****

**MODULE 3**

**AIDI 1003 – CAPSTONE TERM 1**

**AMIT MARAJ**

**AHNCH BALA 100424062**

**SONAKSHI KARKERA 100720763**

**SURBHI THAKUR 100732335**

**ARUN KALAESWARAN 100771700**

**December 6, 2019**

**Executive Summary**

The main goal of this project is to apply machine learning to predict a restaurant’s success based on the customer’s star rating and finding which restaurant features have the most impact on its average rating by performing sentiment analysis on the text of user reviews.

We’ll be using the dataset provided by Yelp as part of their Dataset Challenge 2019 to train and test our classification model. The dataset includes information about local businesses in 10 metropolitan areas across 2 countries.

Along with the main goal, we have two additional goals. The second goal is to help Yelp differentiate themselves from competitors. The review industry is a saturated market with many competitors vying for the top spot. Yelp, arguably one of the most popular review site in North America, would like to remain at the top by using the data they have in creative ways. Our model will help open a new revenue channel for Yelp in terms of business development. Our third and final goal is to ultimately help local restaurants achieve success within their city. Our model will let them be aware of any imminent closure, so that they can take the appropriate steps to rectify their business strategy.

**Introduction**

In the year 2018, nearly 147 restaurants in the Toronto region were shut down with no reason given (Daily Hive Staff, 2018). There could be many reasons due to their closure, whether it be poor customer service, inadequate location choice or ultimately the lack of marketing. The reasons and assumptions can be infinite, but with so many resources available why was it so hard for many restaurant owners to figure out where they went wrong? Could they have known beforehand that they were going to close?

As a customer, it is easy for us to go online and search for reviews of specific restaurants to ensure that our experience would meet our expectations. We can even go further and look for specific niches like ambience, music or even if its pet friendly. We can find all this information with a simple search on sites like Open Table or Google Reviews. But perhaps the most informative site of them all would be Yelp.

The satisfaction of customers is tied to the success of restaurants. According to Yelp, there has been an average of nearly 140 million users per month in Q2 of 2019 (Yelp, 2019). Furthermore, as of June 30th, 2019, there have been 192 million reviews contributed to the website (Yelp, 2019). This means that there is an enormous amount of data available for Yelp to use to determine if a restaurant will close before it happens. Yelp can use this information to open up new channels of revenue, specifically in the form of helping failing restaurants come back from the brinks of failure.

**Rationale Statement**

Restaurant owners do not know about the closure of their restaurant early enough to prevent it from happening. We have proposed a solution that can warn business owners of closure early enough so that they can take preventative measures. As stated before, our goal of this project is to look out for features that better predict the closure of a restaurant.

Our prototype will be able to predict this based on the customer’s star rating and finding which restaurant features have the most impact on restaurant’s average rating by performing sentiment analysis on user reviews. Our key metric for model evaluation will be precision and accuracy. Though accuracy will indicate how well our model is performing, we’ll be focusing on a high precision value for the “closed” class.

There will be 3 beneficiaries of our prototype. Yelp, our main beneficiary, will get to open a new revenue stream through consulting.  Secondly, clients/restaurants get to receive expert advice from Yelp in order to prevent closure and financial loss. Finally, consumers will receive a more fine-tuned and catered food experience.

**Problem**

**1.** When restaurants are already active on Yelp based on their Yelp profile we will be able to determine if they will close or not.

**2.** Using this model, Yelp can differentiate themselves from their competitors like Google Reviews, OpenTable, etc. Yelp would be providing a service that can help their clients make business decisions instead of just being a platform that consumers would use for making food decisions.

**3.** Existing restaurants will be able to increase their customer experience using our recommendation. By determining whether they will close or not would give clients an opportunity to improve their user experience and try and prevent their business from closing.

**Data Requirements**

We require our data set to have the following requirements: Business Name, City, Stars, Review Count, Is Open, Categories, Attributes, Hours of Operation, Review Date, Review Text, Review Star. Within these requirements we will constrain certain features like city to specifically 3-5 metropolitan areas. We also plan to segregate the ratings from 0-3 and then 4-5. ‘Is Open’, is the most important feature as it will tell us whether a restaurant is open or not as of January 19, 2019. We may have to make assumptions on the Categories feature, as some restaurants may be fusion but can still be labelled as Thai or Indian food.

Attributes will play a key role in our data as well, as there are many attributes that can affect the longevity of a business like whether it’s a take-out restaurant and if they have street or garage parking. We will have to sort through the many attributes and select the ones that are most important. The number of attributes may essentially lead our dataset into having 20-30 features. This would mean we would have to carefully filter the most important features and apply those to our model.

We are assuming Yelp’s reviews provided aren’t spam reviews and are actual reviews by individuals. It is an ongoing problem of reviews being created by bots, however we will make the assumption all the reviews are intact.

We also assume that the restaurants that are open or closed as of January 19, 2019 are still open or closed as of today.

**Data**

The dataset that will be used has been provided by Yelp. This set has been created for the Yelp Dataset Challenge which wants students to use Yelp’s data for academic or teaching purposes in innovative ways. The dataset contains 6,685,900 reviews, 192,609 businesses, 200,000 pictures within 10 metropolitan areas. It is a large json file that is located on their website at this link; <https://www.yelp.com/dataset/download>.

**Model / Architecture Approach:**

**Selections and Splitting**:

* Filter out various states of Canada.
* Filter out all business of category Restaurants in Toronto region
* Filtering out reviews for each restaurant in the dataset and combining all reviews into one single review for analysis.

