient is 0.72, which represents a high correlation.

The difference in the results lies in the way the two implementations identify the countries mentioned in the emails. The project on Kaggle uses a Python database called *countrycode*

which contains all the country names and their ISO2C and ISO3C codes, and identifies the countries in the email bodies using regular expressions. The implementation done by the non-expert team uses a different approach. It identifies the countries using the nltk package and named entity recognition (NER). This implementation identifies 186 countries, compared to the Kaggle implementation which identifies only 90. The difference in the algorithm is also reflected in the execution time, which is much faster in the Kaggle implementation.

Reddit Sentiment Analysis

The project's goal is to create a chart showing which Reddit comments receive the highest scores, based on the sentiment of the comment. Three sentiment categories were defined - objective, positive and negative. As in the previous project, we decided to use a random 1% sample of the May 2015 data set, due to our storage limitations. Both implementations, the Kaggle and the non-expert team, were tested and compared on the same data set. The results are very similar, as visible in figure 4 - the average ranking scores for the posive, negative and objective comment categories are 6.18, 6,78, 5.96 in the Kaggle project, and 5.75, 6.22, 6.34 in the Upwork project done by the non-expert team.

We also compared the ranking values in each sentiment category by performing a t-test on the results of the two projects. The outcome of the calculated t-statistic and the p-value is as follows:

- Positive
  (1.18, 0.23)
- Negative
  (1.61, 0.10)
- Objective
  (-0.62, 0.52)

In all of the cases the p-value was over 0.05, meaning that there is actually no difference between the ranking means in each sentiment category. We also checked if the ranking scores follow a normal distribution, and this was the case in all of the six data sets.

Regarding the implementation, both projects used the Sentiword nltk package. However, the classification of the comments into one of the three sentiment categories was done differently. The Kaggle project is classifying the comments by selecting only the comments with values above average (top quartile or top 3/8) for each sentiment, while the project done by the non-expert team first normalizes all sentiment scores (through division by mean) and only then classifies every comment. Nevertheless, the results are almost identical.

Based on the results above, we can conclude that the hypothesis H2 is also true.

The tool evaluation

Assignment Completion

At the end of the project, we asked the participants in the experiments to evaluate the tool and how they were able to use it to complete their project. All participants said they were able to solve their assignment using the platform. Only 3 out of the 9 participants said they also used another external or local tool to work on their assignments. We can therefore conclude that the platform was efficient in helping them solve their individual assignments and bring the project to the end result.

Communication

66% of the crowd workers said they communicated more than 3 times with their project colleagues during their assignments. 5 of the 9 participants claimed they communicated well with their colleagues, but did so using other tools; and another 4 claimed the tool was not essential to their communication. This means that there are improvements to be made on using the tool for communication among team members. One of the most requested features by the participants in the experiments was a system of notifications, which would push emails or another sort of alert to the other members when there is a comment on their notebook, or when they are mentioned. Also, perhaps a live chat feature and direct communication method with the project owner can be helpful to improve live communication on the projects. Thus, the conclusion is that there is further work needed on the communication ability within the platform.

Helpful features

We also asked the freelancers to evaluate the current features of our platform. Five of the participants said the most helpful feature was being able to see what the others were doing and commenting on their work. This was also similar to the feedback received by the student group supervisor. One person thought the ability to assign responsibility to different team members was most helpful, while another thought the sticky note was helpful. Another one of the freelancers rated the live code feature as most helpful. Two participants mentioned that being able to merge the notebooks was the most useful feature.

Privacy

Another important aspect mentioned by the supervisor of the student project was privacy. Even if for the group workers privacy is not really an issue, as they are just interested in loading the data and working with it, restricting the access only to members inside the project group was essential for the project manager. As the data set can contain sensitive or costly information, the framework needs to be able to protect such information from unwanted access. In this prototype, each group had its own temporary folder to upload data, but this did technically not prohibit the different groups from looking into the other's files. Although download was not enabled, they could have just printed out the data in the notebooks, which could be an issue if sensitive data were used.

Suggestions for improvement

When asked which features they think are missing on our platform, 6 out of 9 participants mentioned a notification system or better communication as their primary request for enhancement. One person mentioned a separate notebook file, in addition to the shared file, where one can add notes related to that particular notebook, as well as a means of communicating directly with the project manager perhaps through a

