Learning Development Project

Group 1

Federico Riva, Georgios Efthymiadis, Mertkan Usta, Jack Mills

Introduction

Nowadays customer reviews are more important than ever. With the advent of websites like Tripadvisor and Yelp, it is crucial for businesses such as restaurants to gain recommendations and good feedbacks on their online profiles. This system is not only useful for new customers to choose a worthy restaurant, but it can be the reason for the success or the failure of an activity, according to its reviews. The massive amount of data collected can therefore generate huge amounts of information and assets. Analyse which restaurants failed, where, and possibly why, creates new knowledge and can be very helpful for a huge variety of people.

Our project has its roots exactly here. We aimed at creating a tool to analyse how the restaurant activities in one city perform. Using the Yelp Dataset (2018) we were able to investigate the past and ongoing activities in every city of the USA. Through our model we were able both to show the distribution of restaurants in some areas, taking into account different type of cuisines, and to predict the closure of a restaurant by using two regression models.

The implementation of such a project will benefit a great variety of people. First, predicting the risk of closure of a business will benefit the owner himself, as he will be able to know whether it is worth creating a business with specific attributes in a determined location. Further, potential employees, such as chefs and waiters, will be able to assess their job choices based on the risk of closure. Moreover, this project can help restaurant lenders, such as banks and investors. Knowing the risk of closure of a business, they will be able to decide whether they should invest in that business. Benefitting from the implementation of this project, the government could adapt its tax regulations in order to help them survive and avoid bankruptcy, if not to succeed. Therefore, it turns out that the implementation of such a project not only has a social impact, but an economic one as well.

Project Setup

The starting point of our project consisted in the setting up of our work environment. Since we decided to give our work the structure and the layout of a data analysis project, we needed to settle how we were going to proceed as a team and how we wanted to manage our data.

The team was given a project manager who needed to outline and supervise the work that had to be done. We decided that it was better for us to work following an Agile process, so that each task was dynamically allocated and the changes to the design of our project could be better handled. We had to set up requirements which were going to give the boundaries of our work and the tasks to be carried out to complete the project. In order to be sure to address our work correctly, we identified four users that could be interested in our data analysis: restaurant owners, workers, investors and the government. The requirements were then identified on these users and their needs if they were to request our tool. On this regard, the first three users should be able to estimate whether it is worth it to, respectively, open, work or invest in a certain type of restaurant in a specific area; the last one, though, must be able to monitor the situation of an area to define appropriate policies. We knew that our analysis should have also been able to predict the outcome of a restaurant and show relevant statistics concerning the business activities of each region. We then set a work breakdown structure to be aware of each of the macro tasks needed to deliver the completed project.

The first task to be completed was the Data Management Plan. For a complex project it is crucial to have a well settled method to handle the data so that every process concerning the core of the work is clear and organized. We therefore had to precisely lay out our data lifecycle both during and after our project was finished. Our data can be retrieved from Kaggle and it is called Yelp Dataset (2018). The original dataset has a dimension of 9 GB since it incorporates many information that does not serve our purpose (such as images). We planned to deliver updates every quarter year, even if the original data is renewed every time a new business becomes operational. The reason behind this last choice is that we want to carry out only substantial updates that can have an impact on the data analysis. Each time the data gets updated, we will release a version of the source files, so that the structure of the filesystem will always be the same and older versions can be accessed easily. Following the 3-2-1 rule we will be having copies of our data. The main one will be stored in the department's drive so that we can easily have access to it (reading and writing) and we minimize the risk of loss and theft. A second copy of the data will be stored in a cloud account and another one on an external hard drive, to ensure transferability of the data. At the end of the project, we will keep only the last version of the dataset used for our purposes for a maximum of one year (accordingly to Yelp's regulations). The data will be transferred by us at the end of the analysis process to the Open University Data Repository in order to let other researchers benefit from the project. As we have already mentioned, Yelp poses some restrictions to who can access the data. During the project we are only able to share the data metrics and summaries amongst the group members and the supervisor. At the end of the project we are allowed to publish an academic article regarding the data

without limitations, but the publishing of the data to any platform should be restricted by the licensing in cooperation with Yelp, and therefore anyone willing to access it should agree to their terms and conditions.

Once the DMP was finished, we could proceed with the other tasks of the work breakdown structure. At first, we completed the data cleaning and preparation by removing the non-useful instances (such as all the images) and by creating a density variable which pointed out how many restaurants of a certain type could be found in a defined area. The original data, since it is directly compiled by Yelp users (restaurants or customers), was very heterogeneous and messy, and therefore we had to deliver many improvements during the cleaning process. For instance, we removed every business record that did not contain information about the postcode. The reason behind this is that, since a specific business does not have a postcode, it means that it cannot be approached and, therefore, proved that it exists. The resulting dataset is a single 30 MB CSV file, consisting of approximately 160 thousand records organised in 20 attributes, easier to manage than the one we had at the beginning.

In the following chapters we will go through the other Work Breakdown Structure's tasks, explain their implementation and discuss the results. We will begin with the Exploratory Data Analysis, then pass on to the Data Analytics and the regression model and finally we will see the Data Visualization. We will then proceed to evaluate the whole model.

Data Analysis

Now that the cleaning and preparation of the initial dataset has finished, the next step is to perform an exploratory data analysis of the existing data, in an effort of gaining an insight of it, discovering any patterns, determining any existing relationships between the variables, examining the direction and the size of these relationships and possibly reducing the dimensionality of our data. For this purpose, the following procedures are involved:

- Graphical Analysis (Histograms, Correlation Analysis)
- Principal Component Analysis

Starting with the Graphical Analysis, the following Figures (Figures 1-3) illustrate histograms for the variables “Average Working Hours”, “Stars” and “Density”:

