

FACULTY OF TECHNOLOGY

SCHOOL OF COMPUTER SCIENCE

Restaurants total Sales predictions and Status in a period of time prediction using Machine Learning

**Student Name: Fung Yue (Lewis)**

**Project Supervisor: Dr. Simon Yiu, Dr. Ming Jiang and Dr. King-Hong Cheung**

# ABSTRACT

This research is intended to find the best machine learning model and use machine learning as the main method of performing new Restaurants opening predictions with different settings to provide some predictive insights for customers and is based on private data from Ambros Soft Ltd - Hong Kong Restaurants and public data from the Hong Kong Government Public Data Site (Ref. [Restaurant Receipts and Purchases](https://data.gov.hk/en-data/dataset/hk-censtatd-tablechart-qsr)). This research provided a literature review of related works and demonstrated a complete methodology and result assessment with visualization of results. The research models have also followed a varies machine learning methodology, eg. deep learning. To predict the location of Restaurants, Keras regression, and the Form of Restaurants, timeframes, payments, and employers were used. The model of deep learning with regression relies on features to predict overall restaurant sales. To predict whether the client can open restaurants or not, classification models will be considered and used. Additionally, in the case of overall sales, the algorithms k neighbor regression and random forest regression work best, while in the case of restaurant opening, the algorithms k neighbor regression and random forest regression work best. The results of the model show that using random forest regression to determine whether or not a customer would open a restaurant can be useful.

Keywords: Supervised Learning, Classification, Regression, Support Vector Regression (SVR), Restaurants Opening Prediction, Sales Prediction and Keras

Contents

[ABSTRACT 2](#_Toc69569686)

[**CHAPTER 1:** 7](#_Toc69569687)

[**INTRODUCTION** 7](#_Toc69569688)

[**1.1** **Project Overview** 7](#_Toc69569689)

[**1.2** **Client’s Overview** 8](#_Toc69569690)

[**1.3** **Client Requirements** 8](#_Toc69569691)

[**1.4** **Project Objectives** 8](#_Toc69569692)

[**1.5** **Constraints** 9](#_Toc69569693)

[**1.6** **Relevance to Program** 9](#_Toc69569694)

[**1.7** **Chapter Structure** 9](#_Toc69569695)

[**CHAPTER 2:** 11](#_Toc69569696)

[**LITRATURE REVIEW** 11](#_Toc69569697)

[**2.1** **Introduction** 11](#_Toc69569698)

[**2.2** **Restaurants Status and Sales Predictions** 11](#_Toc69569699)

[2.2.1 Restaurant Status Predictions 11](#_Toc69569700)

[2.2.2 Data source reference and linkage 11](#_Toc69569701)

[2.2.3 Feature set and target 12](#_Toc69569702)

[2.2.4 Method Algorithms 12](#_Toc69569703)

[**2.3** **Machine Learning** 12](#_Toc69569704)

[2.3.1 Why Use Machine Learning? 12](#_Toc69569705)

[2.3.2 Types of Machine Learning Algorithm 13](#_Toc69569706)

[2.3.3 Decision Trees 13](#_Toc69569707)

[2.3.4 Support Vector regression (SVR) 14](#_Toc69569708)

[2.3.5 Random Forest Regression: 16](#_Toc69569709)

[2.3.6 K-Nearest Neighbor Regression 17](#_Toc69569710)

[2.3.7 Artificial neural networks (ANN) - MLP Regression 20](#_Toc69569711)

[**2.4** **Chapter Summary** 21](#_Toc69569712)

[**CHAPTER 3:** 22](#_Toc69569713)

[**RESEARCH METHODOLOGY** 22](#_Toc69569714)

[**3.1** **Introduction** 22](#_Toc69569715)

[**3.2** **Data Science Analysis (EDA)** 22](#_Toc69569716)

[**3.2.1 Understand the Problem** 24](#_Toc69569717)

[**3.2.2 Understand the Data** 22](#_Toc69569718)

[**3.2.3 Select the Tool** 24](#_Toc69569719)

[**Libraries:** 25](#_Toc69569720)

[**Machine Learning Theory:** 26](#_Toc69569721)

[**3.2.4 Data Preparation** 26](#_Toc69569722)

[**3.2.5 Model Analyze** 27](#_Toc69569723)

[**3.2.6 Present the Results** 27](#_Toc69569724)

[**3.2.7 Deploy the Model** 27](#_Toc69569725)

[**3.3** **Methodology** 27](#_Toc69569726)

[3.3.1 Deep Sequential Neural Network 34](#_Toc69569727)

[**3.4** **Procedures and Data Definitions** 43](#_Toc69569728)

[**3.5** **Evaluation Methods** 43](#_Toc69569729)

[Accuracy Measures techniques used for Classification 43](#_Toc69569730)

[Accuracy Measures techniques used for Regression 44](#_Toc69569731)

[**3.6** **Chapter Summary** 44](#_Toc69569732)

[**CHAPTER 4:** 45](#_Toc69569733)

[**RESULTS AND EVALUATIONS** 45](#_Toc69569734)

[**4.1** **Results** 45](#_Toc69569735)

[Boxplot for distributions of each feature 46](#_Toc69569736)

[Classification Algorithms Results for Restaurant Status Prediction 63](#_Toc69569737)

[Regression Algorithms Results for Total Sales Prediction 72](#_Toc69569738)

[**4.2** **Discussions** 79](#_Toc69569739)

[**CHAPTER 5:** 80](#_Toc69569740)

[**CONCLUSIONS** 80](#_Toc69569741)

[**5.1** **Key Findings** 80](#_Toc69569742)

[**5.2** **Limitations** 80](#_Toc69569743)

[**5.3** **Recommendations and Further Studies** 80](#_Toc69569744)

[**5.4** **Social, Ethical, Professional and Legal Issues** 80](#_Toc69569745)

[**5.5** **Conclusions** 81](#_Toc69569746)

[**References** 82](#_Toc69569747)

[Figure 1: Importing Libraries 27](#_Toc59932243)

[Figure 2: Reading the dataset 28](#_Toc59932244)

[Figure 3: Checking Null Values in each column 28](#_Toc59932245)

[Figure 4: Viewing the overall status 28](#_Toc59932246)

[Figure 5: Code for Boxplot 29](#_Toc59932247)

[Figure 6: Co relation between variables 29](#_Toc59932248)

[Figure 7: Selection of X and Y for classification algorithm 30](#_Toc59932249)

[Figure 8: Train and test splitting the data 30](#_Toc59932250)

[Figure 9: Support vector machine for restaurant status Prediction 31](#_Toc59932251)

[Figure 10: Logistic Regression for restaurant status Prediction 31](#_Toc59932252)

[Figure 11: Decision tree Classifier for restaurant status prediction 32](#_Toc59932253)

[Figure 12: Random Forest Classifier for restaurant status prediction 32](#_Toc59932254)

[Figure 13: Ridge Classifier for restaurant status prediction 33](#_Toc59932255)

[Figure 14: SGD Classifier for restaurant status prediction 33](#_Toc59932256)

[Figure 15: K Neighbor Classifier for restaurant status prediction 34](#_Toc59932257)

[Figure 16: Multi-layer perception neural network for restaurant status prediction 34](#_Toc59932258)

[Figure 17: Deep Sequential neural network for restaurant status prediction 36](#_Toc59932259)

[Figure 18: Combined Plot for accuracy of each classifier 37](#_Toc59932260)

[Figure 19: K-fold cross validation on random forest 37](#_Toc59932261)

[Figure 20: Selection of Variables for Regression Algorithms 38](#_Toc59932262)

[Figure 21: Train Test split on regression algorithms 38](#_Toc59932263)

[Figure 22: Random forest regression for total sales prediction 39](#_Toc59932264)

[Figure 23: K Neighbor Regression for total sales Prediction 39](#_Toc59932265)

