Mining the Insights of Stack Overflow Developer Survey

Mohammad Imrul Jubair
CSCI 5502
University of Colordao Boulder
Boulder, USA
mohammad.jubair@colorado.edu

Mohsena Ashraf
CSCI 5502
University of Colordao Boulder
Boulder, USA
mohsena.ashraf@colorado.edu

Cornelius Onimisi Adejoro
CSCI 5502
University of Colordao Boulder
Boulder, USA
adejoro.cornelius@colorado.edu

Abstract

There has always been a high demand for software development and tech-related jobs in market and the programming community plays a vital role to comprehend this employment sector. While speaking of a tech community, Stack Overflow is the most popular one since programmers and developers from diverse categories participate here very frequently. Over the past few years, Stack Overflow has been conducting yearly surveys on its users to collect information about how they learn and upgrade themselves, which tools they are utilizing, and what they enjoy in development. In this project, we applied an exploratory data analysis on this survey-based dataset—called the *Stack Overflow Developer Survey (SODS)*—to understand current state of the job market. For this purpose, we investigated correlations between factors and variables related the software development industry. We presented data visualization to compare and analyze those variables. We applied different data mining methods on SODS dataset—including frequent pattern analysis and classification techniques. In addtion, we attempted to find answers to few relevant questions and to test a hypothesis that articulates gender diversity in tech organizations. We believe, our findings will help job seekers to understand contemporary situations in tech-based job field. In addition, employers can also exploit our outcomes to examine their hiring strategies. We presented the lessons we learned from our investigation and these observation will work as strong references for advanced research in future on survey-based datasets.

CCS Concepts: • **Social and professional topics** → *Computing occupations.*

Keywords: Stack Overflow Developer Survey, Data Mining Techniques

1 Introduction

From 2021 to 2031—according to the report from *U.S. Bureau of Labor Statistics* [11]—it is anticipated that overall employment in computer and information technology occupations would expand by 15%, which is much faster than the average for all occupations and will add roughly 682, 800 new jobs. From this information, we can realize the demand and rise of technology related job in the market. In this era of technology, the world is witnessing tremendous digitization trends that appear to be on steady increase following continuous innovations, and the emerging application domain of these technologies. However, it has often been said that available software developers are still grossly inadequate; as we can observe from *IDC Market Perspective* report that says—the global shortage of full-time developers will increase from 1.4M to 4.0M for the period of 2021 to 2025; hence the full-time developer labor force will perform 90.8% capacity in 2021 and 84.9% capacity in 2025 [5].

1.1 Problem Domain

The job market is still extremely competitive and of interest to all prospective candidates. It is important to comprehend and analyze current employment condition, locally or globally, so that one can act accordingly. For instance, if a person X has skills in languages like *C++, Java* and *JavaScript* with coding experience of 11 years, what range of salary X could expect in the job market? However, it is very difficult for X to extract these insights, because the market is very broad, and the corresponding population is a complex field to study. One practical way is to deal a smaller sample group or community within this population and consider that for further mining. Without a question, Stack Overflow is today's most well-known community and a potential field for market analysis. It is an autonomous online Question and Answer forum. Stack Overflow's success is largely due to the engaged and active user community that collaboratively manages the site [9]. For the past few years, Stack Overflow (SO) has been surveying its members—known as *Stack Overflow Developer Survey (SODS)*[1]—to discover more about their learning process and progress, what technologies they employ, and what fascinates them. In SODS 2022, more than 70, 000 SO users from different background and experiences participated and the forum has been sharing publicly these responses as datasets for the past few years. Some sample questions are listed below.

- *Which programming, scripting, and markup languages have you done extensive development work in over the past year, and which do you want to work in over the next year?*
- *What is the primary operating system in which you work?*

[1] https://survey.stackoverflow.co/2022/

1.2 Contribution of our Project

In our project, we consider this survey dataset as a representative sample of the enormous tech-related community and want to investigate for intriguing insights. We plan to apply exploratory data analysis EDA) [12] that will help future developers better understand the industry. As part of this research, we conducted an exploratory data analysis on SODS to better comprehend the contemporary job market. In order to do this, we looked at correlations between diverse components and variables in the software development sector. In order to compare and examine these factors, we produced data visualization. We mined the SODS data set using a variety of approaches, such as categorization and the identification of recurrent patterns. Furthermore, we aimed to test a hypothesis that articulates gender diversity in IT businesses by answering a few pertinent questions.

Report Organization. We reviewed several literature works and presented their overviews in Section 2. We describe important details of SODS dataset in Section 2.1 followed by the experimental results our work in Section 3. Section 4 concludes the report with future plans.

For ease of reading, below we list the contribution of our projects with report organization.

- We implemented ***data warehousing*** for SODS dataset for online analytical processing (OLAP). [3.1]
- A thorough ***exploratory data analysis (EDA)*** was performed on the dataset to examine the data before coming to any assumption. We presented important ***data visualization*** of different attributes of SODS. Moreover, we experimented noise handling approaches on the data and filtered essential portions for further analysis. [3.2]
- We highlighted on compensation related attributes in SODS since this is an useful parameter to analyze job market. In this project, we also proposed an ***salary indicator*** to compare a person's income with country-based average salary. [3.2.2]
- In order to observe popular tools in software development field, we performed ***frequent pattern analysis (FPA)*** on particular attributes of SODS. [3.3]
- We performed ***correlation anaysis*** on significant variables. [3.4]
- We applied ***classification and regression algorithm*** for salary range classification and prediction. ***K-means algorithm*** was also applied to identify noises. [3.5, 3.6]
- We attempted to answer a question: *having the same job position, educational background, age and technical experience, do the female employers obtain same remuneration as male employers?* We applied ***Naïve Bayesian model*** for the testing. [3.7]
- In addition, we reported the ***insights*** from our mining which can be used as future reference. Our findings will assist researchers to adopt or abandon particular mining techqniues—especially while working on survey based dataset.