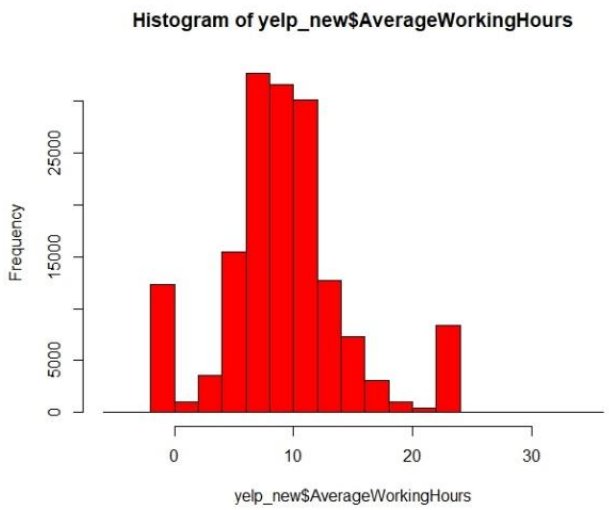


Figure 1 Average Working Hours Histogram

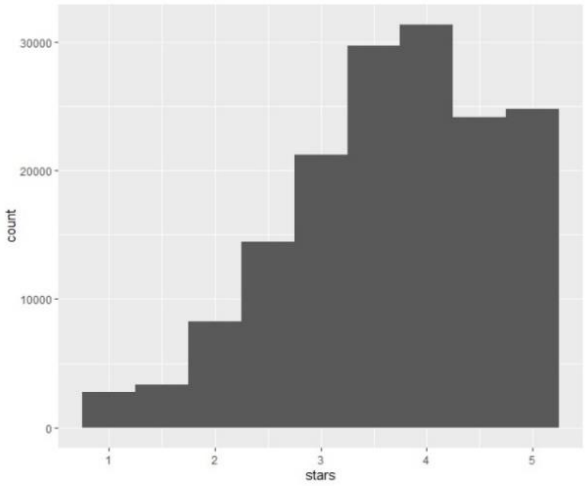


Figure 2 Stars Histograms

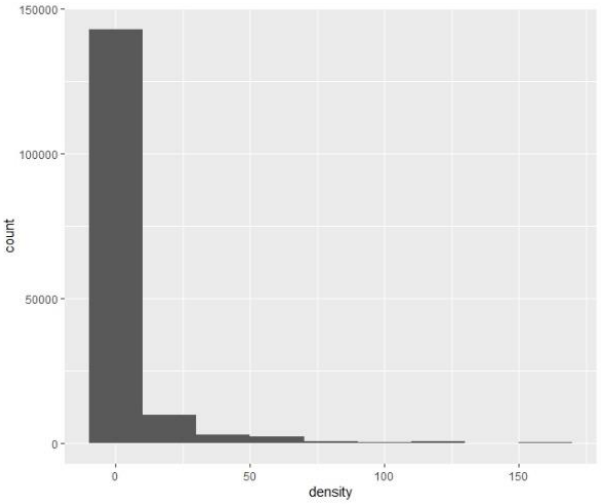


Figure 3 Density Histogram

A look in the above histograms enables us to make the following observations regarding the data:

- Average Working Hours: The curve of this variable resembles a normal distribution, with the great majority of the businesses having a weekly average of 8 working hours.
- Stars: The distribution of the stars variable is left-skewed, with most of the US businesses being rated with 4 stars. This fact proves the high level of the businesses that are opened in the USA.
- Density: The distribution of the business density is right-skewed. Many of the US states contain a small number of businesses for each category. This fact proves that the business competition forces the owners to search for areas which do not contain many restaurants of the same category as the one they intend to open.

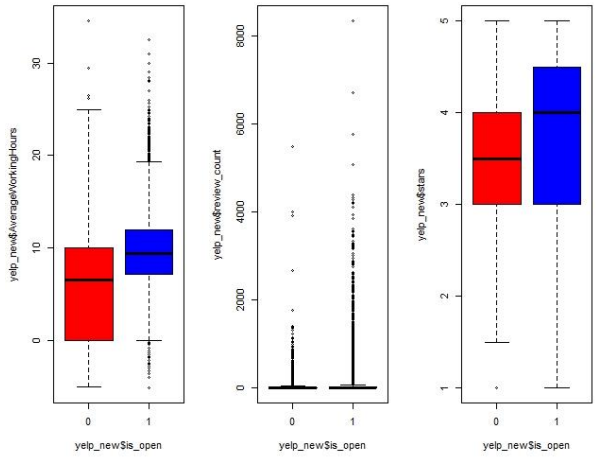


Figure 4 Boxplots

Additionally to the histograms, boxplots for the variables “AverageWorkingHours”, “ReviewCount”, “Stars” grouped by the variable “is_open” variable are also plotted and illustrated in Figure 4. Starting with the AverageWorkingHours variable, we can observe that the boxplot for the closed businesses is wider than the one for the open ones. However, the latter boxplot contains a greater number of outliers. Moreover, the median value for the closed restaurants is slightly lower than the one for the open businesses.

As for the variable for the number of reviews, we cannot make many observations. The only thing that we can say is that both open and closed restaurants contain a great number of outliers, of which the great majority lies in the open restaurants. Finally, in the boxplots for the stars variable, we can see that the graph for the open restaurants is wider than the one for the closed ones, with a significantly higher median value as well. Moreover, we can observe just one outlier in the boxplot of the closed activities.

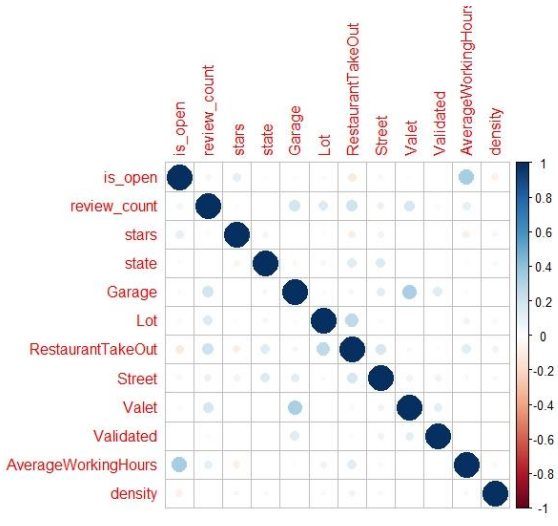


Figure 5 Correlation Matrix

The next step in our Data Analysis for this coursework is the Correlation Analysis, in order to gain an insight of strong and weak correlations between the variables. For this purpose, the following Figure (Figure 5) shows a correlogram of the explanatory variables. The positive correlations are coloured in blue, while the negative ones in red. The size and the colour intensity of the circles indicate the strength of the correlation. As we can observe in the correlation matrix, no variable is significantly correlated with the is_open variable. However, among these variables, the average amount of working hours seems to affect the outcome of the is_open variable more significantly than the remaining variables in a negative way. The star rating, instead, affects the outcome negatively, which means that a better business rating increases the closure probability, a fact which indeed sounds surprising, according to

the common sense. On the other hand, we can barely observe that there is a positive correlation between the density and the is_open variable, while the latter is positively affected by the presence of a Take-out service in the business.

The final step of the Data Analysis phase for our coursework is the Principal Component Analysis, the purpose of which is to reduce the dimensionality of our data, which is too large, as the current form of the dataset consists of 11 variables. After performing the Principal Component analysis, we calculated the loadings for each principal component. The following table provides the variable loadings for each Principal Component. The largest loadings for each Principal Component are highlighted in orange:

	PC1	PC2	PC3	PC4	PC5	PC6	PC7	PC8	PC9	PC10	PC11	
review_c	0.456248471	-0.102939573	0.235291037	-0.069581203	0.098612012	-0.321837175	0.094140596	0.091817165	-0.747620367	0.077712437	-0.159579397	
stars	-0.026863951	0.237627116	0.236300337	-0.705120638	-0.042016732	-0.110037037	0.366478636	-0.429749012	0.07436602	-0.038899444	0.223317908	
state	0.163797413	-0.061488594	-0.59130302	-0.018275013	-0.374737051	-0.051052519	-0.244718107	-0.604141053	-0.176351903	0.017435322	-0.141791251	
Garage	0.437963069	0.403604316	0.052185596	0.190501817	0.011674136	-0.135086971	-0.043786119	-0.022245519	0.198100385	-0.736631178	-0.060095835	
Lot	0.230194904	-0.453329353	0.355145169	-0.250477891	0.171112004	0.231575261	-0.314965447	-0.202427853	0.296024005	-0.046501556	-0.489649104	
Restaurai	0.418426762	-0.468302401	-0.16550082	-0.123948266	0.070720649	0.082571537	-0.093149162	0.116049003	0.092953029	-0.096891922	0.714749113	
Street	0.303846633	0.107595031	-0.43258977	-0.417648493	-0.146879527	0.006154816	0.207415363	0.529711657	0.230148296	0.158275207	-0.333601607	
Valet	0.41055944	0.382467482	0.158545658	0.201521761	0.096961099	-0.160982598	-0.218653376	-0.148246858	0.303053363	0.636157911	0.133305346	
Validated	0.224218887	0.26272708	0.035813024	0.057665785	-0.009963916	0.879475299	0.160335534	-0.06506702	-0.262357181	0.052597399	0.031263371	
AverageV	0.181116743	-0.334311503	0.148542034	0.390962984	-0.378940154	-0.047837582	0.668066471	-0.149788126	0.229706605	0.072715994	-0.088485288	
density	0.021084479	-0.052887613	-0.38379829	0.109623883	0.799371894	-0.035974429	0.35255646	-0.240245589	0.050132362	0.012870761	-0.111612403	