[Figure 24: Decision Tree Regression for Total sales prediction 40](#_Toc59932266)

[Figure 25: Gradient Boosting Regression for Total Sales Predictions 40](#_Toc59932267)

[Figure 26: MLP Regression for total sales prediction 41](#_Toc59932268)

[Figure 27: Neural Networks for total sales predictions 41](#_Toc59932269)

[Figure 28: Code for combined plot of origional and predicted total sales for all algorithms 42](#_Toc59932270)

[Figure 29: Code for combined mean square error plot for regression algorithms 42](#_Toc59932271)

[Figure 30: Cross Validation for K Neighbor Regression for total sales prediction 43](#_Toc59932272)

[Figure 31: Null Values of each columns 45](#_Toc59932273)

[Figure 32: Count of Restaurant Status 46](#_Toc59932274)

[Figure 33 Boxplot visualization for restaurant type 46](#_Toc59932275)

[Figure 34: Boxplot visualization for restaurant area 47](#_Toc59932276)

[Figure 35: Boxplot visualization for Districts 47](#_Toc59932277)

[Figure 36: Boxplot of Number of people able to work from age 15 to 19 base on District 48](#_Toc59932278)

[Figure 37: Boxplot of Number of people able to work from age 20 to 24 base on District 48](#_Toc59932279)

[Figure 38: Boxplot of Number of people able to work from age 25 to 29 base on District 49](#_Toc59932280)

[Figure 39: Boxplot of Number of people able to work from age 30 to 34 base on District 49](#_Toc59932281)

[Figure 40: Boxplot of Number of people able to work from age 35 to 39 base on District 50](#_Toc59932282)

[Figure 41: Boxplot of Number of people able to work from age 40 to 44 base on District 50](#_Toc59932283)

[Figure 42: Boxplot of Number of people able to work from age 45 to 49 base on District 51](#_Toc59932284)

[Figure 43: Boxplot of Number of people able to work from age 50 to 54 base on District 51](#_Toc59932285)

[Figure 44: Boxplot of Number of people able to work from age 55 to 59 base on District 52](#_Toc59932286)

[Figure 45: Boxplot of Number of people able to work from age 60 to 64 base on District 52](#_Toc59932287)

[Figure 46: Boxplot of Number of people able to work from age 65 and above base on District 53](#_Toc59932288)

[Figure 47: Boxplot of Number of people able to work from age 15 to 19 base on Area 53](#_Toc59932289)

[Figure 48: Boxplot of Number of people able to work from age 20 to 24 base on Area 54](#_Toc59932290)

[Figure 49:Boxplot of Number of people able to work from age 25 to 29 base on Area 54](#_Toc59932291)

[Figure 50: Boxplot of Number of people able to work from age 30 to 34 base on Area 55](#_Toc59932292)

[Figure 51: Boxplot of Number of people able to work from age 35 to 39 base on Area 55](#_Toc59932293)

[Figure 52: Boxplot of Number of people able to work from age 40 to 44 base on Area 56](#_Toc59932294)

[Figure 53: Boxplot of Number of people able to work from age 45 to 49 base on Area 56](#_Toc59932295)

[Figure 54: Boxplot of Number of people able to work from age 50 to 54 base on Area 57](#_Toc59932296)

[Figure 55: Boxplot of Number of people able to work from age 55 to 59 base on Area 57](#_Toc59932297)

[Figure 56: Boxplot of Number of people able to work from age 60 to 64 base on Area 58](#_Toc59932298)

[Figure 57: Boxplot of Number of people able to work from age 65 and above base on Area 58](#_Toc59932299)

[Figure 58: Boxplot of total sales 59](#_Toc59932300)

[Figure 59: Boxplot of Number of transactions 59](#_Toc59932301)

[Figure 60: Boxplot of sqf 60](#_Toc59932302)

[Figure 61: Boxplot of monthly rent 60](#_Toc59932303)

[Figure 62: Boxplot of restaurant types 61](#_Toc59932304)

[Figure 63: Boxplot of location types 61](#_Toc59932305)

[Figure 64: Co relation plot for all features 62](#_Toc59932306)

[Figure 65: Results of Support Vector Machine Classifier for Restaurant Status Predictions 63](#_Toc59932307)

[Figure 66: Results of Logistics Regression for Restaurant Status Predictions 64](#_Toc59932308)

[Figure 67: Result of decision tree Classifier for restaurant status prediction 65](#_Toc59932309)

[Figure 68: Results of random forest for restaurant status prediction 66](#_Toc59932310)

[Figure 69: Results of ridge classifier for restaurant status Prediction 67](#_Toc59932311)

[Figure 70: Results of SGD Classifier for restaurant status prediction 68](#_Toc59932312)

[Figure 71: Results of K Neighbor classifier for restaurant status predictions 69](#_Toc59932313)

[Figure 72: Results MLP classifier for restaurant status predictions 69](#_Toc59932314)

[Figure 73: Restaurants Recommendations accuracy comparisons 70](#_Toc59932315)

[Figure 74: K fold validation step-1 results 71](#_Toc59932316)

[Figure 75: K Fold validation step 2 results 71](#_Toc59932317)

[Figure 76: K Fold Validating step 3 results 72](#_Toc59932318)

[Figure 77: Random Forest Regression total sales predictions original and Predicted results for test data points 73](#_Toc59932319)

[Figure 78: K Neighbor Regression total sales predictions original and Predicted results for test data points 73](#_Toc59932320)

[Figure 79: Decision Tree Regression for total sales predictions original and Predicted results for test data points 74](#_Toc59932321)

[Figure 80: Gradient Boosting Regression for total sales predictions original and Predicted results for test data points 74](#_Toc59932322)

[Figure 81: MLP Regression for total sales predictions original and Predicted results for test data points 75](#_Toc59932323)

[Figure 82: Neural Network Regression for total sales predictions original and Predicted results for test data points 75](#_Toc59932324)

[Figure 83: Combined plot for regression results 76](#_Toc59932325)

[Figure 84: Mean Squared Algorithm for each algorithm to predict sales 76](#_Toc59932326)

[Figure 85: K Fold Validation Split 1 results 77](#_Toc59932327)

[Figure 86: K Fold Validation Split 2 results 77](#_Toc59932328)

[Figure 87: K Fold Validation split 3 results 78](#_Toc59932329)

[Figure 88: K Fold Validation split 4 results 78](#_Toc59932330)

[Figure 89: K Fold Validation split 5 results 79](#_Toc59932331)

# **CHAPTER 1:**

# **INTRODUCTION**

## **Project Overview**

The Hong Kong restaurant industry is mature and saturated, and growth in this industry is primarily driven by the overall growth of the local economy. Revenue for the restaurant sector fell 5.9 per cent to HK$112.5 billion (US$14.5 billion) in 2019 from the previous year, according to provisional data from the Census and Statistics Department released on https://www.censtatd.gov.hk/en/

In Hong Kong, there are various locations where restaurant’s owner could be considered to start a new business. However, when people want to open a new restaurant but don't know anything about the area, district, or other variables, they'll have a harder time planning and budgeting for it than open a restaurant based on data and evidence (Patel et al., 2011), To prepare this study and find the best approach for opening a restaurant in Hong Kong, we will leverage restaurant point of sales data given by Ambros Soft Ltd(Ambros), public data from Hong Kong government website and property agency website.

Identifying and investigating parameters and variables that influence restaurant’s revenue and able to predict whether the business can be sustainable for at least a period of year are the most difficult aspects of forecasting. Those variables corelated from location, revenue, age group, etc.. all play a role of analyzing factors. (KIM, H. & GU, Z. 2006) used logistic regression analysis for predicting bankruptcy in the restaurant industry and discovered that this information influences the market at a certain extent (Kim and Upnejy, 2014). In this area, there are numerous academic and commercial publications available, but only a portion of them is accessible to the public, such as F. M. T. Hossain, M. I. Hossain and S. Nawshin, used supervised machine learning (ML) techniques analysis to predict proliferation of location prediction; (LIAN, J., ZHANG, F., XIE, X. & SUN, G.) provided statistical formula to calculate restaurant survival indicator.