2 Related Work

This section contains earlier studies that examined Stack Overflow or similar platform related data. Numerous research articles examined user participation on the SO platform and drew conclusions from it. *Fu et al.* [7] discovered that the interest change rate of suers follows a power-law distribution, which is different from the research-based interest change. According to the author, this phenomenon indicates that the community is more inclined to exploration strategy. In the technical forum, the study of *Brooke* [3] reveals a dominant male influence. *Moutidis et al.* examines the migration of the users between communities [9]. Note that, the aforementioned studies considered their dataset by scraping Stack Exchange forum [2]. Moroever, Some publications analyze their own conducted surveys of SO users, such as [2, 14, 15]. *Georgiou et al.* [8] incorporated SODS (2020) and scraped data to find COVID-19 impact on the Stack Overflow. Similarly, *Ford et al.* [6] used SODS from the year of 2017 with scraped data and found that women were more likely to start interacting with one another in SO than those who did not when they came across another woman. In 2020, *Nivala et al.* [10] analyzed SODS (2011–2018) and reported their findings. They concluded that, the novices are becoming more involved in the community, while number of expert users who provide good answers has decreased. According to them, developing countries are becoming more and more visible in SO. However, the apparent gender gap in programming has not been addressed by the Stack Overflow community. In [4] by *Dada et al.*, the authors exhibited different visualizations of different attributes from the SODS (2020) and measure the frequencies. For example, *JavaScript* was the top programming skills used in various IT roles. The authors identified and ranked the top popular IT tools for 23 distinct IT professions, including operating systems (OS), databases, programming languages, and collaborative applications.

We can see from the aforementioned analysis of the literature that none of the studies—to the best of our knowledge—addressed the analytical perspective we are interested in, and we think our project will help to close this gap.

2.1 Overview of the SODS Dataset

The SODS dataset consists of 73268 samples where there are different types of questions with different answering formats, such as multiple choice (single answers or multiples answers), Likert, Writing Text in the box. Some questions

[2] https://stackexchange.com/

are mandatory to response, while some are optional (which may cause "NaN" in the entries).

Our study will attempt to give a thorough examination of each of these entities separately as well as an analysis of their connections. Note that, there are other types of questions in the SODS that targets SO Usage, Community and the survey itself; we avoid this types from the scope of our project since they are irrelevant to the job-market analysis. We also ignore the questions that requires answers as texts in the boxes.

3 Experimental Results

In this section we explain our implementations and show results along with observation. For implementation, we used `Python` programming language with the few packages, which are: `numpy`, `scipy`, `pandas`, `matplotlib`, `sklearn`, `mlxtend`, `pycountry_convert`, `geopy`, and `geopandas` and `atoti`, etc.

3.1 Data Warehouse

Since SODS dataset is huge, we implemented a data warehousing for a bird's eye overview of the dataset and improved data analytics. We applied *Atoti*[3] via `atoti` python package. After establishing necessary sessions and connection with our SODS, launching of *Atoti* from Jupyter Lab provides a link to dashboard of the dataset which allowed us intial exploration. Fig. 1 shows a snapshot from Atoti dashboard with treemap options. Here we splitted education level (`EdLevel`) grouped by countries. We also explored slices and dices on the cube created on SODS using our warehouse. A sample of such exploration is shown in Fig. 2.

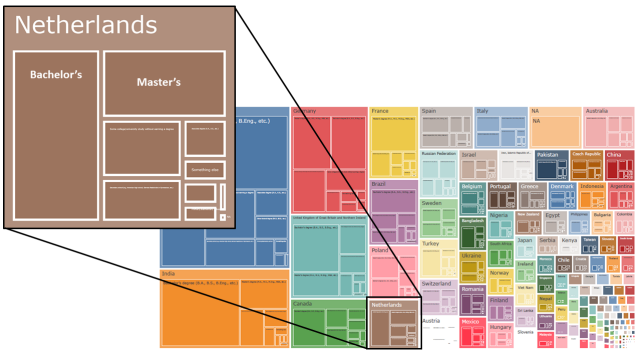

Figure 1. Treemaps of `EdLevel` grouped by *Country* attribute. Each colored rectangle represents a country and its size indicates number of samples under that particular country. Similarly, sub-rectangles represents the samples with different education level, e.g., Bachelor's. Master's, etc. Visualization for the country *Netherlands* is zoomed in for better understanding.

[3]https://www.atoti.io/

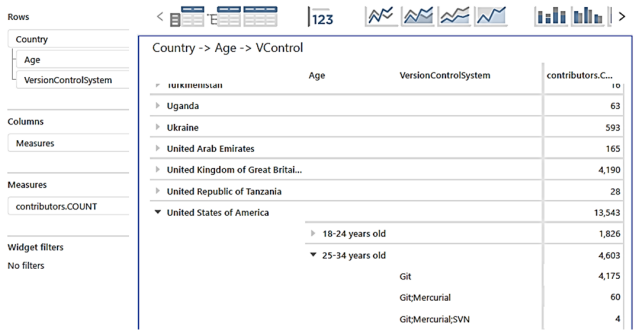

Figure 2. Exploration on data cube of SODS dataset from our data warehouse. Here we applied slices on countries and investigated further on *age* and *version control tools*.

3.2 Exploratory Analysis and Data Visualization

To work with our dataset, as our dataset was large and contained various range of values per sample, first we decided to visualize different properties of the dataset. Visualizations from our different exploration are shown below.

3.2.1 Preliminary Analysis. We attempted to visualize how many unique values we have in each attribute which will help us to be able to choose interesting attributes to work with. Figure 1 shows the demonstration of the number of unique values each attribute. See Fig. 3. This is to assist us have an easier view of the stories surrounding data into a form easier to understand.

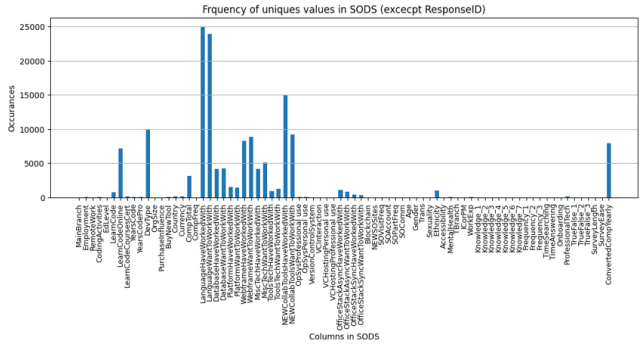

Figure 3. Frequency of unique values in SODS Dataset.

In order to enhance the performance of the dataset used for this project for all metrics, a preprocessing procedure has been implemented to ensure and to "flesh out" `NaN` values which represent undefined or unrepresentable data as depicted in Fig. 4*(top)*. This *cleaning* step is important as to avoid missing values, noisy data, and unwanted inconsistencies in the SOSD dataset. It was at this point that we noted the entire attributes will not be needed for this project.