Figure 6 Principal Component Loadings

Additionally to the calculation of the loadings, we also plotted a bar chart for the percentage of explained variance of each principal component and a line chart showing the cumulative percentage of explained variance as we progressed from the first to the last principal component. These charts are shown in the following figures (Figures 7-8):

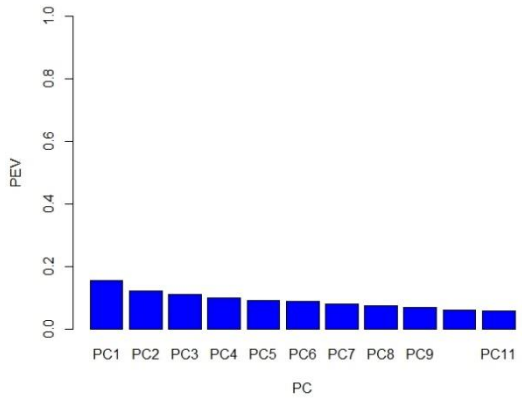


Figure 7 PEV Barplot

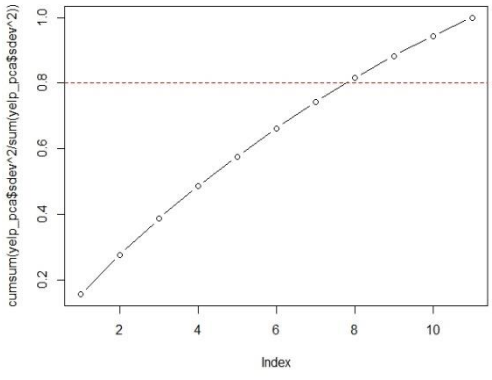


Figure 8 Cumulative PEV Line Chart

As we can observe in the barplot of Figure 7, Principal Component 1 explains the largest percentage of the dataset variance, a percentage approximately equal to 15%. This percentage drops as we move towards the last principal component, which explains only 5% of the total information. Our initial purpose was to reduce the dimensionality of the data used by the machine learning methods and explain as much proportion of total variance as possible. Therefore, we chose to set the cumulative variance proportion to 80%. By looking at the line chart of Figure 8, we can observe that this proportion is covered by the first 8 Principal Components. Therefore, we have proceeded to the Machine Learning methods by including only these 8 Principal Components in our training formula.

Data Analytics

The research question that was defined in the introduction chapter and the purpose of this research aimed at understanding whether there is a way to predict the probability of closure for a new business in the United States. In this chapter, an attempt to predict this probability, as well as the closure state of a business (remaining open or closing), is presented. For this purpose, we are going to leverage two different machine learning methods, the logistic regression and the k-NN nearest neighbour algorithm. A logistic regression is the appropriate analysis to conduct when the dependent variable is binary and when having to do with questions requesting the probability of a specific binary outcome, such as in our case. On the other hand, the advantages of the k-NN nearest neighbour are numerous. A first advantage is the speed in which the algorithm is trained, that is crucial in our case, especially if the size of the dataset is considered. In both machine learning methods, we are going to use a 70/30 training/test split of the dataset. In the case of the logistic regression, both the probability of a business remaining open and the outcome of the business are esimated. In each case we are going to compare the predicted values with the actual ones and calculate the method-s accuracy.

```
Call:
glm(formula = ml_formula, family = "binomial", data = ml_training)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-4.6397   0.1302   0.4834   0.6644   2.3954

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)  1.690107   0.009552 176.939  < 2e-16 ***
PC1          0.024050   0.006567   3.662  0.00025 ***
PC2         -0.136412   0.007092 -19.234  < 2e-16 ***
PC3          0.402374   0.007418  54.242  < 2e-16 ***
PC4          0.362033   0.008161  44.359  < 2e-16 ***
PC5         -0.612043   0.008309 -73.658  < 2e-16 ***
PC6         -0.219391   0.008611 -25.478  < 2e-16 ***
PC7          1.046707   0.010272 101.894  < 2e-16 ***
PC8         -0.328885   0.008887 -37.006  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 115881  on 111924  degrees of freedom
Residual deviance:  95146  on 111916  degrees of freedom
AIC: 95164

Number of Fisher Scoring iterations: 5
```

Figure 9 Logistic Regression

Logistic Regression

The first machine learning method we will apply is the logistic regression model. As mentioned above, we are going to include the first 8 Principal Components as explanatory variables in the training formula and the "is_open" variable will be used as the response variable. The following figure (Figure 9) illustrates a summary of the logistic regression model trained with the above formula. According to the summary, we can extract the following conclusions about the regression model:

- An increase in the first principal component by one unit will increase the log odds of a business remaining open by 0.024, and its p-value indicates that it is significant in determining the probability.
- An increase in the second principal component by one unit causes a decrease in the log odds by 0.1364, while a p-value equal to 0.00025 indicates its significance in determining the probability.
- An increase in the third principal component by one unit will increase the log odds of a business remaining open by 0.4, and its p-value indicates that it is significant in determining the probability.
- An increase in the fourth principal component by one unit will increase the log odds of a business remaining open by 0.362, and its p-value proves its significance in determining the probability.
- An increase in the fifth principal component by one unit will decrease the log odds of a business remaining open by 0.612, and its p-value indicates that it is significant in determining the probability.
- An increase in the sixth principal component by one unit will decrease the log odds of a business remaining open by 0.219, and its p-value indicates that it is significant in determining the probability.

- An increase in the seventh principal component by one unit will increase the log odds of a business remaining open by 1.047, and its p-value indicates that it is significant in determining the probability.
- An increase in the first principal component by one unit will decrease the log odds of a business remaining open by 0.329, and its p-value indicates that it is significant in determining the probability.

The difference between the Null deviance and the Residual deviance tells us that the model is good fit.

Afterwards, we applied our prediction model to the records of the test set. This model predicts the probability of a business remaining open. Therefore, we additionally predict the outcome of a business by simply rounding the predicted value to the closest binary value (0/1), in order to estimate the accuracy of the model. The accuracy of the model turned out to be close to 85%, as it was estimated through the comparison of the predicted outcomes against the real values of the test set. The level of the accuracy allows us to answer the research question posed in the first chapter, by safely stating that we can predict the probability of a business remaining open or closing with the use of a regression model.

k-Nearest Neighbour

As an addition to the logistic regression model, we also trained and applied the k-Nearest Neighbour algorithm to predict the outcome of the business closure. Firstly, we trained the algorithm with the use of a k-value equal to the square root of the training set length. Thus, since the training set contains 9 attributes, the k-value will be equal to 3. In the next chapter, we will evaluate the performance of the algorithm with the use of different k-values and different training and test set splitting. Afterwards, we applied the algorithm to the records of the test set and then compared the predicted values against the existing real values of the it. The prediction accuracy of the algorithm stood at similar levels as the logistic regression model, reaching a peak of 85%. The high level of the prediction algorithm provides sufficient proofs in order to safely state that the k-Nearest Neighbour algorithm can also be used for the prediction of a business closure state.

Evaluation

In this section we are going to evaluate the performance of the two machine learning methods used in the previous section. For this purpose, we are going to perform a k-cross validation for both machine learning methods. In each iteration, the Root Mean Square Error value is going to be

calculated and stored in a matrix. Because we expect the matrix which stores the values to be too large, we are going to make sense of the data with the use of visualizations. Finally, an evaluation of the proposed solution in terms of customer requirements is also included in this section.