In this project, the aim of the analysis is to find an answer for two questions.

The first question is that if we can predict whether or not after a period of time the restaurant on their location is still open (survive). The ML classification algorithms will be used to answer the classification question.

Secondly, we will predict a restaurant revenue based on its private data from Ambros Point of Sales(POS) with real estate and government public data. This is a regression problem, and ML regression algorithms are used to solve this question.

## **Client’s Overview**

Ambros is a company that specializes in providing food and beverage (F&B) solutions. Ambros will continuing to develop new services and cloud-based products in order to maintain their rapid growth of business. In this project, Ambros assigned me the task of finding the best ML algorithms for their new product/services planning offer in coming years. Every restaurant owner in the technological era wanted to use big data to boost revenue. This project's target client is to address the need of all restaurant business owner who wants to open a new restaurant in selected or planned location in Hong Kong city and wants to know whether it will be able to sustain the growth of the busines in a period of time with supported by revenue forecast from this project.

## **Client Requirements**

Since Ambros' s clients are restaurant owners, they aren't interested in math or ML, but they do want the best end result of forecasting restaurant sustainability according to different variables after a period of time and revenue for their restaurant. Their client wanted to know if they could open a restaurant in Hong Kong based on various data. These restaurant owners are also curious about the sales of other restaurants based on various variables so that they can estimate their own revenue.

* Use the public data of the Hong Kong restaurant, government population and real estate as the main dataset.
* Investigate and adopt ML approaches to prepare this research.
* Design and develop an indication model for the best location and revenue forecast.
* Provide comprehensive documentation, report and model results, including limitations and future suggestions.

## **Project Objectives**

The goal of this research is to use ML models as methods for predicting sustainability by Ambros data named as “restaurant status” and revenue in a variety of settings, including features sections and K Fold Cross Validations. It is necessary to provide restaurant owners with information.

The objectives of this research are listed below.

* To conduct literature review on restaurant status and revenue prediction techniques and cutting-edge methodologies and frameworks.
* To investigate the state of the art in data application design and development with ML models. (K Neighbors, Random Forest, Neural Networks etc.).
* To investigate restaurant statistical based on features like number of people available in districts and area with age group.
* To develop K Fold Cross validation framework to predict the sales and status of restaurant.
* To develop Neural Networks to achieve the predictions outcome
* To visualize original and predicted results.
* To obtain client’s evaluation on delivered product.

## **Constraints**

For this research few conditions would be constraints

* Limited facilities, Machine with high computing power and large memory.
* Limited literature review available.
* Limited study materials and workshops provided by the client.
* Limitation on the accuracy and features of the data due to sensitive information of different restaurants which we only have around 5000 restaurant data provided by Ambros.
* Limited time to reach a certain level of accuracy for train and test.
* Missing some variable factors which could be useful for this research

1. Restaurant operational efficiency in each restaurant.
2. Chef capability and “Star Chef” effect.
3. Seasonality of food ingredients from varies suppliers.
4. Different ingredient supplier pricing strategy to restaurant owner.
5. Online comment of the restaurant.
6. Logistic and cost control from Central Kitchen Management chain restaurant.
7. Non-financial growth measures include growth of restaurant employment, customer satisfaction and loyalty program (Brown and Mitchell, 1993).

## **Relevance to Program**

This MSc Data Science final project focuses mainly on research, design and development of effective solutions to real-life data problems in the hospitality – restaurant sector. The aim of this research was to solve the problem of restaurant sustainability and revenue prediction by using supervised learning. In this course, supervised learning as a subset of ML is part of the syllabus, CETM 50 and ML and Data Analytics (CETM 24) introduce the concept of AI and ML and the applicability of algorithms. CETM24 explained the Data Science Foundation and CETM46 explained the design, development and best practice of data visualization for data products. Documentation on the reports, research methodology and literature review studies covered by CETM 47 is also adopted.

## **Chapter Structure**

**Chapter 1:** This chapter is a brief introduction to this study and the motivation of the problem/question statements. It also covers client’s information, requirements, project objectives and the constraints.

**Chapter 2:** Literature review is conducted which involved the restaurant predictions, operations and methods of ML regression and classification.

**Chapter 3:** This chapter introduces comprehensive view on the whole ML models, the methodology, assumptions, procedures, and evaluation methods.

**Chapter 4:** This chapter discusses about the results and evaluations on the ML models.

**Chapter 5:** This chapter summarizes the main key findings on ML models. Also, it provides recommendations, limitations and further research directions of this research. Also, the social ethical professional and legal issues will be discussed.

# **CHAPTER 2:**

# **LITRATURE REVIEW**

## **Introduction**

This chapter will present a research to this study and then examine the theories that underpin them in order to determine if the framework in this research is convincing and has adequate academic support.

## **Restaurants Status and Sales Predictions**

Our topic is restaurant opening suggestion using the historical data. In literature review currently, this exact topic has not been studied. But some researches on restaurant data is done and revenue forecasting is also presented in relevance research accordingly. Based on our literature review on the factors influencing of restaurants, we collect information from two different types of data: (1) not publicly available data provided by Ambros’s restaurant data, and (2) publicly available data from Hong Kong Government and real estate agency website.

I firstly elaborate the data provided by Ambros, which contain information about restaurants and mainly serve as a ground-truth, i.e. labelled data for supervised ML. Next, I describe various web data sources collected to derive the growth indicating factors, which serve as input features to train multiple prediction models. I had also used Ambros’s restaurant data to predict the sales revenue which reference from (Rickard Adolfsson, Eric Andersson, George Osipov, 2019) but here in our case we have individual restaurant, so forecasting algorithms may not be fully applied here. The best thing to do for both cases is to apply supervised ML algorithms (LASEK, A., CERCONE, N. & SAUNDERS, J. 2016).

## 2.2.1 Restaurant Status Predictions

The restaurant survival indicator (LIAN, J., ZHANG, F., XIE, X. & SUN, G.) is the most important factor to identify our Restaurant Status prediction statement. The journal provides similar problem statement from my planning project, eg, the restaurants which belong to normal shops or closed shops categories for study. It leads to ask, can the restaurant survival prediction be solved using ML. They examine the performance of logistic regression(LR), gradient boosted decision tree and supported vector machine (SVM). This paper also discusses the problem of restaurant survival prediction by modelling four perspectives: geographical metrics, user mobility, rating scores, and review text. Accordingly, in our dataset from Ambros, we also include geographical data and related categories.

### 2.2.2 Data source reference and linkage

From <http://www.yelp.com> (Yelp) data reference from (Mr. N. Varatharajan1 , J. Guruprasad , K. Mathumitha ), the journal proposed propose a ML algorithms to resolve the issues of personalized restaurant selection relying upon Yelp data. It contains 10,000 No of restaurants with restaurant location, menu item and ratings. From our data source – Ambros provide more than 5000 individual restaurant data similar to Yelp. Data quality management is a crucial challenge in data management aiming at an improved usability and reliability of the data. Entity identification is defined as the detection and merging of two or more records representing the same real-world identity across multiple data sets, which is relevant in duplicate detection and elimination as well as data integration. Apart from data cleaning, data integration and data warehousing, entity identification is closely related to information retrieval, pattern recognition and data mining as well, thus, making use of ideas from several research areas (e.g. Bilenko et al., 2003).