We also implemented a *reduction* technique that extracts attribute based on a minimum amount of allowed `NaN` values. Therefore, we can filter out a subset of SODS with a

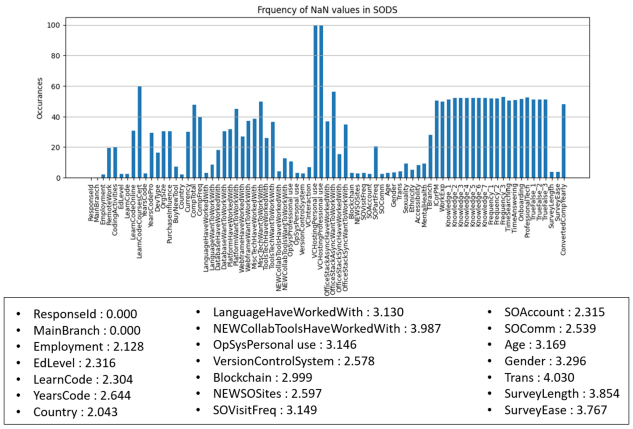

Figure 4. *Top:* Distribution of NaN values in SODS. *Bottom:* Attributes which has less than 5% of NaN values in SODS. It shows the name of the attributes with percentages of NaN values.

minumum percentage on NaN values, e.g., 5% NaN values. See Fig. 4*(bottom)*. We can also work with a sebset which does not contain any NaN value. The idea is to apply AND operation on the attributes to keep entries which has no NaNs in the rows.

Gender based Analysis. Focusing specifically on the gender group, we wanted to visualize the distribution of age in each gender to check the percentage in software development. We found an interesting finding here which shows that the age range is similar for both men and women i.e., most of the people involved in software development have an age of 25-34 years old. For the male persons, the percentage is 39.6%, for females it is 46.8%, and for other genders it is 29.5% which can be seen in Fig. 5.

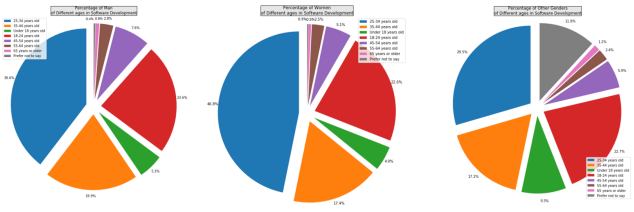

Figure 5. Gender wise Age Distribution. *Left to right:* Male, Female and Other gender.

In Fig. 6, we can see the gender ratio in software development along with each gender's average salary. A significant finding can be seen from this figure that, the participation of men in software development is almost 70000, whereas only around 5000 women are involved in this sector. Also, the average salary for male members is remarkably higher than the female members which is depicted in the figure.

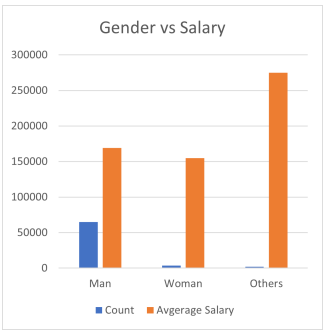

Figure 6. Gender wise participation in software Development with their average salary with frequency of participation.

3.2.2 Salary Analysis. Fig. 7 portrays the information of the yearly distribution of salary of the people in software development. The `Converted Comp Yearly` attribute holds the yearly salary converted to USD for the participants. As we got a skewed distribution in Fig. 7*(top)*, we performed a log-based transformation in Fig. 7*(bottom)* to find out a well distributed range which will help us to identify noise, and outliers.

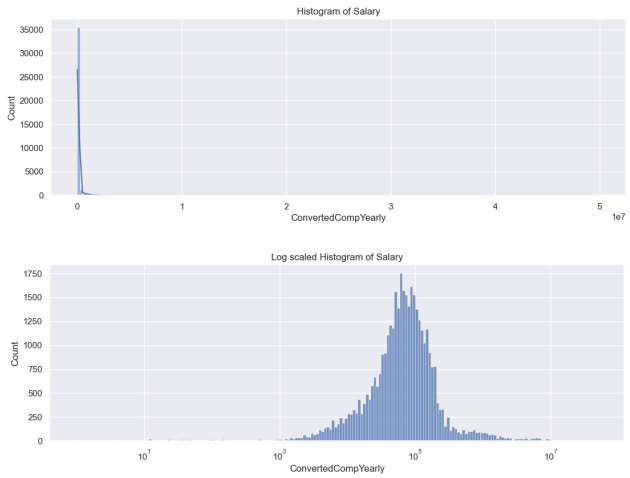

Figure 7. *Top:* Yearly salary distribution of people in software development, *Bottom:* Log-scaled visualization.

Since SODS dataset is collected from survey, presence of noisy data is highly expected. Particularly when it comes to salary, participants can enter noisy data which can be treaed as errors. In this regard, we tried our best to avoid those entries and we assume a particular middle range of the salary distribution can serve as uncorrupted data. Therefore we experimented by selecting percentiles. For example, Fig. 8 provides the boxplot of the yearly salary in between 3 and 85 percentile.

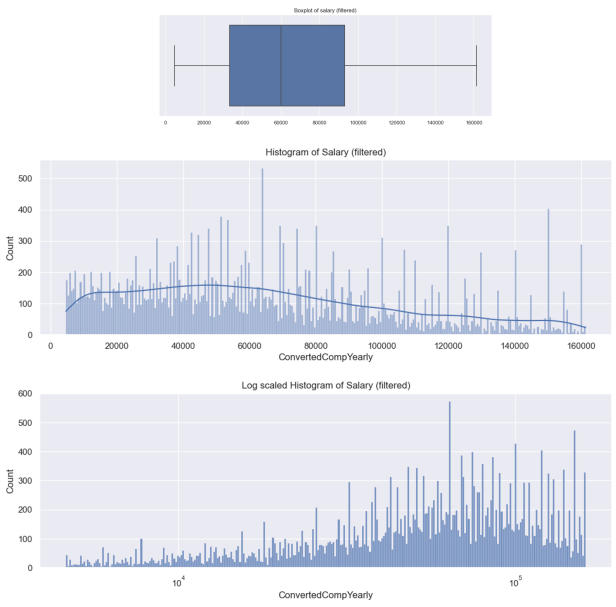

Figure 8. Boxplot of the salary (*Top:* `ConvertedCompYearly`) from the range of 3 and 85 percentile. *Middle:* Salary distribution (between 3 and 85). *Bottom:* Log-scaled visualization of the data.