Logistic Regression Evaluation

In the case of the Logistic Regression, we will perform a 10-cross validation for different training and test sizes. More specifically, we will increase the training set size from 10% to 90% of the total dataset by 10% each time. In each iteration, we will calculate and store the RMSE value for each iteration, which means that we are going to calculate 100 RMSE values in total. We monitored the variation of the median RMSE value depending on the length of the training set. To achieve this, we are going to use the line chart illustrated in the figure on the left side (Figure 10), where the x-axis represents the size of the training set as proportion of the total size of the dataset, while the y-axis shows the median RMSE value. A look in the line chart enables us to observe that the median RMSE follows an unstable trend, marking its minimum value for a training set size equal to 40% of the total dataset.

Figure 10 Median RMSE line chart

k-Nearest Neighbour Evaluation

The evaluation of the k-Nearest Neighbour algorithm requires a greater amount of work, compared to the evaluation of the Logistic Regression Model. The first part of the assessment will be similar to the one performed for the purposes of the Logistic Regression Evaluation. More specifically, we are going to plot the trend of the median RMSE value depending on the training set size and we are going to spot the dimension of the training set for which the median RMSE reaches its minimum value. Afterwards we are going to plot the trend of the median RMSE depending on the k-value used in the algorithm. In order to achieve that, we are going to calculate the median RMSE for different k-values ranging from 1 to 10. Finally, we are going to use the line chart to spot the k-value for which the median RMSE reaches its minimum.

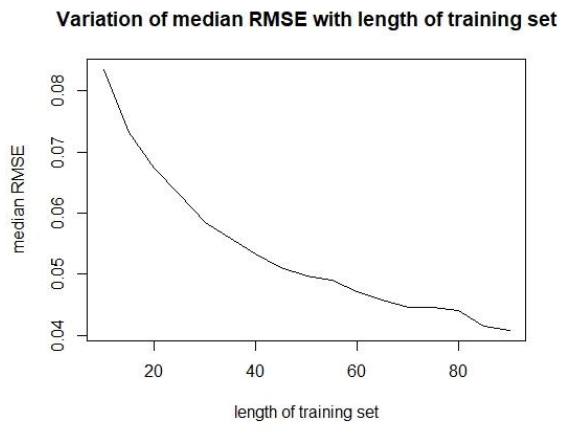


Figure 11 Median RMSE variation with size of training set

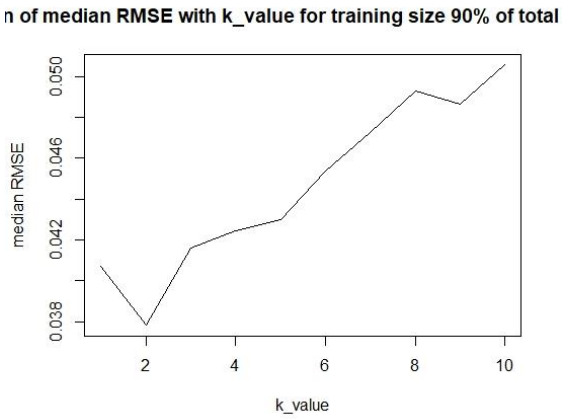
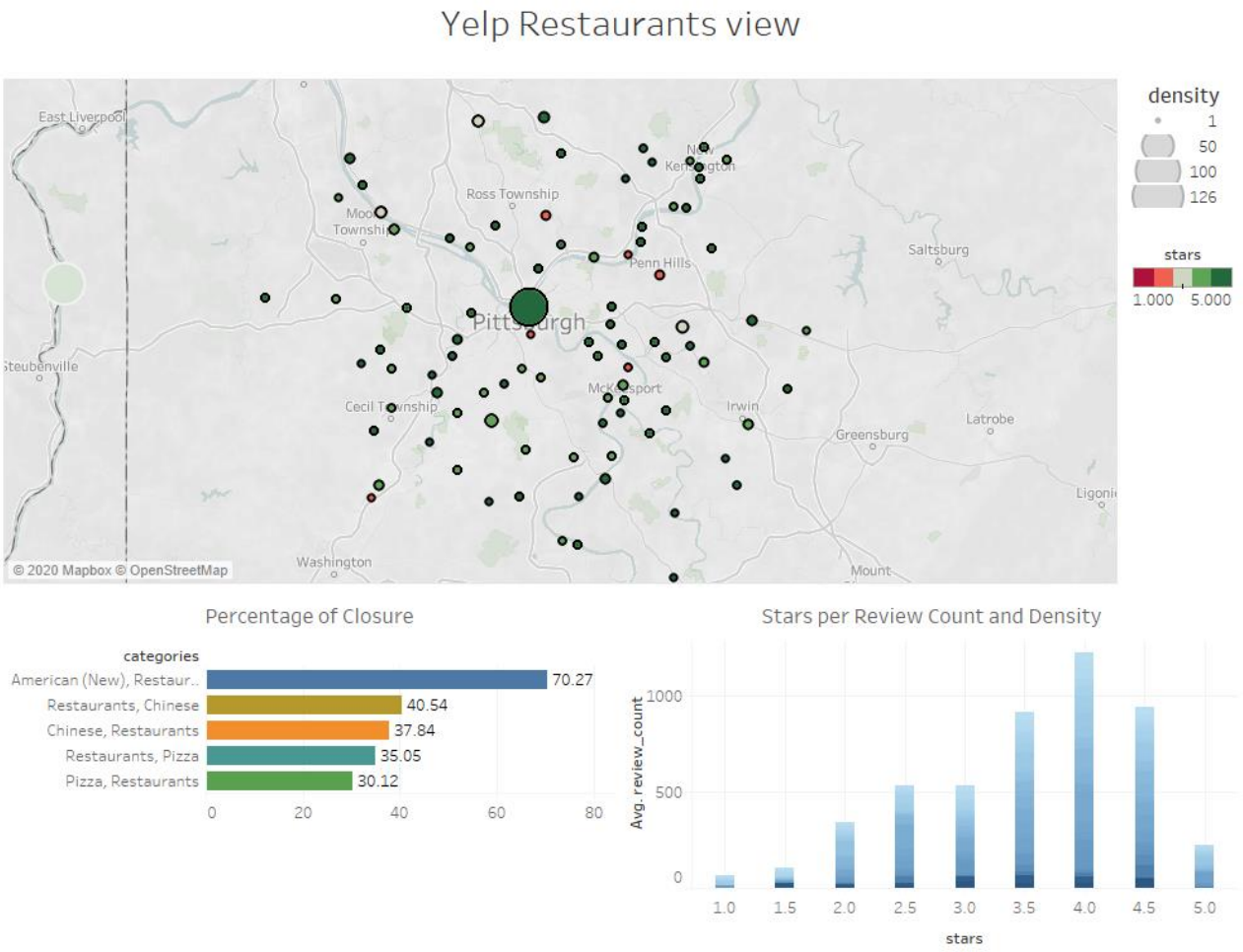


Figure 12 Median RMSE variation with k-value

The first figure (Figure 11) shows the median RMSE variation depending on the size of the training set. As we can clearly observe in the line chart, the median RMSE begins with a value slightly greater than 0.08 for a training set size equal to 10% of the total dataset. As the size of the training set increases, we can see that the corresponding value drops significantly, reaching its minimum value for a training set size equal to 90% of the total data. Therefore, we have chosen the 90/10 training/test split in order to examine the median RMSE trend for different k-values of the algorithm. This chart is illustrated in the following figure (Figure 12). As we can observe, the median RMSE begins with a value of approximately 0.041. An increase of the k-value by one unit causes the corresponding value to drop and reach its minimum, which is slightly lower than 0.038. From this and onwards, the median RMSE constantly rises as the k-value increases and reaches the amount of 10, which is the maximum of this chart. Overall, the smallest median RMSE was noticed for a training set size equal to 90% of the total variation and for a k-value equal to 2.