### 2.2.3 Feature set and target

The first set of variables includes detailed information about the restaurant's start date, year, payment type, location type, restaurant type, and revenue. The second dataset from the government and real estate agency includes district worker age ranges ranging from 20 to 65 plus, monthly rent, and restaurant size. These variables are related to each other (Stephan Gogolev, Evgeniy M. 2020). The next step is to retrieve the target and features from the dataset. When it comes to the target variable, it's important to keep in mind that smaller sample sizes can lead to inaccurate results. The need for a minimum number of subjects underlines the importance of data sharing initiatives such as Openneuro.org (https://openneuro.org/) and schizconnect (http://schizconnect.org)

### 2.2.4 Method Algorithms

Restaurant growth is a highly complex mechanism, thus predicting the growth of restaurants requires ML algorithms which are capable to handle a high level of complexity. Therefore, we use varies ML models, which are able to model complex interactions between the input variables and thus, share a predominant role in a range of research domains (Cutler et al. 2007). Furthermore, logistic regression (LR) is chosen as a benchmark due to its wide use for economic modelling in the past (Youn and Gu, 2010). Reference from the journal (ML based class level prediction of restaurant reviews). They used four algorithms to build a model using a training dataset. These are Multinomial Naïve Bayes (MNB), Support Vector Machine (SVM) K Nearest Neighbor (KNN) and Logistic Regression (LR). They used k-fold cross validation to measure the performance of our built-in model, which is a very effective way to find the accuracy of the predictive model. It divides the data set into the training and testing set and calculates the performance and used 10 times in their experiment, which means k=10. Logistic Regression (LR) has produced the best result with more than 77% accuracy. The ML can be considered in our experiment as well.

## **Machine Learning**

## 2.3.1 Why Use Machine Learning?

1. First, I'd look at how data is typically presented. In my case, I may notice that certain figures or numbers appear frequently in the subject. Perhaps I'd notice a few other patterns in the restaurant's name, the type of location, and so on.
2. I would write a detect algorithm for each of the patterns I noticed, and my program would flag as correlation of independent or dependent variable if several of these patterns were detected.
3. I'll test my program and repeat steps 1 and 2 until it's good enough.

Since the problem is not trivial, my program will most likely turn into a long list of complex rules that will be difficult to maintain. In practice, a ML based technique automatically learns which dependent and independent variables are good predictors of data changes and insight. The program will be much shorter, easier to maintain, and probably more accurate.

Appling ML is great for :

* Problem for which existing solution require a lot of hand-turning or long list of rules, ML algorithm can often simplify code and perform better.
* Complex problems for which there is no good solution at all: the ML techniques can find a solution.
* Fluctuating environments: a ML system can adapt to new data.
* Getting insights about complex problems and large amount of data.

The growing popularity of ML has led to the development of a number of tools designed to make the data or application of this approach accessible to the novice learning machine. These efforts have resulted in tools such as PRoNTo (Schrouff et al., 2013) and NeuroMiner (https://pronia.eu/neurominer/) that do not require programming skills. However, learning how to program a ML pipeline (even if it's a simple one) is an excellent way to gain insight into the strengths of this analytical approach as well as the potential "distortions" that may occur along the ML pipeline (Tandon & Tandon, 2018). It also allows greater flexibility, such as the use of any ML algorithm or data modality of interest.

In this chapter, we select different ML Models, and the discussion is taken from the official site of sklearn. (Jérémie du Boisberranger, 2007-2020)

## 2.3.2 Types of Machine Learning Algorithm

## Since there are so many different types of ML algorithms, it's helpful to categorize them into large categories. For this research, I chose several ML algorithms that I learned from various MSc Data Science courses and referenced from other journals.

## 2.3.3 Decision Trees

Decision Trees(DT) is a non-parametric regulated learning strategy utilized for characterization and relapse which we will using it in this research.

It can perform both classification and regression tasks, and even multioutput tasks. DT are also the fundamental components of Random Forests.

Formula:

Entropy at a give node t:

Entropy(t) = - (j|t)log p(j|t)

NOTE: p(j|t) is the relative frequency of class j at note t.

(CETM24-decisiontrees.pdf)

The concept of entropy originated in thermodynamics as a measure of molecular disorder: entropy approaches zero when molecules are still and well ordered. It later spread to a wide variety of domains, including Shannon’s information theory. I will use Scikit-Learn for the Classification And Regression Tree (CART) algorithm to train Decision Trees (also called “growing” trees). The idea is really quite simple: the algorithm first splits the training set in two subsets using a single feature and a threshold.

Decision Trees make very few assumptions about the training data. If left unconstrained, the tree structure will adapt itself to the training data, fitting it very closely, and most likely overfitting it. Such a model is often called a nonparametric model, not because it does not have any parameters (it often has a lot) but because the number of parameters is not determined prior to training, so the model structure is free to stick closely to the data. In contrast, a parametric model such as a linear model has a predetermined number of parameters, so its degree of freedom is limited, reducing the risk of overfitting.

## 2.3.4 Support Vector regression (SVR)

Support vector machines (SVMs) is a very powerful and versatile ML model, capable of performing linear or nonlinear classification, regression, and even outlier detection. It is one of the most popular models in ML. SVMs are particularly well suited for classification of complex but small- or medium-sized datasets.

SVMs algorithm can classify both linear and non-linear data. It first maps each data item into an n-dimensional feature space where *n* is the number of features. It then identifies the hyperplane that separates the data items into two classes while maximizing the marginal distance for both classes and minimizing the classification errors. The marginal distance for a class is the distance between the decision hyperplane and its nearest instance which is a member of that class. More formally, each data point is plotted first as a point in an n-dimension space (where *n* is the number of features) with the value of each feature being the value of a specific coordinate. To perform the classification, we then need to find the hyperplane that differentiates the two classes by the maximum margin. Figure below provides a simplified illustration of an SVM classifier.

Diagram

Description automatically generated

The distance from either set between the hyperplane and the nearest data point is known as the margin. This hyperplane separates the data better because it is as far from these support vectors as possible, which is another way of saying that we maximized the margin. The goal is to choose a hyperplane with the greatest possible margin within the training set between the hyperplane and any point, giving a better chance of correctly classifying new data. SVMs find the hyperplane h for separable training sets, which distinguishes the positive and negative training instances with the highest margin. Help Vectors are considered the instances nearest to the hyperplane. A weighted sum of support vectors is defined as the separating hyperplane. For our case in which the training data is linearly separable, we consider the ideal separating hyperplane. Then we generalize the principle of optimal hyperplane separation to our non-separable data case.

Slack variables ξi , *i = 1,2,...* are used to define how much a point fails to have a margin of ***g*** from the hyperplane:

ξi *=max(0, g - yi* (<**w,xi>**+ *b*) **)**

Find ***w*** and *b* such that  
**Φ**(**w**) **= *<w,w>/2*** + *C* Σ*ξi* is minimized and for all {(***xi*** *,yi*)} *yi (*<**w,xi>**+ *b*) **≥** 1- *ξi* and *ξi* **≥** 0, *i=1,2,...,n*

Parameter ***C*** can be viewed as a way to control overfitting. ***C > 0*** determines the trade-off between margin maximisation and training error minimisation

The benefits of support vector machines are:

Viable in high dimensional spaces.

Still viable in situations where number of measurements is more noteworthy than the quantity of tests.

Utilizations a subset of preparing focuses in the Decision capacity (called uphold vectors), so it is likewise memory effective.

Adaptable: diverse Kernel capacities can be indicated for the Decision capacity. Normal portions are given; however, it is additionally conceivable to indicate custom bits.

The detriments of support vector machines include:

In the event that the quantity of highlights is a lot more noteworthy than the quantity of tests, abstain from over-fitting in picking Kernel capacities and regularization term is urgent.

SVMs don't straightforwardly give likelihood appraises, these are determined utilizing a costly five-overlap cross-approval (see Scores and probabilities, underneath).