**Cleaning and pre-processing**

* Categorize all restaurants by cuisine type using the matching keywords.
* Delete all records with more than 50% data missing.
* Feature extraction
* Dropping irrelevant features
* Label features containing values as True and False to binary values and converting them into numerical features.
* Getting dummies for categorical variables with multiple classes.
* Apply 'bag of words': the frequencies of various words appeared in each review as features and conduct SVM model to get score of each word.

**Model Selection**

        .     The 6 models we used to test were logistic regression, random forests,

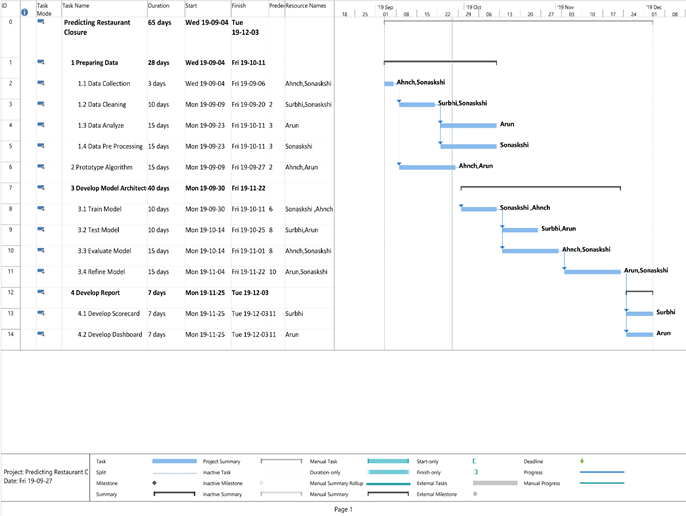
               support vector machine, K- nearest neighbor and decision tree.

       .     To select our models and classify our data, we used the scikit ­learn machine learning package for python. This package used for processing data using various models, outputting the training accuracy, test accuracy, test precision and test recall.

       .     We experimented with changing different hyperparameters for our models. Top 3 performing models were selected for further tweaking the model for better results.

**Project Work Breakdown**

The following is break down of the work required for this project. Each deadline will be met and the work for the task will be submitted via a cloud-based service. Any tasks done outside of the breakdown will be communicated amongst the group and added to the breakdown.