Country-based Salary Analysis. Geolocation based analysis was carried out on one hundred and eighty (181) countries. We investigated to determine the few research questions, i.e., *Developers from which countries secure more salaries?* We faced several challenges to encounter this analysis, for example, *(i)* Not all the countries were in common format, e.g., *North Macedonia* is entered as *The former Yugoslav Republic of Macedonia* in SODS. *(ii)* Lattitude and longitude of the countries were necessary to plot multivariate information on world map for better visualization. Therefore, we performed necessary *transformation* for converting country names. Fig. 9 shows geolocation based analysis of the salary (`ConvertedCompYearly`). It shows the maximum salary (converted to USD) for individual countries. We can observe that, maximum salary occurs in Europe.

In addition, we also applied similar plotting for the salary filtered by percentile range (3^{rd} –85^{th} percentile). Fig. 9 *(bottom)* shows the geographical map.

Salary Ratio. Though SODS dataset has information regarding compensation, we need further investigation to compare this information based on geo-location. In this report we propose an indicator that helps us understanding the affect of the salary for a particular person. For instance, we would like to see whether salary of a person X living in a country c is sufficient. To accomplish this, we collected a list of average salaries (for software developers) in major countries [13]. We scanned through SODS and determine a

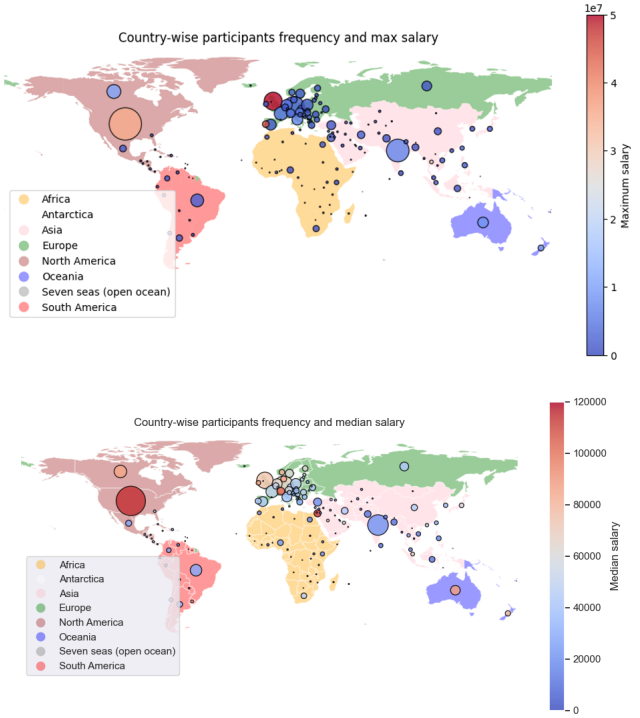

Figure 9. *Top:* Geolocation based analysis of the salary (`ConvertedCompYearly`). The radius of the circles shows the number of participants in SODS for an individual location. The colors represent the maximum salary. *Bottom:* Similar plot based on percentile filtering (3^{rd} –85^{th} percentile).

person-to-country ($p2c$) salary ratio using Eq. 1.

$$p2c = \frac{S(X_c)}{S_{avg}(c)} \tag{1}$$

Here $S(X_c)$ is the yearly salary of a person X of country Y, and $S_{avg}(c)$ is the average salary obtained by the software developers in country c.

Our intention is to use this indicator, $p2c$, to represent the salary status with respect to an individual country. For instance, if $p2c > 1.0$ for a person, we can say the person secures satisfactory amount of compensation. We performed a thorough analysis based on our $p2c$ indicator and presented our results in Fig. 10. Here we applied the $p2c$ salary ratio to all the entries in SODS and group them by countries to observe the median values. Lastly, we visualize a map showing the medians results for better observation (see Fig. 11).

3.3 Frequent Pattern Analysis

In this section, we delineate our findings from the frequent pattern analysis using `Apriori` and `FP-Growth` algorithm.

3.3.1 Apriori Algorithm. It is really important to observe the current situation in the tech community since we aim to mine job market-related insights from SODS. Therefore,

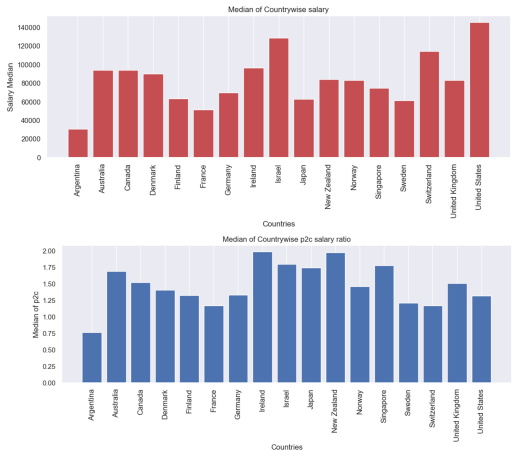

Figure 10. *In top:* median of salaries obtained by developers in major countries major countries. *Bottom:* median of *p2c* ratios of salaries obtained by developers for the same countries.

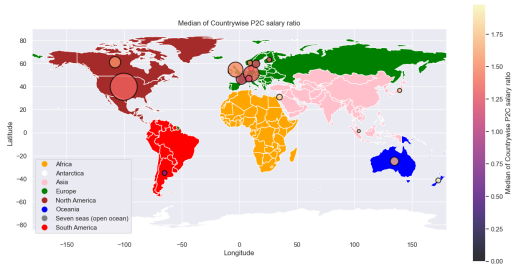

Figure 11. Geolocation based analysis of the salary (`ConvertedCompYearly`) based on percentile filtering (3–85 percentile). The radius of the circles shows the number of participants in SODS for an individual location. The colors represent the maximum salary with *cool-to-warm* colormap.

we attempted to find the most frequently used programming languages from SODS. Instead of extracting one popular language, we worked on extracting a set of frequently used programming languages. In the survey, SO allowed the participants to enter multiple entries of programming languages as their favorite ones. Hence, we applied the *Apriori* algorithm [1] for frequent pattern mining of the attribute (`LanguageHaveWorkedWith`).

For this purpose, we performed string preprocessing as the attribute `LanguageHaveWorkedWith` contains values separated by semicolon, where our implementation required them to be stored as `python`'s list structure. After that, we applied simple string splitting and list appending to obtain the desired format.

Fig. 12 shows the frequent pattern analysis performed to find the most frequently used programming languages using the *Apriori* algorithm. The result showed that the frequently used programming languages are `HTML/CSS` and `JavaScript` when the minimum support is set for 0.4 (Fig. 12).