The support vector machines in scikit-learn uphold both thick (numpy.ndarray and convertible to that by numpy.asarray) and scanty (any scipy.sparse) test vectors as info. In any Position, to utilize a SVM to make expectations for scanty information, it probably been fit on such information. For ideal execution, use C-requested numpy.ndarray (thick) or scipy.sparse.csr\_matrix (inadequate) with dtype=float64.

The strategy for Support Vector Classification can be stretched out to take care of regression issues. This technique is called Support Vector Regression.

The model delivered by support vector characterization (as portrayed above) relies just upon a subset of the preparation information, on the grounds that the cost work for building the model couldn't care less about preparing focuses that lie past the edge. Comparably, the model delivered by Support Vector Regression relies just upon a subset of the preparation information, in light of the fact that the cost work disregards tests whose forecast is near their objective.

There are three unique executions of Support Vector Regression: SVR, NuSVR and LinearSVR. LinearSVR gives a quicker execution than SVR however just thinks about the straight part, while NuSVR actualizes a somewhat unexpected plan in comparison to SVR and LinearSVR. See Implementation subtleties for additional subtleties.

Likewise, with arrangement classes, the fit technique will take as contention vectors X, y, just that for this situation y is relied upon to have drifting point esteems rather than whole number qualities.

### 2.3.5 Random Forest Regression:

Random Forest is an ensemble of Decision Trees, generally train a group of Decision Tree classifiers, each on a different random subset of the training set. To make predictions, we just obtain the predictions of all individual trees, then predict the class that gets the most votes. Such an ensemble of Decision Trees is called a Random Forest, and despite its simplicity, this is one of the most common ML algorithms available today. The Random Forest algorithm introduces extra randomness when growing trees; instead of searching for the very best feature when splitting a node, it searches for the best feature among a random subset of features. This results in a greater tree diversity, which trades a higher bias for a lower variance, generally yielding an overall better model. Feature importance: Yet another great quality of Random Forests is that they make it easy to measure the relative importance of each feature. The tools Scikit-Learn measures a feature’s importance by looking at how much the tree nodes that use that feature reduce impurity on average (across all trees in the forest). More precisely, it is a weighted average, where each node’s weight is equal to the number of training samples that are associated with it.

## 2.3.6 K-Nearest Neighbor Regression

This is the K-Nearest Neighbor (KNN) solution, which performs exceptionally well. It takes two forms: a regression, where we want a value, or a classification. To apply KNN to our problem of restaurant Status, we would just have to find the nearest K neighbors.

The KNN algorithm was originally introduced by Drs. Evelyn Fix and J. L. Hodges Jr, in an unpublished technical report written for the U.S. Air Force School of Aviation Medicine. Fix and Hodges’ original research focused on splitting up classification problems into a few subproblems:

Sklearn.neighbors gives usefulness to solo and directed neighbors-based learning techniques. Solo closest neighbors are the establishment of numerous other learning techniques, remarkably complex learning and otherworldly grouping. Regulated neighbors-based learning comes in two flavors: order for information with discrete marks, and relapse for information with ceaseless names.

K-Nearest Neighbours (KNN) is a form of instance-based learning where their classification is provided with a set of training instances, a new example can be categorized according to k nearest neighbours, determined by Euclidean distance, the closest given instance from the memory coded 10 and figure 1

Figure 1

Chart, scatter chart

Description automatically generated

How did it do that?

Step 1 – Choose the number K of neighbours

Step 2 – Take the K nearest neighbours of the new data point, according to the **Euclidean Distance**

(the distance between two instances and is defined to be (where

( = 2 )

Step 3 – Among these K neighbours, count the number of data points in each category

Step 4 – Assign the new data point to the category where we counted the most neighbours

Step 5 – Our model is ready

Let K(x) denote the indices of the data points that are the K nearest to x for any x in p-dimensional real space. In a problem with a regression: -

where δ (v, g(xi)) = 1 if v= g(xi)

δ (v, g(xi)) = 0 otherwise

*V* is a set of all values of g*(x)* (i.e. all classes)

argmax f(x) returns the value of x which maximises f(x)

To predict the classification of the new instance when selecting the nearest k, the average or plurality class vote is used with training examples from categories 1 and 2. It gives closer neighbors more weight.

KNN assumes that the cases close in distance are equivalent, unlike the Naive Bayes classifier that needs prior knowledge of data distribution, and this is valid in terms of simplicity and intuition for most problems. It is possible to control model complexity by k, Which restricts sensitivity to noise and overfitting. The hyperparameter value of k has a significant effect on the algorithm's performance. The lower the value of k, the complexity of the model is greater. Usually, the square root of the number of training examples - coded 46 - is set to be the ideal k. Provided that the KNN is susceptible to minimal, negligible or redundant quantities which, owing to similar distances, can be applied to become relevant at all points, coded 47. In order to prevent this drawback, careful selection of features is an important practice. KNN requires long prediction time and high machine costs, but it is still marginally fair to trade off its overall satisfactory classification performance in our case.

The guideline behind closest neighbor techniques is to discover a predefined number of preparing tests nearest in separation to the new point and anticipate the mark from these. The quantity of tests can be a client characterized consistent (k-closest neighbor learning) or fluctuate dependent on the nearby thickness of focuses (range-based neighbor learning). The separation can, by and large, be any measurement measure: standard Euclidean separation is the most widely recognized decision. Neighbors-based techniques are known as non-summing up AI strategies, since they basically "recollect" the entirety of its preparation information (conceivably changed into a quick ordering structure, for example, a Ball Tree or KD Tree).

Regardless of its effortlessness, closest neighbors have been fruitful in countless grouping and relapse issues, including transcribed digits and satellite picture scenes. Being a non-parametric technique, it is frequently effective in arrangement circumstances where the choice limit is sporadic.

The classes in sklearn.neighbors can deal with either NumPy exhibits or scipy.sparse lattices as information. For thick lattices, an enormous number of conceivable separation measurements are upheld. For scanty grids, discretionary Minkowski measurements are upheld for look.

There are many learning schedules which depend on closest neighbors at their center. One model is portion thickness assessment, talked about in the thickness assessment area.

Neighbors-based relapse can be utilized in situations where the information names are ceaseless instead of discrete factors. The name doled out to a question point is registered dependent on the mean of the marks of its closest neighbors.

scikit-learn actualizes two distinct neighbors regressors: KNeighborsRegressor executes learning dependent on the closest neighbors of each question point, where is a whole number worth indicated by the client. RadiusNeighborsRegressor executes learning dependent on the neighbors inside a fixed span of the inquiry point, where is a drifting point esteem indicated by the client.

The fundamental closest neighbors relapse utilizes uniform loads: that is, each point in the nearby area contributes consistently to the characterization of a question point. Under certain conditions, it tends to be invaluable to weight focuses with the end goal that close by focuses offer more to the relapse than faraway focuses. This can be refined through the loads catchphrase. The default esteem, loads = 'uniform', allocates equivalent loads to all focuses. loads = 'separation' allots loads relative to the backwards of the good ways from the inquiry point. On the other hand, a client characterized capacity of the separation can be provided, which will be utilized to figure the loads.

## 

## 2.3.7 Artificial neural networks (ANN) - MLP Regression

ANNs are at the very core of Deep Learning. They are versatile, powerful, and scalable, making them ideal to tackle large and highly complex ML tasks, such as classifying billions of images (e.g., Google Images), powering speech recognition services (e.g., Apple’s Siri), recommending the best videos to watch to hundreds of millions of users every day (e.g., YouTube).

ANN architectures leading up to Multi-layer Perceptron (MLP) is a regulated learning calculation that learns a capacity via preparing on a dataset, where is the quantity of measurements for input and is the quantity of measurements for yield. The Perceptron is one of the simplest ANN architectures, invented in 1957 by Frank Rosenblatt. It is based on a slightly different artificial neuron called a threshold logic unit (TLU), or sometimes a linear threshold unit (LTU): the inputs and output are now numbers (instead of binary on/off values) and each input con‐ nection is associated with a weight. The TLU computes a weighted sum of its inputs (z = w1 x1 + w2 x2 + ⋯ + wn xn = xT w), then applies a step function to that sum and outputs the result: hw(x) = step(z), where z = xT w.