Data Visualization



The Data visualization was achieved by using Tableau with the idea of creating a single view which could fit all our users' needs. The view had to be simple and pleasant to read, since it is addressed to non-experts in the data analysis field. We therefore put together three graphs that show the star rating and the percentage of closing in each field. The first visualisation, on the top part of the dashboard, is a map. It has the advantage of being involving for the user, since it shows something very familiar. The downside, though, is that it must show as little information as possible, in order not to become overwhelming. As we can see from the example below (a visualisation for the Pittsburgh, PA area), each town is represented by a bubble. The size of the bubble symbolises how many restaurants are in the area. The colour of the bubble, instead, displays the mean rating of these restaurants through a very common colour scheme: red means under the average, green means over the average. To understand more about the star rating of the area, on the bottom left there is a bar chart showing how many reviews, on average, assign how many stars, divided per density area. This simple graph shows that the review count follows a normal curve and that most restaurants, independently by how dense the area is, have a four stars rating. This is probably relatable to the fact that people tend to spend

time reviewing what they appreciated more. It is therefore mainly addressed at government and investors, and it suggests looking not only at the rating, but also at the number of ratings. This because not only are one-star restaurants bad, but also it could be that a four-stars restaurant is better than a five-stars one, due to the high number of reviewers of the first one. Finally, the graph on the bottom left side clearly and simply summarizes the percentage of restaurants closed, in the selected area, per type. This tool is very useful for each user who wants to understand how a certain kind of restaurant is performing in a specific area.

Project Evaluation

We must now review and evaluate the project base on the customer requirements set at the beginning, in order to be able to decide whether and at what percentage those requirements were met. The user requirements were addressed with the use of multiple machine learning methods and data visualisation.

- A business owner must be able to estimate whether the business he intends to start in a specific area will result in a failure or success. It is evident that this requirement was achieved. As we have seen in the previous chapters, the inclusion of location variables was crucial to pinpoint this requirement both in the machine learning parts and in the data visualisation one.
- The investors must be able to assess whether it is worth investing in a specific restaurant. Supposing that the prediction of a business closure is very absolute, the logistic regression model that was introduced during the data analysis additionally predicts the probability of a business closure. After having considered the insight of this probability, estimating whether it is worth investing in a specific business is at the discretion of each investor. As we have seen in the visualisation, then, there are tools that help the investors understand where the best restaurants are and how to spot them. Therefore, this requirement is clearly satisfied.
- A potential business employee must be able to know whether it could be a good option to look for a job in a specific area. This requirement is similar to the first mentioned one, therefore it is satisfied.
- The government must be able to monitor the situation of an area in order to define appropriate economic policies. Knowing that each state is governed by a specific governor and that a tax policy can be the same for all the cities in the same state, the use of the state variable in the machine learning models is enough to satisfy the specific requirement. In

addition to that, the density variable, which represents the number of businesses of the same category within the same state, is also included in the machine learning model and can contribute towards satisfying this requirement.

Finally, we must mention the functional requirements. At the beginning we stated that both the outcome of a business and the probability of a business closure should be estimated through this project. The requirements are clearly satisfied, as it was analysed in the previous chapters, with the use of the logistic regression model and the k-Nearest Neighbour algorithm. In addition to this, in the data visualization there are some relevant statistics, concerning distribution, proportion and business activities, which complement the requirements.

Authorship Contribution

F.R acted as a project manager, designing, supervising and coordinating the project. M.U. did the data preparation and cleaning and the exploratory data analysis, G.E. did the Data Analytics with machine learning methods and the performance evaluation. J.M. did the data visualization. F.R and G.E did the project evaluation against the customer requirements and took part in writing the article.

Appendix

Data Management Plan for Research Students

1. **Overview**

Researcher: Federico Riva, Georgios Efthymiadis, Mertkan Usta, Jack Mills
Project title: Yelp Business Data Analysis
Project duration: 2 Months
Project context: The project aims to answer to the following research question:

Can we predict the risk of closure of a business based on its star rating, the density of similar businesses in the same district/city, its category and its attributes?

2. Defining your data/research sources

2.1 Where will your data/research sources come from?

The dataset we are going to use is the Yelp dataset, which was retrieved from ***www.kaggle.com***. An alternative source for the same dataset is the official website of Yelp (***www.yelp.com/dataset***).

2.2 How often will you get new data?

The Yelp dataset is renewed every time a new business starts its operation. Because we wanted enough new businesses to have opened in every data collection process, we agreed as a team to collect new data once every quarter of a year.

2.3 How much data/information will you generate?

The total size of the Yelp Dataset is approximately 9 GB. The reason behind this gigantic size is the inclusion of pictures for every business. After the data cleaning and preparation process, we estimate that the final dataset used for data analysis will be around 350-400 MB.

2.4 What file formats will you use?

The Yelp dataset comes in the form of JSON files. With the use of pandas library, which is a built-in library of Python programming language, we are going to convert the JSON files into CSV format.

3. Organizing your data

3.1 How will you structure and name your folders and files?

The dataset we are going to use consists of 2 files. The first file is called “business” and it will be a record of the existing restaurants. The second file is called “review” and it will contain statistics about the total number of reviews concerning a specific restaurant. Apart from that we can append the time period in the name file, for example businessQ1_2020, reviewQ4_2020 and create a folder for each dataset

type. The files will be sorted by the name, and therefore by the time period during which they were collected.

3.2 What additional information is required to understand each data file?

The attributes of the dataset are going to be named in such a way that they will be understandable by everyone. Moreover, we are going to create a documentation, in which each attribute purpose will be explained.

3.3 What different versions of each data file or source will you create?

We will create a new version every quarter year, alongside the uploading of new data. As mentioned above, we will append the time period in which the data was collected to the name of each file. In that way, we can distinguish the files and create a hierarchy of them based on the time period.

4. Looking after your data

4.1 Where will you store your data?

We are going to keep our primary copy of the data in the department's drive. That way, we can minimize the risk of data loss or theft. From there we will be able to access it in order to deliver updates or use it for a project.

4.2 How will your data be backed up?

Following the 3-2-1 rule, we are going to keep three copies of our data. The primary copy was mentioned in question 4.1. The department's drive is backed up every night. This backup is kept on site for 35 days. An additional backup will be run every 4 weeks and will be kept off-site for 175 days (~6 months). Additionally, we are going to keep a second copy in a cloud account (Google Drive, Amazon Drive or Dropbox). The final copy of the data will be kept in an external hard drive. That way, we can ensure the transferability of our data. The cloud backup will be renewed using this copy.

4.3 How will you test whether you can restore from your backups?

The copy we keep in the department's drive is frequently backed up. As for the rest copies, we are going to periodically monitor the date on which the data was backed up for the last time and ensure that they are not corrupted or destroyed.

5. Sharing your data

5.1 Who owns the data you generate? According to the Yelp dataset terms of use, Yelp Inc owns the data generated.
5.2 Who else has a right to see or use this data? Being in accordance with the terms and conditions set by Yelp Inc, data access will be restricted to group members and our supervisor only.
5.3 Who else should reasonably have access to this data when you share it? According to the Yelp dataset terms of use, we are not allowed to create, redistribute or disclose any summary of/metrics related to the Data to any third party or on any website or other electronic media. However, as part of an academic project, as of now, we are allowed to publish an academic article concerning the use of the Data without any limitation.
5.4 What should/shouldn't be shared and why? As mentioned inside the Yelp dataset terms of use, none of the Data can be shared with people who are not members of the team.