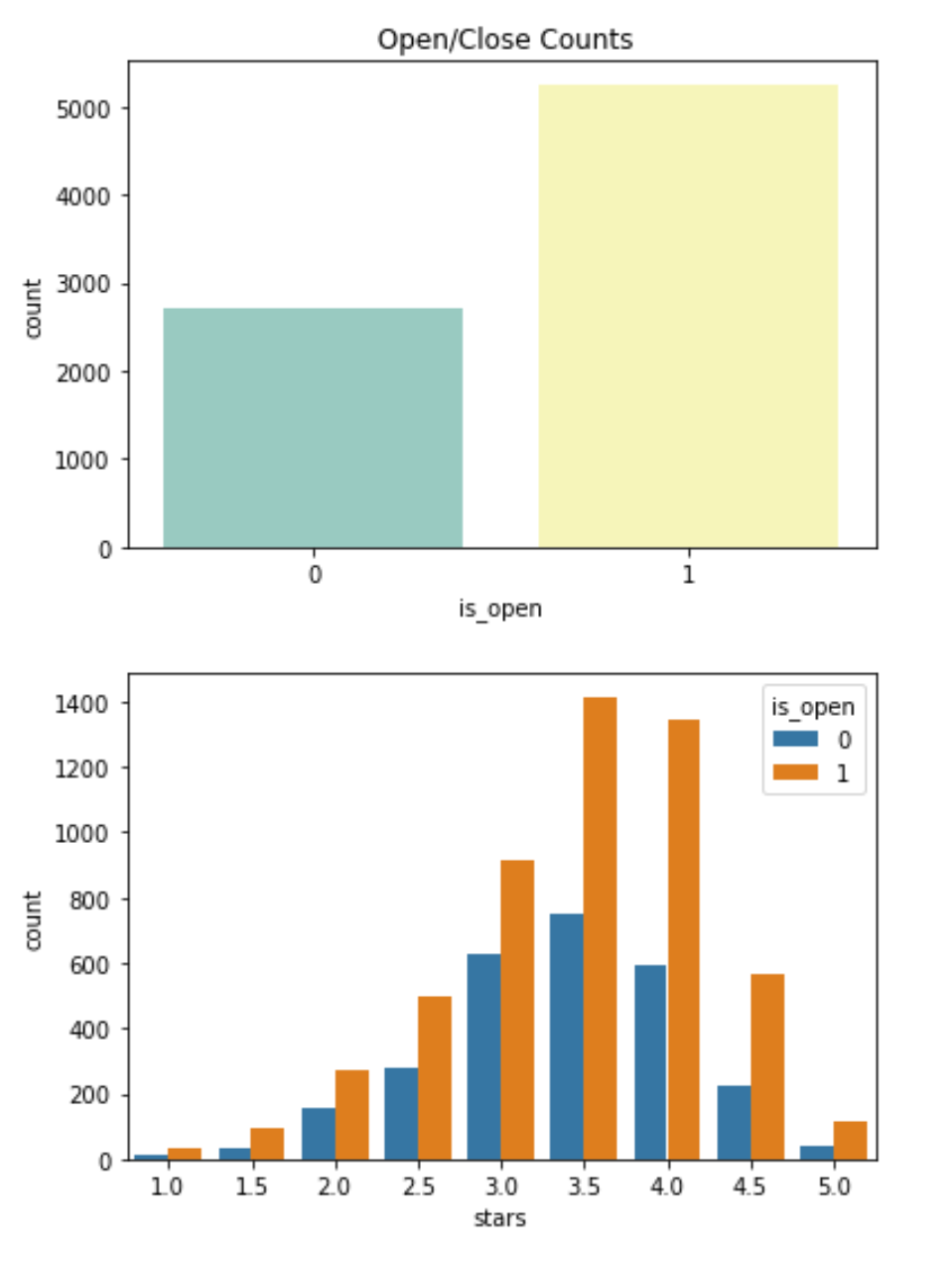


**Exploratory Data Analysis**

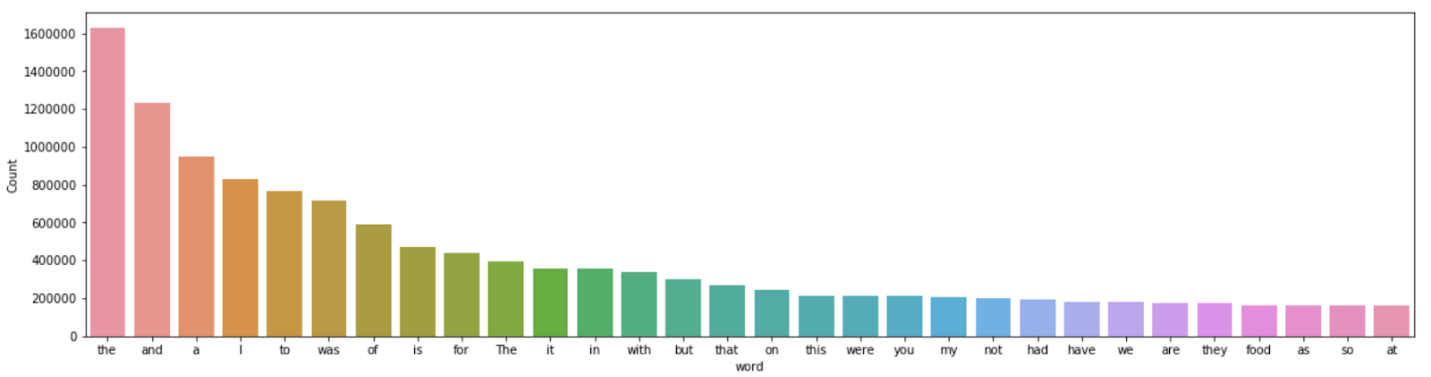
**Visualization**

**Visualization**

The two graphs below visualize the distributions of the restaurants that are open vs ones that are closed. The second graph depicts the distribution of “stars” that have been awarded for both open and closed restaurants.



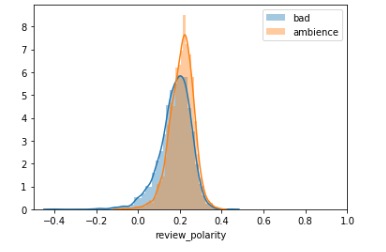
The next graph shows the count of certain words that appear in the reviews:



This word cloud represents the frequency of words within reviews:



The following graph shows the polarity of key words – bad and ambience:



**Data Pre-Processing:**

**Feature Engineering**

*Business Data*

In our previous version deliverable, we trained our model without any feature extractions and our models’ accuracy wasn't good enough.

So, a good place to collect some features was from business data, initially we included businesses that contained 'Restaurants' in categories. We further wanted to extract several cuisines and group restaurant based on that for our analysis. We extracted 17 types of cuisine.

Next, we had to tackle the attributes feature. This feature contained a list for each restaurant with various descriptions. Take for example the list from the second restaurant in the data frame. Since not every restaurant has all of the attributes listed above, hence we extracted all the features in the attribute’s column as separate features. Next, we dropped all the features which had more than 50% of the data missing.

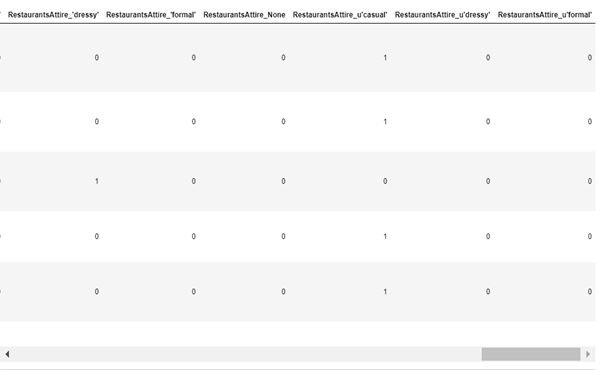
This way, we identified the ones that we thought will be valuable for including as a feature and check to see how many restaurants contain this complete subset.



We then further extracted Ambience and Business Parking. We also moved ahead with refactoring them as features by imputing 1 and 0 in place of True and False and converted those features into numerical features.

Encoding for categorical variables:

We used get\_dummies to separate each string in the caller series at the passed separator. If the string exists at that same index, then value is 1, otherwise 0 as seen in the image below:



Dropping Features that are no longer needed

 Some of these features are not going to be useful to us. In particular, we can removed:

·       address

·       business\_id

·       is\_open

·       latitude

·       longitude

·       neighborhood

·       postal\_code

·       type

·       hours

·       divey

·       Goodformeal

As shown above, the feature 'hours' contains 102464 non-null entries, meaning that 29% of the data is missing for this feature. Moving forward, it makes sense to remove this feature. The remaining features are missing much less data, and will remove observations that are missing data rather than remove attributes. Similarly, for Divey, it just has one value for all records in the dataset.