In the Multilayer perceptron, there can more than one linear layer (combinations of **neurons**). If we take the simple example the three-layer network, first layer will be the *input layer* and last will be *output layer* and middle layer will be called *hidden layer.*We feed our input data into the input layer and take the output from the output layer. We can increase the number of the hidden layer as much as we want, to make the model more complex according to our task.

Diagram

Description automatically generated

An MLP is composed of one (passthrough) input layer, one or more layers of TLUs, called hidden layers, and one final layer of TLUs called the output layer. The layers close to the input layer are usually called the lower layers, and the ones close to the outputs are usually called the upper layers. Every layer except the output layer includes a bias neuron and is fully connected to the next layer.

## **Chapter Summary**

Previously no specific researches were available for individual restaurants open recommendations using supervised data. The research on sale predictions is done previously and also discussed here. Supervised and Unsupervised algorithms are discussed.

# **CHAPTER 3:**

# **RESEARCH METHODOLOGY**

## **Introduction**

This chapter is to introduce the comprehensive view on the data engine based on ML model of this research. The whole process obeys the lifecycle of data science: from problem statement, raw data, to data engineering, model building, evaluation and parameters tunings. At last, the result visualization and model deployment.

## **Data Science Analysis (EDA)**

In a work offered the lifecycle of data analysis and development. We often want a numerical prediction over a categorical target. In current Restaurant Status research, a basic yes/no prediction of whether a restaurant is likely to continue for keep the business running may not be sufficient; we want to model the probability that the restaurant will continue. This is still considered classification modeling rather than regression because the underlying target is categorical. The process includes main data from Ambros, Hong Kong government and property agency starting from identifying the problem to completed in result – deployment and insights illustration. There are six main stages including understanding the data, selecting appropriate tools and methodologies, data preparation, building a model, test it and analyze the results. Each stage has own scope of tasks and its own objectives. In the following subchapters will be shown main activities and jobs accomplished in this project at all stages. (Deanne Larson, Victor Chang, 2016)

Understand the Problem

Model and Analyze

Data Preparation

Select the Tool

Deploy the Model

Present the Results

Understand the Data

### **3.2.1 Understand the Data / Data Preparation**

The dataset available is containing the information of different restaurants of Hong Kong. The dataset has features based on ages, areas and different type of payments. With all these there were features of total sales and restaurant status. This makes the data supervised.

Obviously, if Ambros gave us those data is full of errors, outliers, and noise (e.g., due to poor- quality measurements), it will make it harder for the ML to detect the underlying patterns, so it is less likely to perform well. It is often well worth the effort to spend time cleaning up our data. The truth is, most data scientists spend a significant part of their time doing just that.

For example:

• If some instances are clearly outliers, it may help to simply discard them or try to fix the errors manually.

• If some instances are missing a few features, we must decide whether we want to ignore this attribute altogether, ignore these instances, fill in the missing values (e.g., with the location or age), or train one model with the feature and one model without it, and so on.

Garbage in, garbage out!

We will only be capable of learning if the training data contains enough relevant features and not too many irrelevant ones. A critical part of the success of a ML project is coming up with a good set of features to train on. This process, called feature engineering, involves:

• Feature selection: selecting the most useful features to train on among existing features.

• Feature extraction: combining existing features to produce a more useful one (as we saw earlier, dimensionality reduction algorithms can help).

• Creating new features by gathering new data.

Reviewing Ambros’s private data, I guessed it could be underfitting data. It occurs when the model is too simple to learn the underlying structure of the data. The key actions to fix this problem are:

• Selecting a more powerful model, with more parameters add on from public data. Eg. Government and Property agency data

• Feeding better features to the learning algorithm (feature engineering)

• Reducing the constraints on the model (e.g., reducing the regularization hyper‐ parameter)

The next step is

1. Look at the big picture.

2. Combine all private and public data.

3. Discover and visualize the data to gain insights.

4. Prepare the data for ML algorithms.

5. Select a model and train it.

6. Fine-tune the model.

7. Present the solution.

### **3.2.2 Frame the Problem**

The first question to ask is what exactly is the business objective; building a model is probably not the end goal. How does the Ambros expect to use and benefit from the models? This is important because it will determine how we frame the problem, what algorithms we will select, what performance measure we will use to evaluate our model, and how much effort we should spend tweaking it. The next question to ask is what the current solution looks like (if any). It will often give us a reference performance, as well as insights on how to solve the problem, with all this information we are now ready to start designing our code. First, we need to frame the problem: is it supervised, unsupervised, or Reinforce Learning? Is it a classification task, a regression task, or something else? Should we use batch learning or online learning techniques?

The main problem is to predict whether the location will be appropriate for opening a restaurant or not. Also, the main problem is the predict the total sales predictions of different restaurants.

Select a Performance Measure

Our next step is to select a performance measure. A typical performance measure for regression problems is the Root Mean Square Error (RMSE). It gives an idea of how much error the code typically makes in its predictions, with a higher weight for large errors.

Root Mean Square Error (RMSE)

RMSE(X,h) =

### **Select the Tool**

There are various programing languages and interfaces that could provide us the implementation of ML algorithms to classify the normal source packets from the attack source packets. One of the widely used language for this purpose is Python. Jupyter Notebook is an interactive interface provided by Anaconda navigation to make code sequential and make flow interactive and understandable.

#### **Python Programming Language:**

The Python programming language is setting up itself as one of the most well-known dialects for logical processing. Because of its significant level intuitive nature and its developing environment of logical libraries, it is an engaging decision for algorithmic turn of events and exploratory information investigation (K. Jarrod Millman , Michael Aivazis, 2011). However, as a universally useful language, it is progressively utilized in scholastic settings as well as in industry.

#### **Jupyter Notebook:**

The Jupyter stretches out the comfort-based way to deal with intuitive figuring a subjectively new way, giving an online application reasonable for catching the entire calculation measure: creating, archiving, and executing code, just as imparting the outcomes. (Jupyter Notebook Documentation, n.d.)The Jupyter notebook joins two segments:

A web application: a program-based device for intuitive composing of reports which join informative content, science, calculations and their rich media yield.

Notebook Documents: a portrayal of all substance obvious in the web application, including information sources and yields of the calculations, logical content, science, pictures, and rich media portrayals of items.

### **Libraries:**

Following Important Libraries were used for the purpose of DoS attack deduction source from the generated dataset.

#### **Pandas:**

Pandas is a Python library of rich information structures and apparatuses for working with organized informational collections basic to insights, fund, sociologies, and numerous different fields. The library gives incorporated, natural schedules for performing basic information controls and investigation on such informational indexes. It plans to be the fundamental layer for the fate of measurable registering in Python. It fills in as a solid supplement to the current logical Python stack while actualizing and enhancing the sorts of information control instruments found in other measurable programming dialects, for example, R. (Wes McKinney, 2011)

#### **Plotly:**

Plotly is a powerful information perception programming that can cycle exceptionally progressed examination and perform computerized errands without the requirement for JavaScript. Its apparatuses and highlights are ideal for clients that manage large information and exploration that need powerful examination and AI. Clients can transfer, examine, and imagine boundless quantities of information and results with prepared formats and standard recipes. It has an open-source diagramming library that has more than 4,000,000 months to month downloads. Consistent availability contributes essentially to the constant improvement of information science. Every one of its devices and highlights are versatile. Its use is endless as it is utilized by the two understudies and global organizations like Shell, Apple, Cisco, Tesla, and NVIDIA. Global organizations emphatically depend on Plotly with regards to preparing Python and R models. Plotly has an open API that takes into consideration absolute customization over all endpoints. (Camp, n.d.)