6. Archiving your data

What should be archived beyond the end of your project? Beyond the end of the project, we are going to archive the final form of the data which were used for the purposes of our data analysis.
6.2 For how long should it be stored? According to the Yelp dataset terms of use, we are required to possess and store the data for a maximum of one year. Upon the termination of the agreement, we are required to remove all instances of the data.
6.3 When will files be moved into the data archive/repository? The data will be archived once we have completed our data analysis.
6.4 Where will the data be stored? Guided by the groups of people which could benefit from the implementation of this project, we are going to choose the Open University Data Repository (ORDO/Open Research Data Online, ***https://ou.figshare.com***).
6.5 Who is responsible for moving data to the data archive and maintaining it?

We are responsible for moving our data to the selected data archive. Once our data is archived, it is the repository who maintains our data.
6.6 Who should have access and under what conditions? Due to the restrictions set by the Yelp dataset terms of use, the data cannot be publicly available. The data will be licensed in cooperation with Yelp Inc. and anyone who requests access to the data should agree to the terms and conditions set by Yelp Inc.

7. Executing your plan

7.1 Who is responsible for making sure this plan is followed? The project manager (Federico Riva) is responsible for making sure that every action taken by the rest members of the project team is according to the plan.
7.2 How often will this plan be reviewed and updated? The project plan will be reviewed internally on weekly meetings between project members which are held every Monday, and externally with our supervisor on weekly meetings held every Tuesday.
7.3 What actions have you identified from the rest of this plan?
7.4 What further information do you need to carry out these actions?

Source Code

The following chapter includes all the code written for the purposes of our data analysis. The code for the data preparation and cleaning was written using Python, while the rest of the data analysis was performed using R.

Convert JSON to CSV (Python)

```
#The initial form of the data was in JSON file format.
#We used the pandas library of Python to convert the JSON files to CSV format.

import pandas as pan
import numpy as nump
import math
import json
df = pan.read_json('yelp_academic_business.json',lines=True)
df.to_csv('yelp_academic_dataset_business.csv')
df=pan.read_json('yelp_academic_dataset_user.json', lines=True,
      chunksize=200)
i=1
```

```
for d in df:
      a=d.loc[:,['average_stars','compliment_cool','compliment_cute','c
      ompliment_funny','compliment_hot','compliment_list','compliment_m
      ore','compliment_note','compliment_photos','compliment_plain','co
      mpliment_profile','compliment_writer','cool','elite','fans','funn
      y','name','review_count','useful','user_id','yelping_since']]
      a.to_csv('yelp_academic_dataset_user_'+str(i)+'.csv')
      i=i+1
i=1
maxi=i+999
df=pan.read_csv('yelp_academic_dataset_user_'+str(i)+'.csv')
for b in range(1,9):
      for a in range(i,maxi+1):
            print(str(a))
            df2= pan.read_csv('yelp_academic_dataset_user_'+str(a)+'.csv')
            if (a!=1):
                  df= pan.concat([df,df2], ignore_index=True)
            df.to_csv('yelp_academic_dataset_user_part'+
                        str(int(maxi/1000))+'.csv')
            i=i+1000
            maxi=i+999
df = pan.read_csv('yelp_academic_dataset_user_'+str(i)+'.csv')
while i<=8186:
      if i!=(maxi-999):
            print(str(i))
            df2= pan.read_csv('yelp_academic_dataset_user_'+str(i)+'.csv')
            df= pan.concat([df,df2], ignore_index=True)
      i=i+1
df.to_csv('yelp_academic_dataset_user_part'+str(int(maxi/1000))+'.csv')
df = pan.read_csv('yelp_academic_dataset_user_part_'+str(i)+'.csv')
for a in range(1,10):
      print(str(a))
      df2= pan.read_csv('yelp_academic_dataset_user_part_'+str(a)+'.csv')
      df= pan.concat([df,df2], ignore_index=True)
      i=i+1
df.to_csv('yelp_academic_dataset_user.csv')
df = pan.read_json('yelp_academic_dataset_checkin.json',lines=True)
df.to_csv('yelp_academic_dataset_checkin.csv')
df = pan.read_json('yelp_academic_dataset_rewiew.json',lines=True)
df.to_csv('yelp_academic_dataset_rewiew.csv')
df = pan.read_json('yelp_academic_dataset_review.json', lines=True,
chunksize=200)
i=1
for d in df:
      a=d.loc[:,['business_id','date','stars']]
      a.to_csv('yelp_academic_dataset_review_subpart'+str(i)+'.csv')
      i=i+1
i=1
maxi=i+4774
df = pan.read_csv('yelp_academic_dataset_review_subpart1.csv')
for a in range(1,8):
      for b in range(i,maxi+1):
            print(str(b))
            df2= pan.read_csv('yelp_academic_dataset_review_subpart'+
                  str(b)+'.csv')
            if (b!=1):
                  df= pan.concat([df,df2], ignore_index=True)
```

```
      df.to_csv('yelp_academic_dataset_review_part'+str(int(maxi/4775))
      +'.csv')
      i=i+4775
      maxi=i+4774
df = pan.read_csv('yelp_academic_dataset_review_subpart'+str(i)+'.csv')
while i<=33430:
      if i!=(maxi-4774):
            print(str(i))
            df2=
pan.read_csv('yelp_academic_dataset_review_subpart'+str(i)+'.csv')
            df= pan.concat([df,df2], ignore_index=True)
      i=i+1
df.to_csv('yelp_academic_dataset_review_part'+str(int(maxi/4775))+'.csv')
df = pan.read_csv('yelp_academic_dataset_review_part1.csv')
i=2
while i<=8:
      df2=pan.read_csv(fn+'_part'+str(i)+'.csv')
      df=pan.concat([df,df2])
      i=i+1
df.to_csv('yelp_academic_dataset_review.csv')
```

Data Cleaning (R)

```
#The previous procedure created a number of empty attributes.
#They were removed from the dataset.

file_parts<-c("business","user","review","checkin","tip")
func<-function(a){
  return(paste("yelp_academic_dataset_",file_parts[a],".csv",sep=""))
}
businesses<-read.csv(func(1))
users<-read.csv(func(2))
reviews<-read.csv(func(3))
checkins<-read.csv(func(4))
tips<-read.csv(func(5))
businesses<-subset(businesses,select=-c(X))
users<-subset(users,select=-c(X,Unnamed..0,Unnamed..0.1))
checkins<-subset(checkins,select=-c(X))
tips<-subset(tips,select=-c(X))
reviews<-subset(reviews,select=-c(X,Unnamed..0))
func2<-function(b){
  return(nchar(func(b))-4)
}
c<-func2(1)
write.csv(businesses,paste(substr(func(1),1,c),"_subcleansed.csv",sep=""),row
.names = FALSE)
c<-func2(2)
write.csv(users,paste(substr(func(2),1,c),"_subcleansed.csv",sep=""),row.name
s = FALSE)
c<-func2(3)
write.csv(reviews,paste(substr(func(3),1,c),"_subcleansed.csv",sep=""),row.na
mes = FALSE)
c<-func2(4)
write.csv(checkins,paste(substr(func(4),1,c),"_subcleansed.csv",sep=""),row.n
ames = FALSE)
c<-func2(5)
```

```
write.csv(tips,paste(substr(func(5),1,c),"_subcleansed.csv",sep=""),row.names
= FALSE)
```