*Review Data:*

With the business id in business table to review table, we collected all the reviews of each restaurant and perform sentiment analysis to analyze and get polarity and sentiment score.

SMOTE:

Class imbalance arises due to the fact that model is trained predominantly on the label of majority class and very little on the minority class. One way to get around this is to use SMOTE. SMOTE stands for - Synthetic Minority Over-sampling Technique. the usual reason for oversampling is to correct for a bias in the original dataset.

Train & Test split:

The dataset was split into 80% training set and 20% test set.

Evaluation Metrics

 We chose to use accuracy, precision, recall as evaluation metrics for testing the performance of different machine learning models on our dataset. We will also use confusion matrix in conjunction with accuracy for the sake of interpretation. Nevertheless, we will focus on improving the accuracy of the minority class prediction.

**Model Selection/Evaluation**

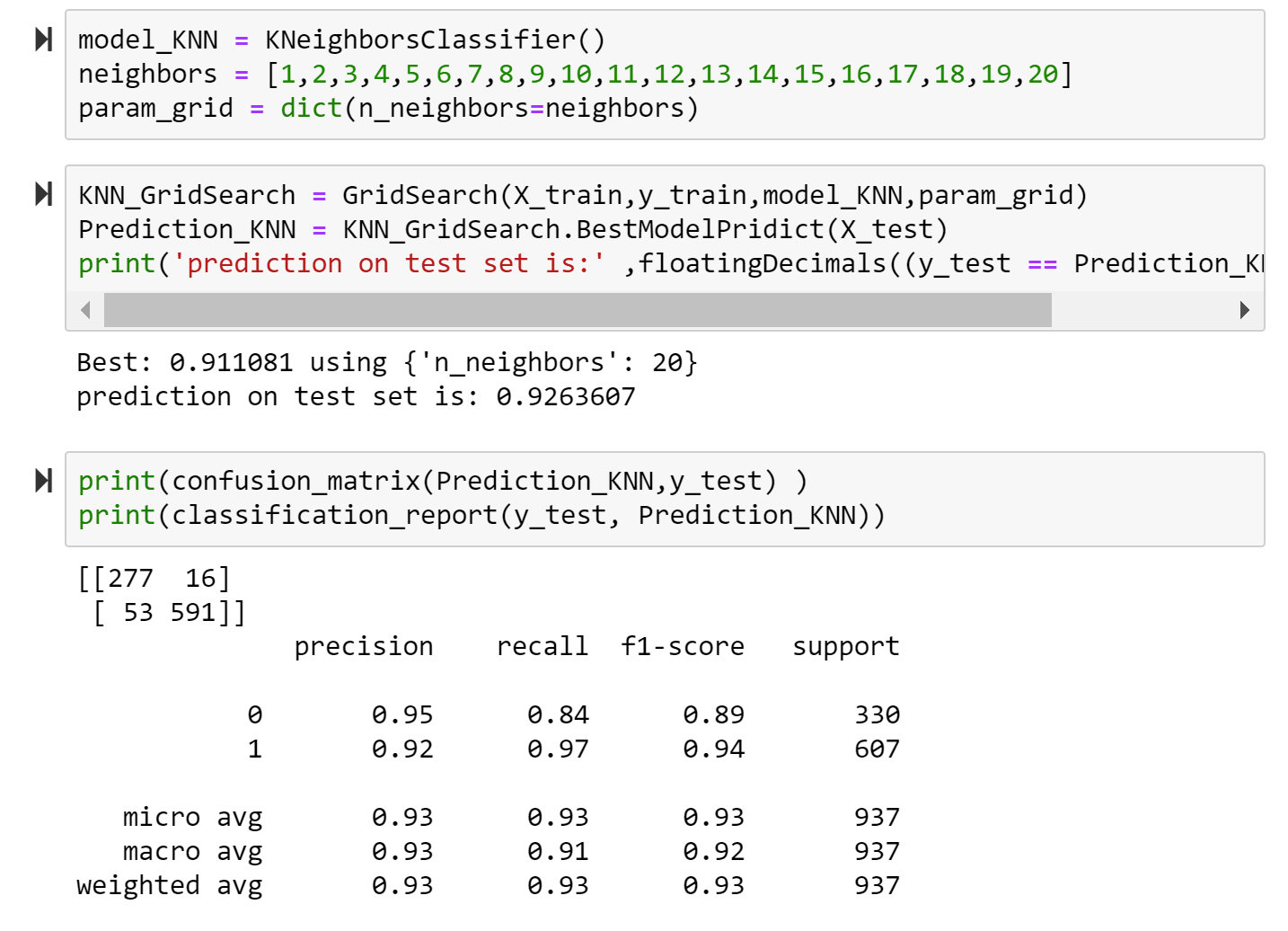
**Results**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Algorithm Name** | **Train Accuracy** | **Test Accuracy** | **Precision\_0** | **Recall\_0** | **Precision\_1** | **Recall\_1** |
| Logistic Regression | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 |
| K Nearest Neighbour | 0.911 | 0.926 | 0.95 | 0.84 | 0.92 | 0.97 |
| Classification and Regression Trees | 0.97 | 0.89 | 0.83 | 0.85 | 0.92 | 0.90 |
| Support Vector Machine | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 |
| Random Forest | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 |

**K-NN**

The K Nearest Neighbor algorithm is a robust classifier and is very easy to understand and implement. It is known as the “lazy” learning algorithm. It clusters the data into several classes in order to predict how a new point will be classified.