#### **Matplotlib:**

Matplotlib is an open-source library that guides in chart plotting. It was at first composed by John D. Tracker, who happened to be a neurobiologist. He composed Matplotlib at the hour of his post-doctoral exploration in Neurobiology. The point of this library was to examine the exercises happening in the cerebral cortex of patients experiencing epilepsy by plotting these exercises in a diagram. The sole reason for plotting diagrams was for better representation and for examining the basic examples in them. The main arrival of Matplotlib was in 2003. After some time Matplotlib ended up being one of the most broadly utilized plotting libraries close by the Python programming language for information and computational diagram plotting. It is stage autonomous and can be run on Windows, Mac OS, and Linux. (Educba, n.d.)

#### **Scikit Learn:**

Scikit-learn bridles this rich condition to give cutting edge usage of many notable AI calculations, while keeping up a simple to-utilize interface firmly coordinated with the Python language. This answers the developing requirement for measurable information examination by non-authorities in the product and web enterprises, just as in fields outside of software engineering, for example, science or physical science. Scikit-take in varies from other AI tool compartments in Python for different reasons: I) it is circulated under the BSD permit ii) it consolidates arranged code for effectiveness, dissimilar to MDP (Tiziano Zito, Niko Wilbert, Laurenz Wiskott and Pietro Berkes, 2008)and pybrain (Tom Schaul, Justin Bayer ,Daan Wierstra,Yi Sun, 2010) it relies just upon numpy and scipy to encourage simple conveyance, not at all like pymvpa (Michael Hanke, Yaroslav O. Halchenko, Per B. Sederberg, Stephen José Hanson, James V. Haxby & Stefan Pollmann , 2009) that has discretionary conditions, for example, R and shogun, and iv) it centers around basic programming, not at all like pybrain which utilizes an information stream system. While the bundle is generally written in Python, it consolidates the C++ libraries LibSVM (Chang, Chih-Chung, and Chih-Jen Lin., 2001) and LibLinear (Fan, Rong-En, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin., 2008) that give reference usage of SVMs and summed up straight models with viable licenses. Paired bundles are accessible on a rich arrangement of stages including Windows and any POSIX stages.

Scikit-learn is a Python module incorporating a wide scope of best in class AI calculations for medium-scale directed and solo issues. This bundle centers around bringing AI to non-experts utilizing a broadly useful elevated level language. Accentuation is put on usability, execution, documentation, and API consistency. It has insignificant conditions and is appropriated under the improved BSD permit, empowering its utilization in both academic and business settings. (Fabian Pedregosa,Olivier Grisel ,Mathieu Blondel , 2011)

### **Machine Learning Theory:**

#### **Classification:**

To differentiate an attacking source and safe source. Classification technique of ML is to be used for the purpose.

Classification is method to classify our information into an ideal and particular number of classes where we can dole out name to each class. It is the assignment of summing up known structure to apply to new information while bunching is the assignment of finding gatherings and structures in the information that are here and there or another comparable, without utilizing known structures in the information. (Suthaharan, 2014)

Uses of Classification are: discourse acknowledgment, penmanship acknowledgment, biometric ID, archive order and so forth.

### **3.2.4 Data Preparation**

Data was prepared for train and test while selecting the important features using human intelligence. These features were important as they all were helping in basis of Restaurant Open and Sales.

### **3.2.5 Model Analyze**

Models were analyzed using accuracy and confusion matrix and confusion report for classification matrix and mean squared errors for the regression models.

### **3.2.6 Present the Results**

The results were presented using plotly library. The accuracy for all classification models were printed togather to show the best model. The mean squared error for all regression algorithms was plotted togather to get the best algorithm with lowest error.

### **3.2.7 Deploy the Model**

The best models for both cases were then passes through the K Fold Cross Validation Techniques for their final selections.

## **Methodology**

The components of the ML pipeline in our example are problem formulation, data planning, feature engineering, model training, model evaluation, and post-hoc analysis. We must first import all of the required libraries, set the random seed to a fixed value, and organize our workspace before we begin. The project was also implemented in python program and it’s a sequence of code and running it randomly may cause the error as the coding is a step-by-step process. So, it is important to consider this while running the code.



Figure 1: Importing Libraries

The libraries are discussed in previous sections were imported at the first place. Warnings are used to avoid any warning that may occur during coding process.

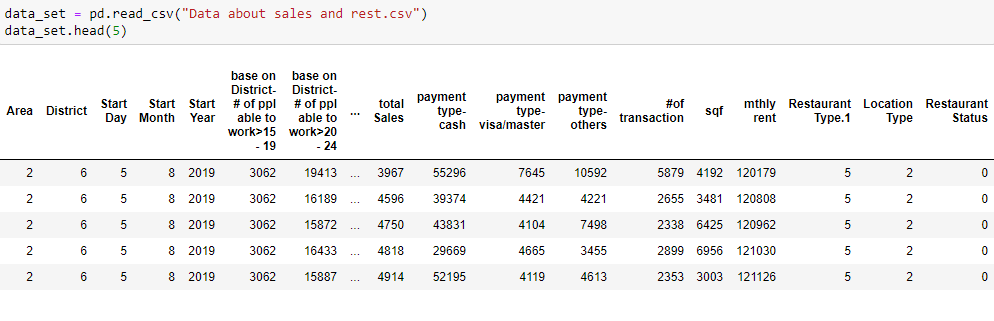


Figure 2: Reading the dataset

The dataset was read using the pandas read csv method. We know the value for the target attribute. It is a classification problem because the target is a category (yes or no) for Restaurant Status which I had changed to 1 = yes, 0 = no when I did the data mining.

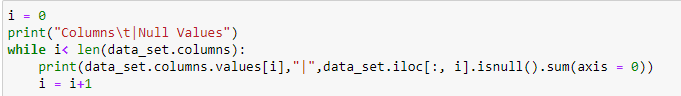


Figure 3: Checking Null Values in each column

Null values for each column were checked using the code in figure 3.

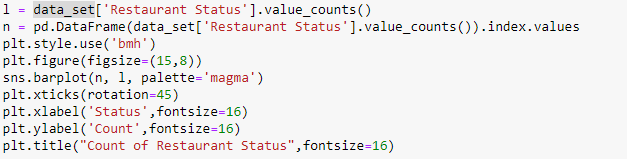


Figure 4: Viewing the overall status

For checking how much data of Restaurant status were available the value counts for both open and close was done.



Figure 5: Code for Boxplot

Boxplot code is used to get insights of each feature that will help for overall status of the data from business point of view.

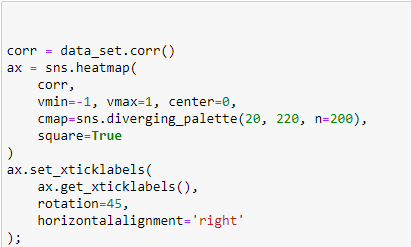


Figure 6: Co relation between variables

The co relation between all variables was checked and plotted using the above code.

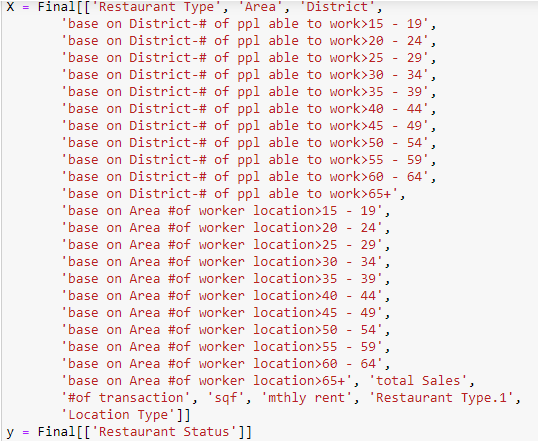


Figure 7: Selection of X and Y for classification algorithm

First step was to use the Classification Algorithms to predict the restaurant stats. In figure 7 the target and the inputs were splitter into X and y.