Again, when the minimum support value is set to 0.06, we found that the most frequently used programming language is only `JavaScript`, which denotes that as we make the minimum support value higher, we get a fewer number of frequent items in the itemset. We also aimed to mine the most popular operating systems currently used by programmers as we believe this can be an insightful information to the job-seeker. Using the *Apriori* algorithm with the `OpSysPersonalUse` attribute at the minimum support value of 0.17, we figured out that the most popular Operating Systems for personal use are the `Linux-based` and `Windows` OS. We used this support value as giving a greater value was only giving us a single OS.

In addition, we found the most frequent job in the current tech industry using `Apriori` and discovered that it is the job of the `full-stack Developer`. And we used a minimum support value of 0.45 this time.

	support	itemsets
0	0.551490	(HTML/CSS)
1	0.333131	(Java)
2	0.654357	(JavaScript)
3	0.481226	(Python)
4	0.494921	(SQL)
5	0.348743	(TypeScript)
6	0.490525	(JavaScript, HTML/CSS)
7	0.332244	(SQL, HTML/CSS)
8	0.311180	(Python, JavaScript)
9	0.373864	(JavaScript, SQL)
10	0.314294	(TypeScript, JavaScript)
11	0.300275	(JavaScript, SQL, HTML/CSS)

min_supp = 0.3

	support	itemsets
0	0.551490	(HTML/CSS)
1	0.654357	(JavaScript)
2	0.481226	(Python)
3	0.494921	(SQL)
4	0.490525	(JavaScript, HTML/CSS)

min_supp = 0.4

	support	itemsets
0	0.551490	(HTML/CSS)
1	0.654357	(JavaScript)

min_supp = 0.5

Figure 12. Results of *Apriori* algorthim for `minimum support` starting from 0.3 to 0.6 *(left to right).*

3.3.2 FP-Growth Algorithm.

To ensure the claim that `HTML/CSS` and `JavaScript` are the most frequently used programming languages, we applied the `FP-Growth` algorithm in addition. It became evident as we got the same result as `Apriori` with a minimum support value of 0.04.

3.4 Correlation Analysis

We performed correlation measurement between different attributes of our dataset to find out if there any correlation exists between them. We employed both lift measurement and χ^2 Analysis.

3.4.1 Lift measurement.

We computed the lift measurement value calculation to find out if the female members are more likely to work in-person, remote, or in hybrid mode. For this reason, we used the gender and remote work status (`RemoteWork`) attribute. However, a challenge we faced in this case was that our gender attribute contained a lot of

noise, e.g., some of the samples contained multiple genders which we needed to handle. For this reason, we considered the single 'Man' values as male members and single 'Woman' values as Female members. The rest of the values were considered as 'Others' gender. Later, we generated the contingency table using the `Gender` (Fig. 13a) and `RemoteWork` attributes and then we computed the lift values. From the lift values, we discovered a positive correlation between female persons and fully remote method of work (Fig. 13b). Therefore, it is evident that female persons prefer to work in fully remote mode, rather than working in in-person or hybrid mode.

RemoteWork	Full in-person	Fully remote	Hybrid
Gender			
Female	415	1341	1183
Male	7870	23057	22957
Others	192	668	611

(a) Contingency table for `Gender` vs `RemoteWork`.

```
lift( 0 , 0 ) = 0.9710 [-ve corr]
lift( 0 , 1 ) = 1.0611 [+ve corr]
lift( 0 , 2 ) = 0.9480 [-ve corr]
lift( 1 , 0 ) = 1.0044 [+ve corr]
lift( 1 , 1 ) = 0.9951 [-ve corr]
lift( 1 , 2 ) = 1.0034 [+ve corr]
lift( 2 , 0 ) = 0.8976 [-ve corr]
lift( 2 , 1 ) = 1.0561 [+ve corr]
lift( 2 , 2 ) = 0.9783 [-ve corr]
```

(b) Lift values based on the contingency table.

Figure 13. Lift Analysis for `Gender vs RemoteWork`

3.4.2 The χ^2 test. To ensure the significance of the insights found from the previous analysis, we performed a Chi-Square Analysis using the contingency table found in Fig. 13a. As our degree of freedom is $df = 4$, for the significance level of 0.05, we used the chi-square table[4] and found that these two attributes (`Gender` and `RemoteWork`) are correlated as the computed chi-square value was greater than the critical value. After that, we calculated the expected values (shown in matrix 2) and mined some significant insights.

$$e_a = \begin{bmatrix} 427.383 & 1263.748 & 1247.867 \\ 7835.706 & 23169.731 & 22878.561 \\ 213.909 & 632.519 & 624.570 \end{bmatrix} \quad (2)$$

For instance, as we can see from the observed values in Fig. 13a and expected values in matrix 2, $o_{11} < e_{11}$, $o_{12} > e_{12}$, and $o_{13} < e_{13}$, we can say that female persons are more likely to work in fully remote mode, rather than working in-person or in hybrid mode which makes the claim from lift analysis stronger as the claims are same in both cases.

Similarly, we executed another chi-square test on the attributes remote work status (`RemoteWork`) and education level (`EdLevel`) to find out if the PhD persons prefer to work remotely or in-person. Here, the attribute `EdLevel` had nine unique values and we categorized them as PhD or non-PhD. Then, we generated a contingency table using these two attributes (Fig. 14). From the contingency table, as $df = 2$ and for the significance level of 0.05, we get our computed χ^2 value greater than the critical value. Also, from the observed values in Fig. 14 and the expected values shown in matrix 3, we see, $o_{21} > e_{21}$, $o_{22} < e_{22}$, and $o_{23} > e_{23}$, it reflects that the PhD persons are more likely to work in person or hybrid, rather than working remotely.

RemoteWork	Full in-person	Fully remote	Hybrid
EdLevel			
Non-PhD	8294	24720	23900
PhD	302	621	1121

Figure 14. Contingency table for `EdLevel` vs `RemoteWork`.

$$e_b = \begin{bmatrix} 8297.98 & 24462.45 & 24153.55 \\ 298.01 & 878.54 & 867.44 \end{bmatrix} \quad (3)$$

3.5 Classification and Regression

In this project, we developed classification and regression models on the SODS dataset. For this purpose we performed feature engineering on the attributes to transform them into usable format.