Data Preparation: Binary attributes extraction & AverageWorkingHours (Python)

```
#The dataset contains an attributes with information for the existence
#of a Take out service for example.
#This information is extracted into the attributes with binary values.
#Moreover, we used the Working Hours attribute to calculate
#a new one for the Average Working Hours of a business.

import pandas as pan
import numpy as nump
import math
import json
f_types=['business','user','review','checkin','tip']
df=pan.read_csv('yelp_academic_dataset_'+f_types[0]+'_subcleansed.csv')
whole_list={'Garage':[],'Street':[],'Validated':[],'Lot':[],'Valet':[],'Resta
urantTakeOut':[]}
for index, row in df.iterrows():
      attr_str=str(row['attributes'])
      if attr_str!="nan":
            attr_str=attr_str.replace("'",'"')
            attr_str=attr_str.replace('"{','{')
            attr_str=attr_str.replace('}"','}')
            attr_str=attr_str.replace(' True','"True"')
            attr_str=attr_str.replace(' False','"False"')
            attr_str=attr_str.replace('""','"')
            attr_str=attr_str.replace('"u"','"')
            jsf=json.loads(attr_str)
            if 'BusinessParking' in jsf:
                  b_parking=str(jsf['BusinessParking']).replace("'",'"')
                  if not (b_parking=='None' or b_parking=='{}'):
                        bpark=json.loads(b_parking)
                        if 'garage' in bpark:
                              if (bpark['garage']=='True'):
                                    whole_list['Garage'].append(1)
                              else:
                                    whole_list['Garage'].append(0)
                        else:
                              whole_list['Garage'].append(0)
                        if 'street' in bpark:
                              if (bpark['street']=='True'):
                                    whole_list['Street'].append(1)
                              else:
                                    whole_list['Street'].append(0)
                        else:
                              whole_list['Street'].append(0)
                        if 'validated' in bpark:
                              if (bpark['validated']=='True'):
                                    whole_list['Validated'].append(1)
                              else:
                                    whole_list['Validated'].append(0)
                        else:
                              whole_list['Validated'].append(0)
                        if 'lot' in bpark:
                              if (bpark['lot']=='True'):
```

```
                                    whole_list['Lot'].append(1)
                              else:
                                    whole_list['Lot'].append(0)
                        else:
                              whole_list['Lot'].append(0)
                        if 'valet' in bpark:
                              if (bpark['valet']=='True'):
                                    whole_list['Valet'].append(1)
                              else:
                                    whole_list['Valet'].append(0)
                        else:
                              whole_list['Valet'].append(0)
                  else:
                        whole_list['Garage'].append(0)
                        whole_list['Street'].append(0)
                        whole_list['Validated'].append(0)
                        whole_list['Lot'].append(0)
                        whole_list['Valet'].append(0)
            else:
                  whole_list['Garage'].append(0)
                  whole_list['Street'].append(0)
                  whole_list['Validated'].append(0)
                  whole_list['Lot'].append(0)
                  whole_list['Valet'].append(0)
            if 'RestaurantsTakeOut' in jsf:
                  rto=str(jsf['RestaurantsTakeOut']).replace("'",'"')
                  if not (rto=='None' or rto=='{}'):
                        if (rto=='True'):
                              whole_list['RestaurantTakeOut'].append(1)
                        else:
                              whole_list['RestaurantTakeOut'].append(0)
                  else:
                        whole_list['RestaurantTakeOut'].append(0)
            else:
                  whole_list['RestaurantTakeOut'].append(0)
      else:
            whole_list['Garage'].append(0)
            whole_list['Street'].append(0)
            whole_list['Validated'].append(0)
            whole_list['Lot'].append(0)
            whole_list['Valet'].append(0)
            whole_list['RestaurantTakeOut'].append(0)
daf=pan.DataFrame(whole_list)
daf.to_csv('TestingBusiness.csv')
df=pan.read_csv('yelp_academic_dataset_'+f_types[0]+'_subcleansed.csv')
li={'AverageWorkingHours':[]}
for index, row in df.iterrows():
      attr_str=str(row['hours'])
      days_week=['Monday','Tuesday','Wednesday','Thursday','Friday','Saturday
','Sunday']
      daily_ho=0
      if attr_str!="nan":
            attr_str=attr_str.replace("'",'"')
            jsf=json.loads(attr_str)
            for dw in days_week:
                  if dw in jsf:
                        if not (jsf[dw]=='None' or jsf[dw]=='{}'):
                              w_hours=jsf[dw]
```

```
                              separ=w_hours.index("-")
                              start_ho=w_hours[0:separ]
                              st_hour_sep=start_ho.index(":")
                              st_ho_ho=int(start_ho[0:st_hour_sep])
                              st_ho_min=int(start_ho[st_hour_sep+1:])
                              st_num_ho=st_ho_ho+(st_ho_min/60)
                              end_ho=w_hours[separ+1:]
                              end_hour_sep=end_ho.index(":")
                              end_ho_ho=int(end_ho[0:end_hour_sep])
                              end_ho_min=int(end_ho[end_hour_sep+1:])
                              if end_ho_ho<12:
                                    end_ho_ho=24+end_ho_ho
                              end_num_ho=end_ho_ho+(end_ho_min/60)

                              daily_ho+=(end_num_ho-st_num_ho)
                              print(dw+": "+str(end_num_ho-st_num_ho)+"
                              hours ("+w_hours+")")
                        else:
                              print(dw+":0 hours")
                  else:
                        print(dw+": 0 hours")
      else:
            for dw in days_week:
                  print(dw+":0 hours")
      li['AverageWorkingHours'].append(daily_ho/7)
      print("**********************")
daf=pan.DataFrame(li)
daf.to_csv('TestingBusinessHours.csv')
```

#Merge into final file (R)

```
#The file containing the business records and the one
#with the average working hours are merged into a single final dataset.

num_list<-read.csv("TestingBusiness.csv")
num_bus<-read.csv("TestingBusinessHours.csv")
num_list<-subset(num_list,select=-c(X))
num_bus<-subset(num_bus,select=-c(X))
busi<-read.csv("yelp_academic_dataset_business_subcleansed.csv")
busi<-subset(busi,select=-c(attributes))
mergie<-data.frame(busi,num_list,num_bus)
write.csv(mergie,"yelp_new.csv",row.names=FALSE)
```

#Density (R)

```
#This block of code calculates the density for each restaurant record.
#The density in our dataset is defined as the number of restaurants
#of the same category in the same city.

df = read.csv("C:/Users/georg/Desktop/yelp.csv")
trim <- function (x) gsub("^\\s+|\\s+$", "", x)
df$city <- trim(df$city)
df$density = 0

cities <- split.data.frame(df,df$city)
```

```
func2 <- function(x){
  x$density <- nrow(x)
  return(x)
}

func <- function(x){
  print(x$city)
  categories <- split.data.frame(x,x$categories,drop = TRUE)
  categories[] <- lapply(categories, func2)
  x <- dplyr::bind_rows(categories)
  return (x)
}

cities[] <- lapply(cities,func)
df <- dplyr::bind_rows(cities)
write.csv(df,"yelp_new.csv")
```