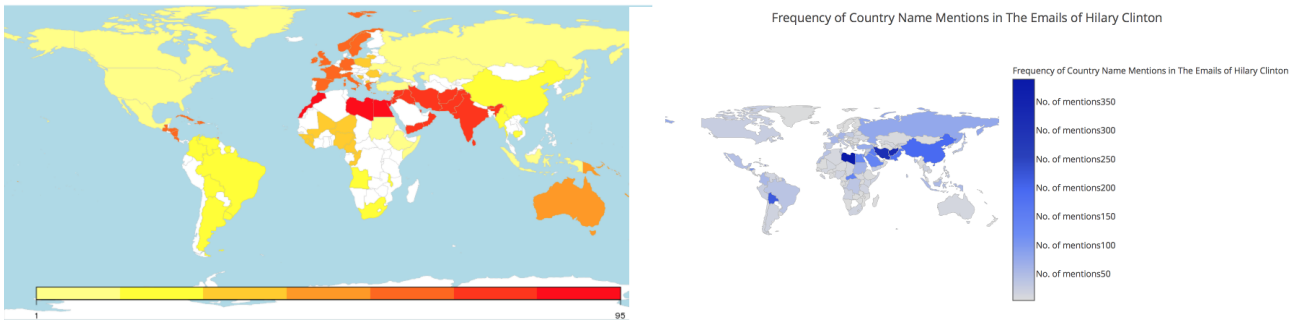

Figure 3. Plot outputs on the Hillary Clinton heat map project. The plot on the left represents the output of the Kaggle project, the one on the right the output of the non-expert team.

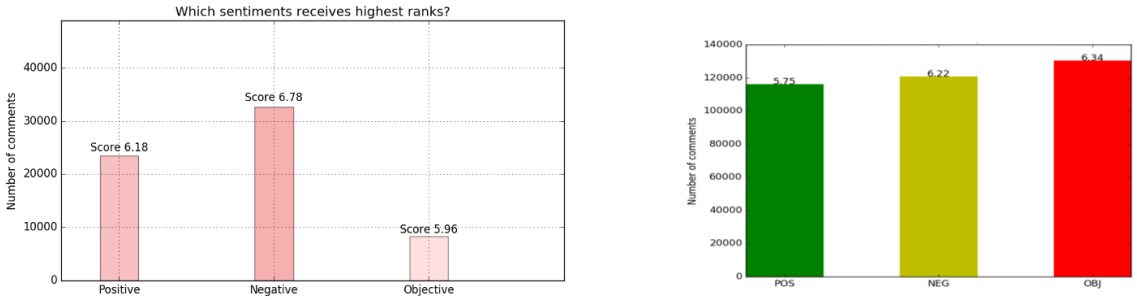

Figure 4. Plot outputs on the Reddit Sentiment analysis projects. The plot on the left is the output of the Kaggle project, the one of the left is the output of the non-expert Upwork team

manager notebook. Two freelancers mentioned that an improvement would be to integrate variables and functions directly from other notebooks or the ability to work in other kernels, as long as the data is synchronized. One person also suggested improving stability without the Authorization limit expired message (which was actually identifie as a bug in one of the external Jupyter extensions used), adding a more easily accessible way to see colleagues notebooks and adding a pre-merge feature, that will merge the notebooks in a temporary master notebook. Allowing the project members to create a ”merged” master notebook on their own was also a request from some of the participants. Also, one other feedback giver mentioned improving the web interface on all browsers. The crowd workers rated their overall experience working with our tool an average 3.9 out of 5.

Other suggestions for improvement received by the student group and their supervisor are:

- Email notification for the project owner. Although the workers receive an email notification about how to add the project to their Google Drive and access their assignment, there is no such notification for the project owner. One of the suggestions was to add notifications for the owner as well.

- UI bug fixes and enhancements, such as changing the way actions are added to assignments, fixing scrolling issues, adding the project name in the ”Delete” popup, Autocomplete for inputs, having the option to save the wizard’s state or being able to view the action’s details, such as description, input and output, when moving the mouse of the action element.

- One important aspect that was mentioned was the ability to edit the project after it was created. Changes are needed when tasks need to be re-assigned to other workers or when new assignments have to be created. This feature was completely missing in this prototype; future versions would need to take this into consideration.

- Another suggestion was to separate the *action* creation process from the project creation process. Actions can be added to the taxonomy only when creating a project through the wizard.

- Cell size restriction of 50 lines should be defined as an editable project attribute.

- User guide or manual.

CONCLUSION

The goal of this paper is to contribute to the data analysis field by discussing and implementing a framework that allows for collaborative data analysis in a crowd sourcing environment. We have created an online platform that permits the efficient splitting of the data pre-processing part of a project into several small tasks. These tasks can then be assigned to crowdsourced workers with little to no expertise on the subject matter of the overall project, but who can solve smaller, simpler assignments.

We also created a working prototype showing that Jupyter Notebook can be used as a tool in distributed environments for performing data analysis projects with crowd workers. The prototype, including the Jupyter Notebook extensions, can be used as a starting point for further research and experiments.

We tested our tool with three projects, taken on by teams of non-experts, as well as with a student group project. Overall, our tool provides the ability to decompose the data preprocessing parts of projects into small enough tasks, such that can be performed by non-experts. Also, the proposed method of teams with a mixed level of expertise perform as well as standard expert-based project. The collaborative aspect that our tool introduces seems to be appreciated by the participants in the empirical testing. There are, however, several improvements that can be made to our tool, such as bug fixing and improving communication and the notification system. Further testing is needed with larger and more heterogenous groups of participants, as well as with more complex projects and tasks.

As a conclusion, we can state that our tool lays the groundwork for an efficient platform that can be used for collaborative data pre-processing and analysis. We can also state that non-expert users can be successfully included in more complex data analysis projects, performing as a team as well as experts in the field.