3.5.1 Decision Tree. We experimented on decision tree and our target was to classify a person's profile into labels of salary range, e.g., low or high. At first we filtered out the a range of salary data based on percentile. In our case, based on empirical analysis, we considered the data between 2 to 90 percentile.

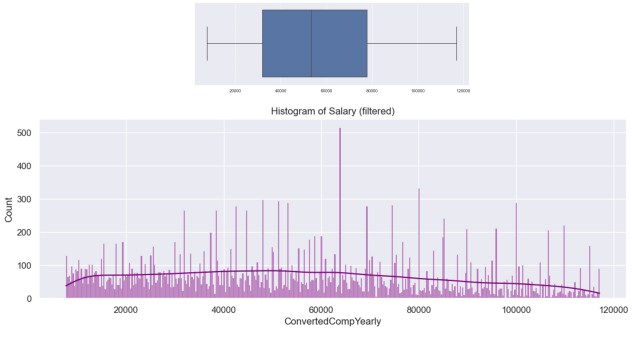

Figure 15. *Top:* Boxplot of the salary (`ConvertedCompYearly`) from the range of 2 and 90 percentile. *Bottom:* histogram of that range of salary.

In our decision tree, we considered Salary as target label and our feature attributes are: programming language experience (`LanguageHaveWorkedWith`), Age, Years of Coding

[4] https://people.richland.edu/james/lecture/m170/tbl-chi.html

(YearsCode) and Work experience (WorkExp). Among several programming language values, we considered a subset: *{ C#, C++, Java, JavaScript, PHP, SQL, Python, HTML/CSS}*. However, feeding this into decision tree model was not straightforward task, since one row might have multiple entries for LanguageHaveWorkedWith column, e.g., Java; C++, Python. Therefore, we transformed the column into multiple ones to introduce indicator or dummy variable (Fig. 16 *(top)*).

We also performed further feature engineering on Age and YearsCode attribute, since they contains unwanted inputs, e.g., Prefer not to say, Less than 1 year, etc. Therefore, we handled them with appropriate processing, such as replacing with corresponding labels, replacing with NaN, etc. In the following stage, we normalize the salary using *min-max* normalization. See Fig. 16*(bottom)* for sample output.

	C#	C++	Java	JavaScript	PHP	SQL	Python	HTML/CSS
0	1	1	0	1	0	0	1	1
1	0	0	0	0	0	0	0	0
2	1	0	0	1	0	1	0	1
3	0	0	0	0	0	1	0	1
4	0	0	0	1	1	0	1	1
...	...	...	...	...	...	...	...	...
26061	1	0	1	1	0	0	0	1
26062	0	0	0	1	0	0	0	1

	C#	C++	Java	JavaScript	PHP	SQL	Python	HTML/CSS	Age	YearsCode	WorkExp	ConvertedCompYearly
8	1	0	0	0	0	1	0	0	34	6	6.0	0.378049
12	0	0	0	1	0	1	1	1	34	12	5.0	0.523729
14	0	0	1	0	0	0	0	0	34	11	5.0	0.934890
21	1	1	1	1	0	0	1	1	34	5	4.0	0.241635
22	0	0	1	0	0	0	0	0	44	25	23.0	0.821638
...	...	...	...	...	...	...	...	...	...	...	...	...
26052	0	0	0	1	0	0	0	1	34	7	2.0	0.010306
26058	0	0	1	1	0	0	0	1	34	20	13.0	0.091333
26059	1	1	0	0	0	0	0	0	34	7	6.0	0.692396

Figure 16. *Top:* Result of applying indicator or dummy variable on programming language subset. *Bottom:* Result after processing unwanted entries.

We also needed to categorize different salary since it was numerical. Therefore we categorize them based on percentile filtering. For example, we consider salary up to 40^{th} percentile as *label-0* or *low*, and the rest as *label-1* or *high*. Though it sounds impractical from the real-world perspective, this can be a pivotal point to start training and observe the outcomes.

Observation. We considered $70-30$ split ratio for training and testing. We found that, considering 10^{th} percentile as *label-0* and rest as *label-1* provides 83% accuracy. Experimental results for other ranges of percentile fitering is stated in Table 1. We can see that, if the filtered dataset is more skewed for higher values (*label-0*), the model works better. However, it introcudes over-fitting as well.

3.5.2 Linear Regression. In terms of predicting salaries, linear regression is the best choice for training the model. Therefore we implemented regression algorithm. For this puprose, we need a different feature engineering. In that case, we normalize all the columns except the ones with indicator variables using *min-max* normalization. Fig. 17 shows a sample of processed data.

Table 1. Accuracy based on different percentile filtering.

Percentile for L-0	Percentile for L-1	Accuracy
10^{th}	90^{th}	83%
20^{th}	80^{th}	71%
30^{th}	70^{th}	64%

	C#	C++	Java	JavaScript	PHP	SQL	Python	HTML/CSS	Age	YearsCode	WorkExp	ConvertedCompYearly
8	0	0	0	0	0	1	1	0	0.3	0.084746	0.12	0.232805
11	0	0	1	1	0	1	0	1	0.5	0.186441	0.28	0.971537
12	0	1	0	0	0	0	0	0	0.3	0.186441	0.10	0.313843
14	0	0	0	1	0	1	1	1	0.3	0.169492	0.10	0.542562
21	0	0	1	1	0	1	1	1	0.3	0.067797	0.08	0.156922
...	...	...	...	...	...	...	...	...	...	...	...	...
33581	1	0	0	0	0	0	0	0	0.3	0.084746	0.10	0.238229
33582	0	0	0	1	0	0	0	1	0.3	0.271186	0.20	0.230610
33591	1	0	0	1	0	0	0	1	0.9	0.694915	0.88	0.525505

Figure 17. Result of normalization on the dataset for linear regression.

Observation. We trained our model with $70-30$ split ratio for training and testing. We used matrices to measure the performance of the linear regression model. We found that *mean absolute error (MAE)*, *mean squared error (MSE)* and *root mean squared errors (rMSE)* are 0.1773799, 0.0498956, and 0.2233732 respectively. A sample results of predicted output and actual data is presented in Fig. 18.

	Actual	Predicted
11610	0.223230	0.274177
18498	0.324232	0.347085
15889	0.949174	0.339960
5238	0.015919	0.252493
12546	0.677999	0.451250
...	...	...
30607	0.341076	0.281437
20198	0.124545	0.285627

Figure 18. Results of our linear regression model for salary prediction.

In our observation, the model performed better in terms of predicting salaries and we recommend to implement linear regression instead of decision tree for salary based research in future.