* Pros
  + Simple and Intuitive
  + Makes no assumptions
  + Constantly Evolving and widely used
  + Easy to implement
* Cons
  + Very slow algorithm
  + Curse of dimensionality (works well with a small number of input variables)
  + Very sensitive to outliers
  + Selection of K, or optimal number of neighbors when classifying new data



In the above snippet of code, we can see that our KNN model scored well, with an accuracy on the training set of 91%. It has also achieved an accuracy score of 92.6% on the test set. The results for determining whether a restaurant was closed was precise with a score of 95%. Recall for closed restaurants was 84%. For open restaurants the KNN model was 92% precise and achieved 97% for recall. Since our primary focus is precision, we can conclude that the KNN model returned more relevant results than irrelevant ones.

True Positives (TP): we correctly predicted that the restaurants will be open: 277

True Negatives (TN): we correctly predicted that the restaurants will be closed: 591

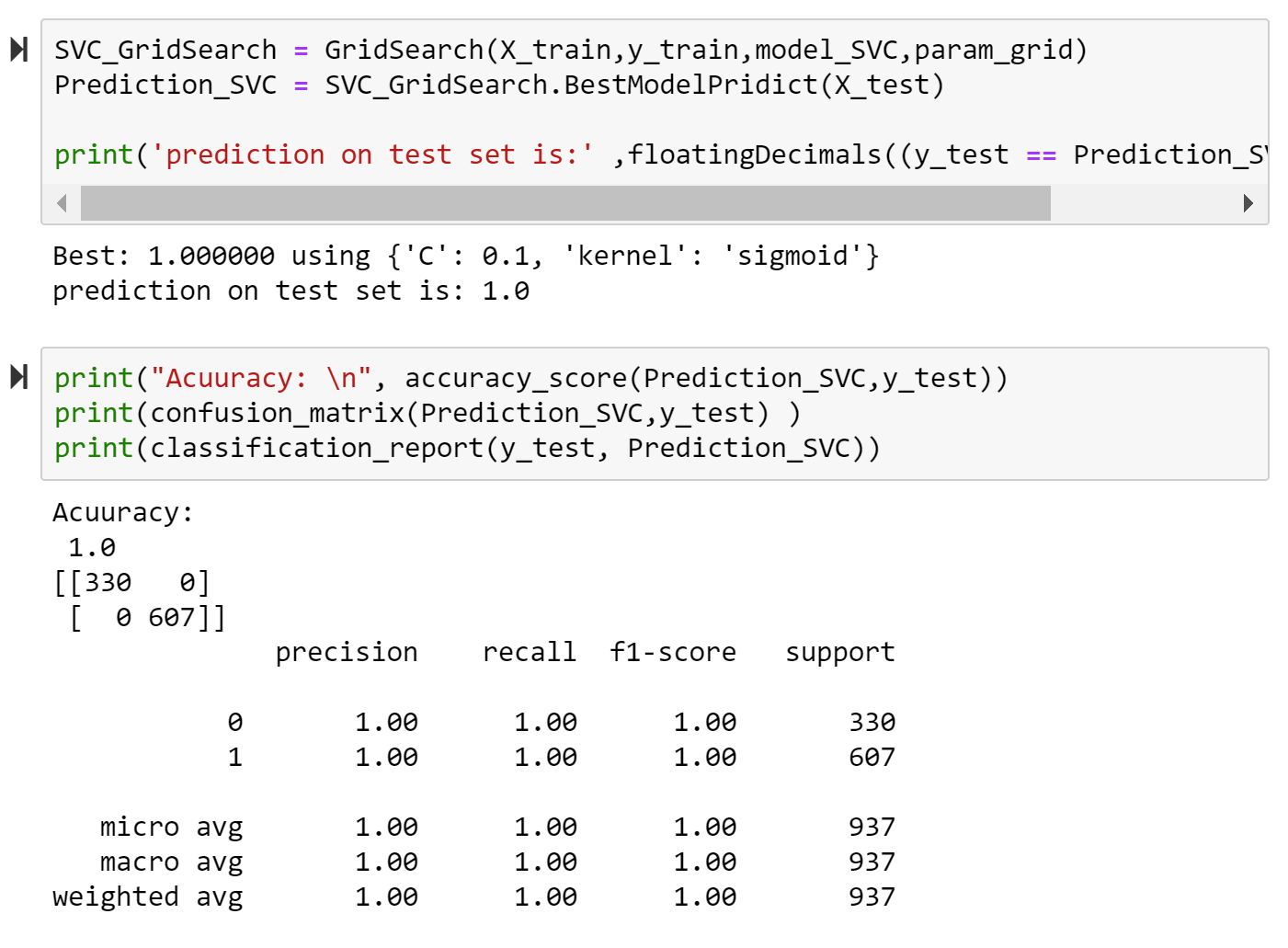
False Positives (FP): we incorrectly predicted that the restaurants will be open s (a "Type I error"):16

False Negatives (FN): we incorrectly predicted that the restaurants will be closed (a "Type II error"): 53

**Support Vector Machines**

Support vector machines (SVM) use kernels to calculate the greatest distance between two classes. The SVM algorithm then finds the boundary that ensures the greatest distance between those two classes. It acts similarly to linear and logistic regression, but has the ability to shift data that is not linearly separable to higher dimensions where it can be linearly separable by the use of a kernel trick. The kernel will then fit a line and shift the data back down to a 2D view. SVM also allows for the control of the margin between the support vectors and the hyperplane using the parameter C. This allows the model to be more flexible in terms of allowing misclassification, in order to maintain generalization. If C is higher, then there is less room for misclassification as precision is of utmost importance. Vice versa, if C is lower, then the model allows for some misclassification in order to create a better separation of data. The latter case is usually helpful for situations where the data of each class is extremely close to the hyperplane.