Exploratory Data Analysis, Machine Learning and Performance Evaluation (R)

```
install.packages("corrgram")
install.packages("ggplot")
install.packages("FNN")
library(corrplot)
library(corrgram)
library(ggplot2)
library(class)
library(FNN)


rmse <- function(m,o){
  sqrt(mean((m-o)^2))
}
yelp_new<-read.csv("yelp_new.csv")
nrow(subset(yelp_new,postal_code==""))

yelp_new<-subset(yelp_new,postal_code!="")
yelp_new<-subset(yelp_new,select=-c(business_id))


str(yelp_new)
hist(yelp_new$AverageWorkingHours,col = "red")
ggplot(yelp_new,aes(density))+geom_histogram(binwidth = 20)
ggplot(yelp_new,aes(stars))+geom_histogram(binwidth = 0.5)


par(mfrow = c(1,3))
boxplot(yelp_new$AverageWorkingHours~yelp_new$is_open,
        col = c("red","blue"))
boxplot(yelp_new$review_count~yelp_new$is_open,
        col = c("red","blue"))
boxplot(yelp_new$stars~yelp_new$is_open,
        col = c("red","blue"))


yelp_new_sub$state <- as.numeric(yelp_new_sub$state)
correlations <- cor(yelp_new_sub)
```

```
corrplot(correlations,method = "circle")

yelp_new_sub<-subset(yelp_new,select=-
c(address,categories,name,postal_code,city,longitude,latitude))
pca__subset <- subset(yelp_new_sub,select = -c(is_open))
pca__subset$state <- as.numeric(pca__subset$state)

str(yelp_new_sub)
yelp_pca<-prcomp(pca__subset, center = TRUE,scale. = TRUE)

plot(cumsum(yelp_pca$sdev^2 / sum(yelp_pca$sdev^2)), type="b")
yelp_pca_pev <- yelp_pca$sdev^2/sum(yelp_pca$sdev^2)
barplot(yelp_pca_pev,ylim = c(0,1),xlab = 'PC',ylab = 'PEV',pch = 20, col =
'blue',names =
c("PC1","PC2","PC3","PC4","PC5","PC6","PC7","PC8","PC9","PC10","PC11"),type =
"l",grid = TRUE)
line <- plot(cumsum(yelp_pca_pev),ylim = c(0,1),xlab = 'PC',ylab =
'Cumulative PEV',pch = 20, col = 'blue',names =
c("PC1","PC2","PC3","PC4","PC5","PC6","PC7","PC8","PC9","PC10","PC11","PC12",
"PC13"),type = "l")
abline(h=0.8,col = 'red',lty = 'dashed')

summary(yelp_pca)

pcaLoadings <- yelp_pca$rotation
pcaLoadings


#Machine Learning
ml_dataset <- data.frame(yelp_pca$x[,1:8],is_open = yelp_new_sub$is_open)
num_of_rows <- dim(ml_dataset)[1]
training_idx <- sample(num_of_rows,num_of_rows*0.7)
ml_training <- data.frame(ml_dataset[training_idx,])
ml_test <- data.frame(ml_dataset[-training_idx,1:8])
ml_test_nn <- ml_test
ml_real <- data.frame(ml_dataset[-training_idx,9])
ml_formula <- reformulate(names(ml_dataset)[1:8],response = "is_open")

#Logistic Regression
logistic <- glm(ml_formula,data = ml_training,family = "binomial")
summary(logistic)
is_open_prob <- predict(logistic,type = "response",newdata = ml_test)
ml_test$is_open_prob <- is_open_prob
ml_test$is_open <- ifelse(ml_test$is_open_prob>0.5,1,0)
confMatrix <-
data.frame(ml_test$is_open,ml_real$ml_dataset..training_idx..8.)ml
accuracyTb <- table(confMatrix)
accuracy <- sum(diag(accuracyTb))/sum(accuracyTb)
accuracy


#k-NN nearest
kNN_training <- ml_training
kNN_test <- data.frame(ml_test[,1:8],is_open = ml_test$is_open)
kNN_real <- ml_real
k_value <- sqrt(length(kNN_training))
knn_pred <- knn.reg(train = kNN_training,test = kNN_test,y =
kNN_training[,9],k_value)
```

```
results <- table(data.frame(kNN_real,round(knn_pred$pred)))
kNN_accuracy <- sum(diag(results))/sum(results)

#Logistic Regression Evaluation
log_eval_dataset <- ml_dataset
log_eval_num_of_rows <- dim(log_eval_dataset)[1]
RMSE.LR <- NULL
RMSE_LR.List <- list()
k <- 10
for (j in seq(10,90,10)){
  for (i in 1:k){
    log_eval_index <-
sample(log_eval_num_of_rows,(j/100)*log_eval_num_of_rows)
    train.log.eval <- log_eval_dataset[log_eval_index,]
    test.log.eval <- log_eval_dataset[-log_eval_index,1:8]
    real.log.eval <- log_eval_dataset[-log_eval_index,9]
    log.eval <- glm(ml_formula,data = train.log.eval,family = "binomial")
    log.eval.pred <- predict(log.eval,type = "response", newdata =
test.log.eval)
    log.eval.pred <- ifelse(log.eval.pred>0.5,1,0)
    RMSE.LR[i] <- rmse(log.eval.pred,real.log.eval)
    print(RMSE.LR[i])
  }
  RMSE_LR.List[[j]] <- RMSE.LR
}
Matrix.RMSE_LR <- do.call(cbind,RMSE_LR.List)
LR_RMSE_med <- colMedians(Matrix.RMSE_LR)
X_axis <- seq(10,90,10)
plot(LR_RMSE_med~X_axis,type = "l",xlab = "length of training set",ylab =
"median RMSE",main = "Variation of median RMSE with length of training set")

#k-NN Performance Evaluation
kNN.eval.dataset <- ml_dataset
kNN.eval.num_of_rows <- dim(kNN.eval.dataset)[1]
RMSE.kNN <- NULL
RMSE_kNN.List <- list()

for (j in seq(10,90,10)){
  for (i in 1:k){
    kNN.eval.index <-
sample(kNN.eval.num_of_rows,(j/100)*kNN.eval.num_of_rows)
    train.kNN.eval <- kNN.eval.dataset[kNN.eval.index,]
    test.kNN.eval <- kNN.eval.dataset[-kNN.eval.index,]
    real.kNN.eval <- kNN.eval.dataset[-kNN.eval.index,9]
    k_Value.eval <- sqrt(length(train.kNN.eval))
    kNN.eval.pred <- knn.reg(train = train.kNN.eval,test = test.kNN.eval,y =
train.kNN.eval[,9],k_Value.eval)
    RMSE.kNN[i] <- rmse(kNN.eval.pred$pred,real.kNN.eval)
  }
  RMSE_kNN.List[[j]] <- RMSE.kNN
}
Matrix.RMSE_kNN <- do.call(cbind,RMSE_kNN.List)
kNN_RMSE_med <- colMedians(Matrix.RMSE_kNN)
x_axis <- seq(10,90,10)
plot(kNN_RMSE_med~x_axis,type = "l",xlab = "length of training set",ylab =
"median RMSE",main = "Variation of median RMSE with length of training set")

RMSE_kNN2 <- NULL
```

```
RMSE.kNN2.List <- list()

for (k_val in seq(1,10)){
  for (i in 1:k){
    kNN_eval2.index <- sample(kNN.eval.num_of_rows,0.9*kNN.eval.num_of_rows)
    train.kNN.eval <- kNN.eval.dataset[kNN_eval2.index,]
    test.kNN.eval <- kNN.eval.dataset[-kNN_eval2.index,]
    real.kNN.eval <- kNN.eval.dataset[-kNN_eval2.index,9]
    kNN.eval2.pred <- knn.reg(train = train.kNN.eval,test = test.kNN.eval,y =
train.kNN.eval[,9],k_val)
    RMSE_kNN2[i] <- rmse(kNN.eval2.pred$pred,real.kNN.eval)
  }
  RMSE.kNN2.List[[k_val]] <- RMSE_kNN2
}
Matrix.RMSE_kNN2 <- do.call(cbind,RMSE.kNN2.List)
kNN2_RMSE_med <- colMedians(Matrix.RMSE_kNN2)
x_axis_2 <- seq(1,10)
plot(kNN2_RMSE_med~x_axis_2,type = "l",xlab = "k_value",ylab = "median
RMSE",main = "Variation of median RMSE with k_value for training size 90% of
total dataset size")
```

References

Yelp, Inc., 2018. Yelp Dataset. [online] Kaggle.com. Available at: <https://www.kaggle.com/yelp-dataset/yelp-dataset> [Accessed 21 February 2020].