3.6 Clustering

As an experiment, we implemented clustering algorithms. We considered *yearly salary* (ConvertedCompYearly) and *years of coding* (YearsCode). We applied k-means clustering algorithm for this two attributes and result is presented in Fig. 19. The clustering output may help us in future to determine noise. For example, the green points can be considered

as noise since higher salaries for less number of years coding seems impractical.

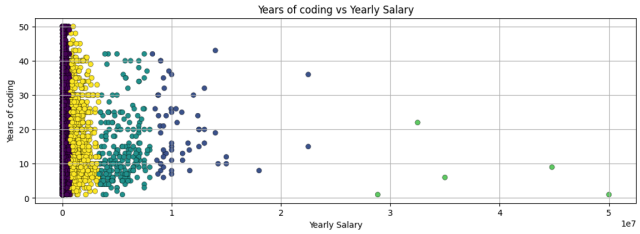

Figure 19. K-means clustering on *yearly salary* vs *years of coding* for $k = 5$.

3.7 Naive Bayes Classification

It is essential to observe if there is any discrimination between the wages paid to male vs female members in the tech industry. For this purpose, we employed Naive Bayes Classification algorithm to perceive if there is any difference between the salaries paid to different sexual orientations. To be more specific, given the same level of employment, educational background, coding experience, working experience, and age, we tried to discover if a female employee is paid the same salary as a male employee. Hence, we worked by creating a DataFrame using the attributes `Employment`, `EdLevel`, `YearsCode`, `Gender`, `Age`, `WorkExp`, as feature variables and `ConvertedCompYearly` attribute as the target variable to detect the salary range (See Fig. 20.

We first normalized the value of `ConvertedCompYearly` using `Min-max Normalization`. After that, we took the first 10^{th} percentile as *label 0* and rest of the values as *label 1*. We used this distribution as our labels as our `ConvertedCompYearly` attribute had imbalanced data and was more biased to the first 10 percent of the data. Then, we encoded the attributes `Employment`, `EdLevel`, and `Gender` as those contained categorical values. For simplification, we also assumed the values of `YearsofCoding` *less than* 1 as 1, and *greater than* 50 as 50. Next, we trained our model with all these feature variables while we split our dataset in a $80-20$ ratio and gained an accuracy of 85.89%. The normalized confusion matrix is shown in Fig. 21. Also, the other evaluation measures are shown in Table. 3.

Here, from Table. 3, we can see that our precision is higher than the recall, which denotes that our rained model returns more relevant results than the irrelevant one i.e., most of our predicted labels are correct when compared to the training labels. However, our accuracy is less than the recall, as the `ConvertedCompYearly` attribute's distribution was imbalanced. From our analysis, predicting the salary range of different genders gave us a significant insight that though the females may have the same skills and experience, still there is a remarkable difference in their salary compared to men. To be more precise, female employees get less amount of salary compared to male employees (An example is shown in Table. 2.)

	Employment	EdLevel	YearsCode	Gender	Age	WorkExp	ConvertedCompYearly
8	Employed, full-time	Master's degree (M.A., M.S., M.Eng., MBA, etc.)	6	Woman	25-34 years old	6.0	49056.0
11	Employed, full-time;Independent contractor, fr...	Bachelor's degree (B.A., B.S., B.Eng., etc.)	12	Man	35-44 years old	14.0	194400.0
12	Employed, full-time	Bachelor's degree (B.A., B.S., B.Eng., etc.)	12	Man	25-34 years old	5.0	65000.0
14	Employed, full-time;Independent contractor, fr...	Master's degree (M.A., M.S., M.Eng., MBA, etc.)	11	Man	25-34 years old	5.0	110000.0
21	Employed, full-time	Master's degree (M.A., M.S., M.Eng., MBA, etc.)	5	Man	25-34 years old	4.0	34126.0
...	...	...	...	...	...	...	...
73110	Employed, full-time	Bachelor's degree (B.A., B.S., B.Eng., etc.)	13	Woman	25-34 years old	9.0	60906.0
73112	Employed, full-time	Bachelor's degree (B.A., B.S., B.Eng., etc.)	10	Man	25-34 years old	3.0	52255.0
73113	Employed, full-time	Master's degree (M.A., M.S., M.Eng., MBA, etc.)	7	Man	25-34 years old	7.0	94000.0
73116	Employed, full-time	Bachelor's degree (B.A., B.S., B.Eng., etc.)	21	Man	35-44 years old	16.0	115000.0
73119	Employed, full-time	Bachelor's degree (B.A., B.S., B.Eng., etc.)	5	Man	25-34 years old	9.0	70000.0

Figure 20. A snapshot of the DatFrame with `Employment`, `EdLevel`, `YearsCode`, `Gender`, `Age`, `WorkExp`, `ConvertedCompYearly` attribute

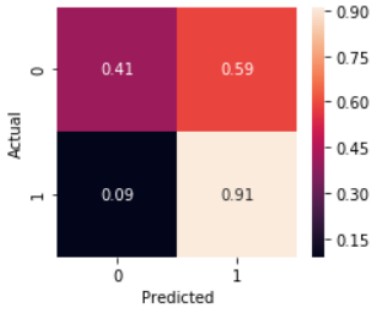

Figure 21. Confusion Matrix of the Naive Bayes Classifier.

4 Conclusion

In this project, we explored Stack Overflow Developer Survey (SODS) and investigated the relationship between different variables of the tech community. We wanted to find out which are the most used programming languages, operating systems, and other important aspects among the IT community. During our analysis, we faced some challenges while working with this dataset. A large portion of the variables of our dataset contained *NaN* values, also the distribution was partially imbalanced. We had to execute a lot of preprocessing and cleaning to handle this diverse dataset. Some of our classifier algorithms could not achieve great accuracy due to this diversity. However, we tried to handle this imbalance as much as possible while finding the proper explanations behind our classifier results. However, we learned some lessons while working with this dataset which can be useful for future researchers to work with. We also could establish a significant hypothesis that men are more likely to get higher salaries in the software community rather than women in spite of having the similar level of experience and skills. In the future, we plan to advance our work using this dataset to fine-tune the classifiers for higher accuracy and

Employ-ment (encoded)	Ed-Level (encoded)	Years-Code (years)	Gender (0 - man) (1 -others) (2-woman)	Age (encoded)	WorkExp (years)	ConvertedCompYearly (Predicted label)
11	7	12	0	1	12.0	1 (above 10th Percentile)
			2			**0 (below 10th Percentile)**
			1			1 (above 10th Percentile)

Table 2. Salary discrimination between Man vs Woman

Accuracy	0.8589
Precision	0.9306
Recall	0.9105
F1 Score	0.9204

Table 3. Evaluation Metrics of Naive Bayes Classification

other relevant algorithms to gain some more meaningful insights.