* Pros
  + Performs well with nonlinear boundary
  + Many kernels to choose from to ensure best output
  + Robust against overfitting
* Cons
  + Memory Intensive especially with large dataset
  + Choosing the correct kernel is a must to ensure highest accuracy



Above we can see that there is a huge overfitting problem with the SVM model, as it is returning a result of 100% for all accuracy, precision and recall sores. We will have to do further research and see if we can rectify the overfitting, allowing us to receive a better score in the future.

True Positives (TP): we correctly predicted that the restaurants will be open: 330

True Negatives (TN): we correctly predicted that the restaurants will be closed: 607

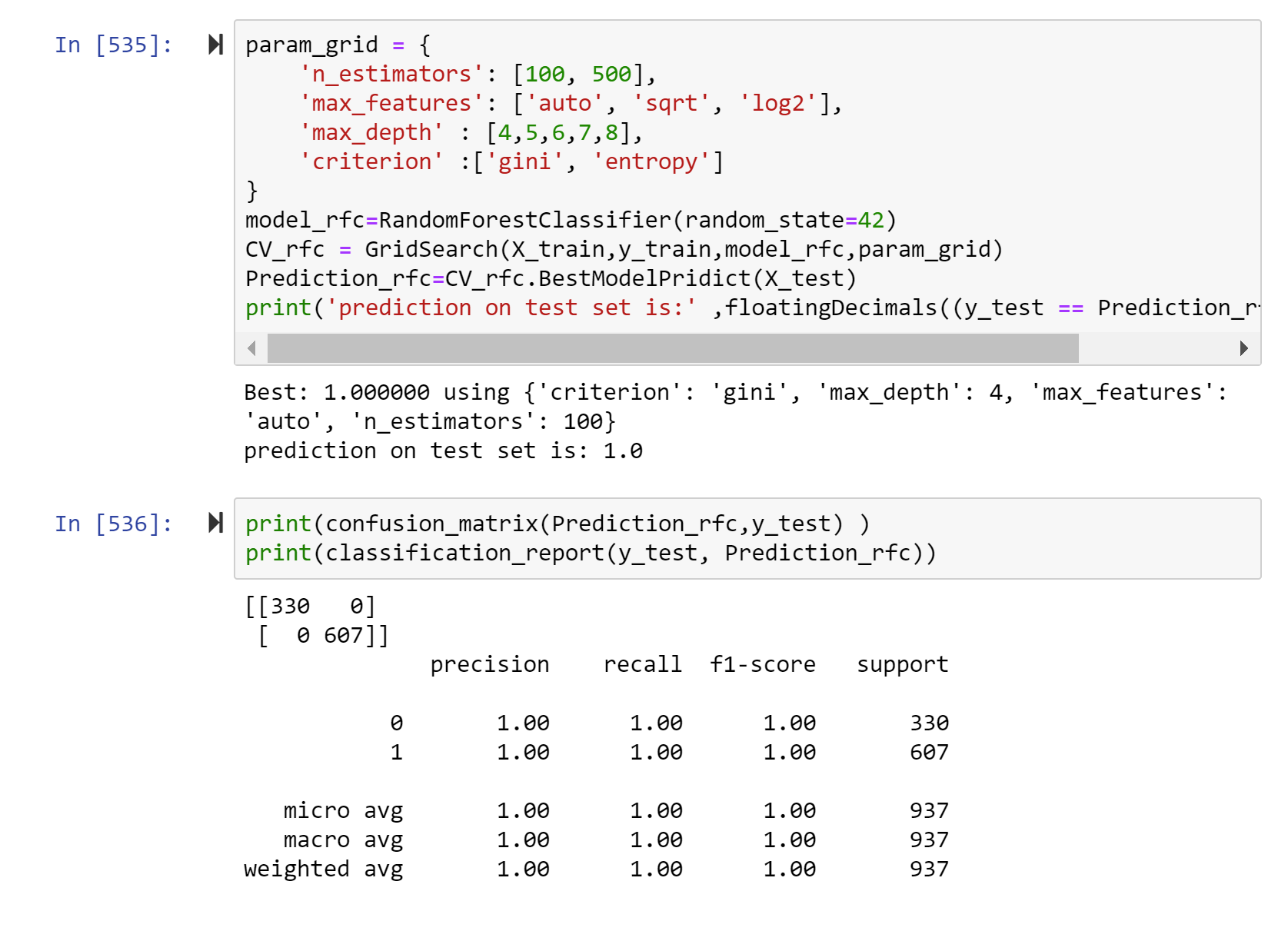
False Positives (FP): we incorrectly predicted that the restaurants will be open s (a "Type I error"):0

False Negatives (FN): we incorrectly predicted that the restaurants will be closed (a "Type II error"):0

**Random Forest**

This algorithm follows the same idea as decision trees, but instead of using one tree, there are multiple decision trees used to produce an outcome.