FUTURE WORK

Including non-experts in complex problem solving

This paper shows that it is possible to have non-expert data analysts contribute to data analysis projects and perform as a team overall as well as experts in the field. The idea of having non-experts contribute to complex problems can be further expanded. The projects we experimented with demonstrate that technical crowd workers could be included in performing tasks such as creating filtering rules using regular expressions (as for the Hillary Clinton email project) or generating lists of alternative spellings, namings and abbreviations for specific categories, such as luxury brands or car models etc. For this type of work, no technical expertise is needed and the tasks can be performed in environment such as Google Docs or Excel. This kind of work can be easily arranged over Amazon Mechanical Turk.

Empirical testing

When expanding on the current platform, it is important that there be further empirical testing on it. One example would be testing its collaborative features with a large group of people (4 and up), as opposed to only 3, as we tried it. Also, the crowd sourced groups we worked with are somewhat culturally homogenous. It would be interesting to test the tool with a more diverse set of people: for example, with participants from India, China, USA, Sub-Saharan Africa, Central and South America, etc.

Another aspect which can be analyzed in more detail is the cost factor. In this experiment we managed to prove that similar results can be achieved by non-experts compared to experts, however we have not analyzed the cost implications in detail. It would be of interest to verify if a group of non-experts can solve a complicated data analysis problem more cost effectively than one or more experts.

Projects & Data sets

Further testing of the platform can be conducted most efficiently with some more complex projects and more challenging data sets, which can be divided into more separate tasks. The data sets and sample projects used in these experiments were rated as relatively easy (2.3 out of 5), so it would be interesting to see if a similar success rate can be achieved with more complex projects.

New features and bug fixing

There are several new features that can be introduced in further work expanding this tool. First off, as suggested by most of the participants in the empirical testing of the platform, there is a need for a notification system for new comments. Without this feature, we have seen that there is a lag in the response to comments and implementation of suggestions. Another important aspect that needs to be further refined is privacy. For this experiment we used the local server's storage, which was accessible through a generic ftp account. This did not provide a high security level, which can be an issue for projects using sensitive data. Also, the tool must be tested on several different browsers to ensure that it provides the same seamless experience across the board.

ACKNOWLEDGMENTS

We thank all of the University volunteers led by Radu Tanase (Andrea Bublitz, Claudia Wenzel, Magnus Liedloff) who provided valuable feedback for the platform, all freelancers who participated in the experiments and in the evaluation, PhD student Michael Feldman, who provided close assistance and extremely helpful ideas and comments throughout the project, and Prof. Abraham Bernstein for supporting this research project with valuable ideas.

References

[1] J Alberto Espinosa et al. "Team knowledge and coordination in geographically distributed software development". In: *Journal of Management Information Systems* 24.1 (2007), pp. 135–169.

[2] AixCAPE e.V. *Data Processing Compendium - Workflows for Knowledge Exploitation in the Process Industries*. 2013. URL: http://dataprocessing.aixcape.org/index.php/Main_Page#A_Catalogue_of_Methods_in_Data_Pre-processing.

[3] Lile Hattori and Michele Lanza. "Syde: a tool for collaborative software development". In: *Proceedings of the 32nd ACM/IEEE International Conference on Software Engineering-Volume 2*. ACM. 2010, pp. 235–238.

[4] James D Herbsleb and Deependra Moitra. "Global software development". In: *Software, IEEE* 18.2 (2001), pp. 16–20.

[5] James Hollan, Edwin Hutchins, and David Kirsh. “Distributed cognition: toward a new foundation for human-computer interaction research”. In: *ACM Transactions on Computer-Human Interaction (TOCHI)* 7.2 (2000), pp. 174–196.

[6] Edwin Hutchins. “A cultural view of distributed cognition”. In: *Unpublished Manuscript, University of California, San Diego* (1989).

[7] Edwin Hutchins. “Distributed cognition”. In: *International Encyclopedia of the Social and Behavioral Sciences. Elsevier Science* (2000).

[8] SB Kotsiantis, D Kanellopoulos, and PE Pintelas. “Data preprocessing for supervised leaning”. In: *International Journal of Computer Science* 1.2 (2006), pp. 111–117.

[9] Thomas W Malone and Kevin Crowston. “The interdisciplinary study of coordination”. In: *ACM Computing Surveys (CSUR)* 26.1 (1994), pp. 87–119.

[10] Thomas W Malone et al. “Tools for inventing organizations: Toward a handbook of organizational processes”. In: *Management Science* 45.3 (1999), pp. 425–443.

[11] George Mangalaraj et al. “Distributed cognition in software design: An experimental investigation of the role of design patterns and collaboration”. In: *Mis Quarterly* 38.1 (2014), pp. 249–274.

[12] Gregory Piatetsky. *Four main languages for Analytics, Data Mining, Data Science*. 2013. URL: http://www.kdnuggets.com/2014/08/four-main-languages-analytics-data-mining-data-science.html.

sample.bib