References

[1] Rakesh Agrawal, Ramakrishnan Srikant, et al. 1994. Fast algorithms for mining association rules. In *Proc. 20th int. conf. very large data bases, VLDB*, Vol. 1215. Santiago, Chile, 487–499.

[2] Tanveer Ahmed and Abhishek Srivastava. 2017. Understanding and evaluating the behavior of technical users. A study of developer interaction at StackOverflow. *Human-centric Computing and Information Sciences* 7, 1 (2017), 1–18.

[3] SJ Brooke. 2021. Trouble in programmer's paradise: gender-biases in sharing and recognising technical knowledge on Stack Overflow. *Information, Communication & Society* 24, 14 (2021), 2091–2112.

[4] Oluwaseun Alexander Dada, George Obaido, Ismaila Temitayo Sanusi, Kehinde Aruleba, and Abdullahi Abubakar Yunusa. 2022. Hidden Gold for IT Professionals, Educators, and Students: Insights From Stack Overflow Survey. *IEEE Transactions on Computational Social Systems* (2022).

[5] Arnal Dayaratna. 2021. Quantifying the Worldwide Shortage of Full-Time Developers. https://www.idc.com/getdoc.jsp?containerId=US48223621 Accessed: 2022-09-28.

[6] Denae Ford, Alisse Harkins, and Chris Parnin. 2017. Someone like me: How does peer parity influence participation of women on stack overflow?. In *2017 IEEE symposium on visual languages and human-centric computing (VL/HCC)*. IEEE, 239–243.

[7] Chenbo Fu, Xinchen Yue, Bin Shen, Shanqing Yu, and Yong Min. 2022. Patterns of interest change in stack overflow. *Scientific reports* 12, 1 (2022), 1–10.

[8] Konstantinos Georgiou, Nikolaos Mittas, Alexandros Chatzigeorgiou, and Lefteris Angelis. 2021. An empirical study of COVID-19 related posts on Stack Overflow: Topics and technologies. *Journal of Systems and Software* 182 (2021), 111089.

[9] Iraklis Moutidis and Hywel T. P. Williams. 2021. Community evolution on Stack Overflow. *PLoS ONE* 16 (2021).

[10] Markus Nivala, Alena Seredko, Tanya Osborne, and Thomas Hillman. 2020. Stack Overflow–Informal learning and the global expansion of professional development and opportunities in programming?. In *2020 IEEE Global Engineering Education Conference (EDUCON)*. IEEE, 402–408.

[11] U.S. Bureau of Labor Statistics. 2022. Occupational Outlook Handbook. https://www.bls.gov/ooh/computer-and-information-technology/computer-programmers.htm Accessed: 2022-09-28.

[12] Prasad Patil. 2022. What is Exploratory Data Analysis? https://towardsdatascience.com/exploratory-data-analysis-8fc1cb20fd15 Accessed: 2022-10-5.

[13] Tracy Phillips. 2021. Average Software Engineering Salaries by Country in 2022 (Comparison of 20+ Countries). https://codesubmit.io/blog/software-engineer-salary-by-country/ Accessed: 2022-12-06.

[14] Chaiyong Ragkhitwetsagul, Jens Krinke, Matheus Paixao, Giuseppe Bianco, and Rocco Oliveto. 2019. Toxic code snippets on stack overflow. *IEEE Transactions on Software Engineering* 47, 3 (2019), 560–581.

[15] Yuhao Wu, Shaowei Wang, Cor-Paul Bezemer, and Katsuro Inoue. 2019. How do developers utilize source code from stack overflow? *Empirical Software Engineering* 24, 2 (2019), 637–673.

A Individual Author's Contribution

The contributions from individual project members are stated below. Table 4 provides a summary of the responsibilities of taken by authors.

Mohammad Imrul Jubair. The author contributed to implementing data warehousing and utilizing it, exploratory data analysis of country-wise salary, and map visualization. Furthermore, he worked on related work of the project, worked for finding the project dataset, and implemented percentile filtering. He also proposed the idea of the *p2c* indicator and applied it to SODS. The implementation of k-means clustering, frequent pattern analysis, and correlation was implemented by Jubair. He also fully implemented the decision tree and regression algorithm with the necessary feature engineering and transformation of the data. He also performed a thorough analysis to find the pros and cons of applying different classifiers on SODS. He also contributed to a significant portion of the report and slides (for proposal, checkpoints and final). Collating works of other members was also a major responsibility of Jubair. He maintained the main repository of the project codes and resources.

Mohsena Ashraf. The author contributed by implementing a preliminary exploratory analysis of the project, along with significant data visualization. She performed analysis on previous research works related to the project, and also analyzed gender-wise salary distribution, age-wise, and gender-based participation in the tech market. She researched on finding the main dataset collection, along with finding the appropriate dataset for *p2c* indicator and analyzing it. Mohsena

Table 4. Responsibilities of individual members.

	Data Warehouse	EDA & Visualization	p2c salary ratio	Frequent Pattern	Correlation	Clustering	Decision Tree	Regression	Naive Bayesian	Reports	Slides
Jubair	✓	✓	✓	✓	✓	✓	✓	✓		✓	✓
Mohsena		✓	✓	✓	✓	✓			✓	✓	✓
Cornelius		✓		✓	✓					✓	

also took the responsibility of implementing several correlation measures, and clustering analysis, and scrutinizing corresponding outcomes while also performing data preprocessing. She was also responsible for most of the logical explanations of the correlation measures and frequent pattern analysis. She fully implemented Naïve Bayesian model with necessary feature engineering for answering our gender-based research question. She also contributed significantly to the report, making slides and presenting them (for proposal, checkpoints, and final). Investigating errors in the results and finding theoretical explanations behind them was also a major task of Mohsena.

Cornelius Onimisi Adejoro. The author contributed to several visualization and exploratory analysis while performing data cleaning. Cornelius re-implemented Apriori to observe the correctness of the implementation. He implemented heat-maps for correlation. He contributed to present problem domain of our project in the report (for proposal).

B Honor Code

On my honor, as a University of Colorado Boulder student I have neither given nor received unauthorized assistance.