* Pros
  + Removes correlation between trees
  + Reduced variance compared to singular decision tree
* Cons
  + Harder to visualize compared to regular decision trees
  + May require heavy computation



The Random Forest model is also going through an overfitting problem, as can be seen above. The results for all metrics are given as 100%, the same as SVM. This means we will have to take a further look at the problem and find a solution for a more accurate result.

True Positives (TP): we correctly predicted that the restaurants will be open: 607

True Negatives (TN): we correctly predicted that the restaurants will be closed: 330

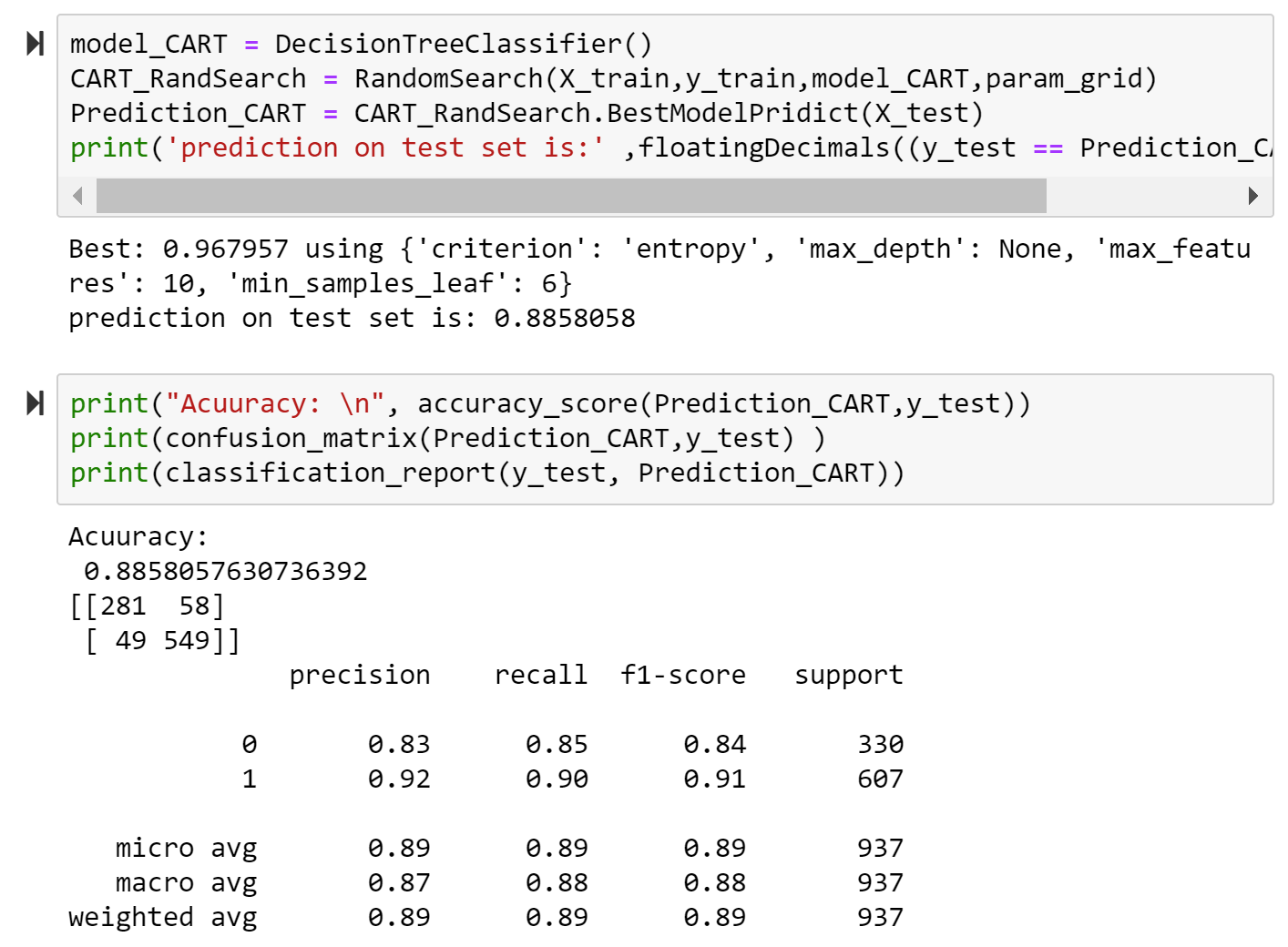
False Positives (FP): we incorrectly predicted that the restaurants will be open s (a "Type I error"):0

False Negatives (FN): we incorrectly predicted that the restaurants will be closed (a "Type II error"):0

**Decision Tree**

Decision trees is a supervised algorithm that consistently split based on the given parameters until there is an output. It consists of decision nodes and leaves.

* Pros
  + Easy to interpret and visually represent
  + Mimics human decision making
  + Can be used for regression or classification
  + Feature selection happens automatically
* Cons
  + Tends to overfit
  + Only axis aligned splits of data
  + Can be inaccurate, especially with large dataset



In the snippet above, we can see that the Decision Tree Classifier got a score of 96.8% on the training set, and 88.6% on the test set. It has also received a precision and recall score of 0.83 and 0.85 on closed restaurants respectively. For open restaurants, the result was 0.92 for precision and 0.90 for recall.

True Positives (TP): we correctly predicted that the restaurants will be open: 549

True Negatives (TN): we correctly predicted that the restaurants will be closed: 281

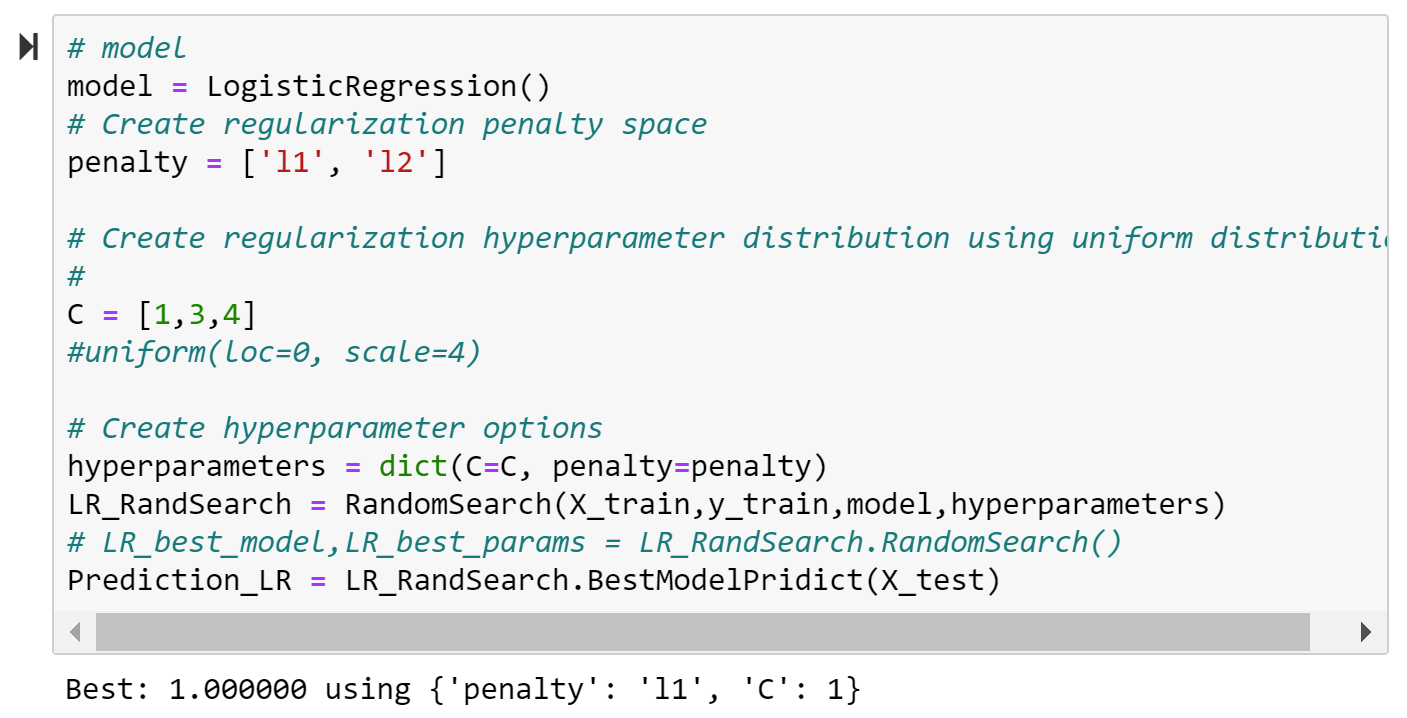
False Positives (FP): we incorrectly predicted that the restaurants will be open s (a "Type I error"):58

False Negatives (FN): we incorrectly predicted that the restaurants will be closed (a "Type II error"):49

**Logistic Regression**

Logistic regression is the classification counterpart of linear regression, used to make predictions between 0 and 1. In our case, we used this to predict whether a restaurant will close or not.

* Pros:
  + Can be regularized to avoid overfitting
  + Easy to implement
  + Outputs are easy to interpret
  + Adding new data will not cause problems
* Cons:
  + Hard to predict more complex problems
  + Will underperform when the decision boundaries are non-linear or are plentiful



Our Logistic Regression model also suffers from overfitting, as the result for the accuracy is 1.00.

**Candidate Algorithm Selection and Rationale**

Using the above information in regard to the algorithm evaluation, we can come to the conclusion that best algorithms to use were KNN followed by Decision Tree. Our target prediction is to determine whether a restaurant will close or not, hence we rely on the precision score. The more precise our predictions, the more confident we can be when providing our predictions to our clients. The weighted average of the precision metric for each were as follows:

**Weighted** **KNN Precision: 0.93**

**Weighted Decision Tree Precision: 0.89**

**Final Inference:**

We tested our model with the training dataset which will initially split during the test and train split. We ran the prediction on 10 restaurants out of which 5 were closed and 5 were open. Our classifier was able to predict correctly 4 restaurants as open and 3 as closed.

We’ll further run our test on a different province too see how our model performs.

**Conclusion**

With help of Natural Language Processing (NLP), we dealt with the text data, that helped in data mining or text mining (extracting important words from the review).

By visualisation of dataset, we analysed the imbalance in the data, which resolved by sampling.

Using Textblob package, sentiment orientation of reviews gives a sentiment polarity and sentiment subjectivity which helped us in labelling and training the model.

Hence, here in this project we were able to correctly analyse and visualise the data, extracted features and dealt with reviews and were able to train several classifiers out of which KNN gave us a great accuracy score of 93.00 %, which correctly predicts the if the restaurant will close or not.