Figure 8: Train and test splitting the data

33 % of the data was chosen as the test size to get the maximam accuracy in future.

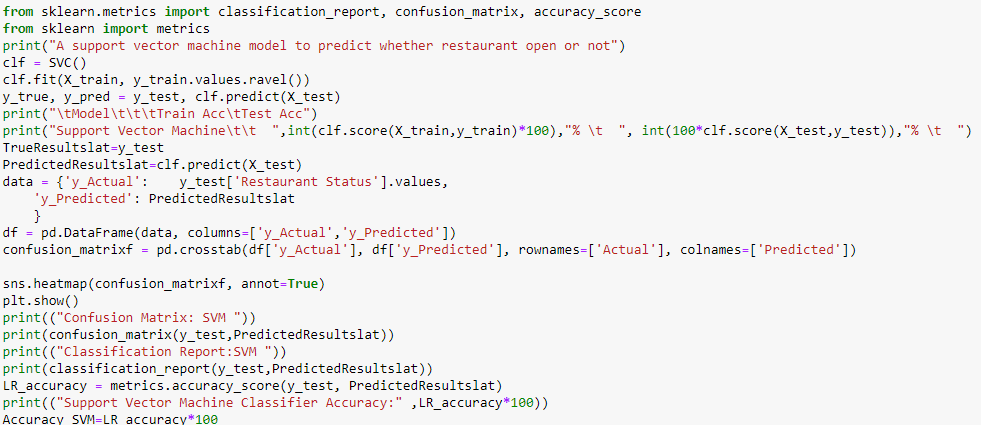


Figure 9: Support vector machine for restaurant status Prediction

The above code is showing the implementation of ML Algorithms for restaurant predictions.



Figure 10: Logistic Regression for restaurant status Prediction

The above algorithm is showing the implementation of Logistics Regression for the restaurant status prediction.

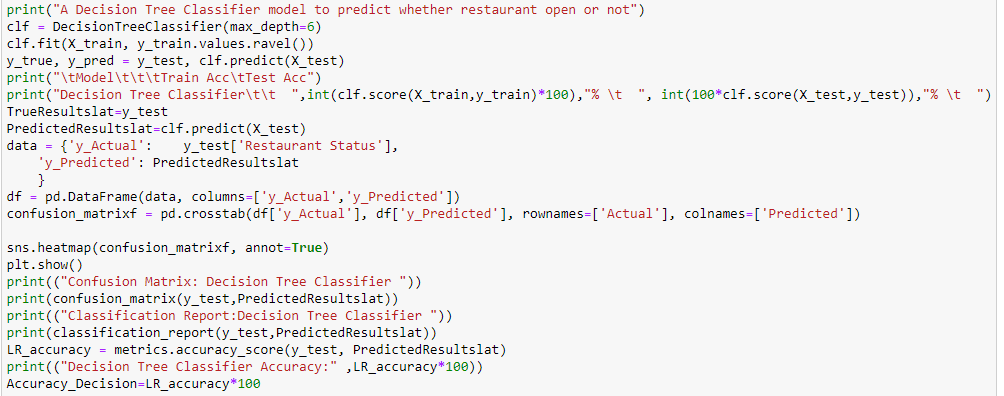


Figure 11: Decision tree Classifier for restaurant status prediction

The above code is showing the implementation of Decision Tree Classification for the restaurant status prediction.

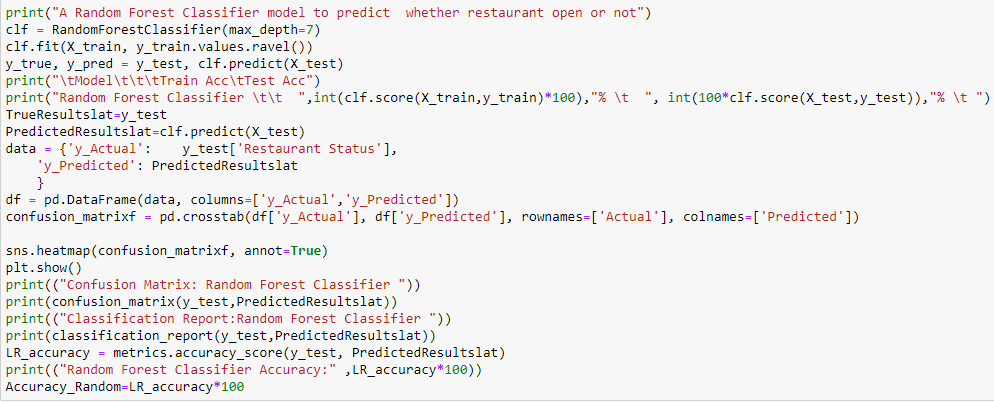


Figure 12: Random Forest Classifier for restaurant status prediction

The above code is showing the implementation of Random Forest Classification for the restaurant status prediction.

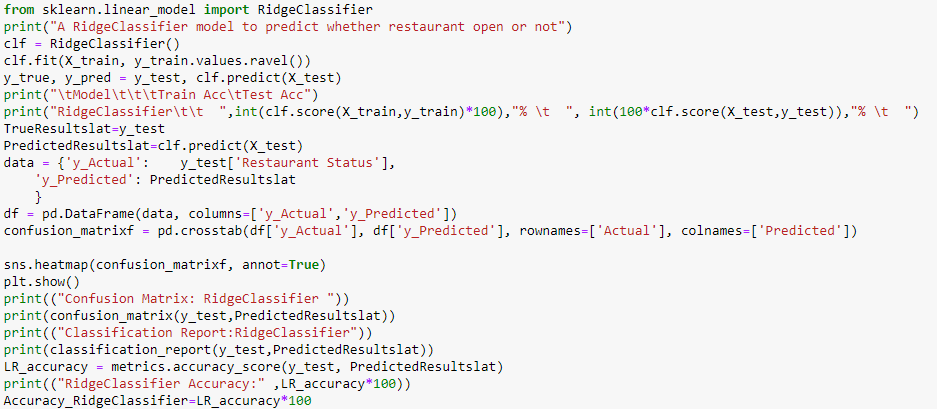


Figure 13: Ridge Classifier for restaurant status prediction

The above code is showing the implementation of Ridge Classification for the restaurant status prediction.

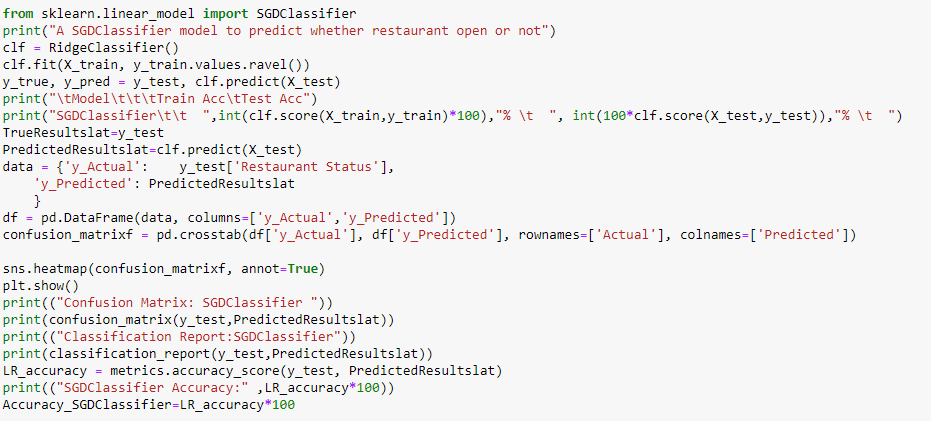


Figure 14: SGD Classifier for restaurant status prediction

The above code is showing the implementation of Stochastic Gradient Descent Classification for the restaurant status prediction.

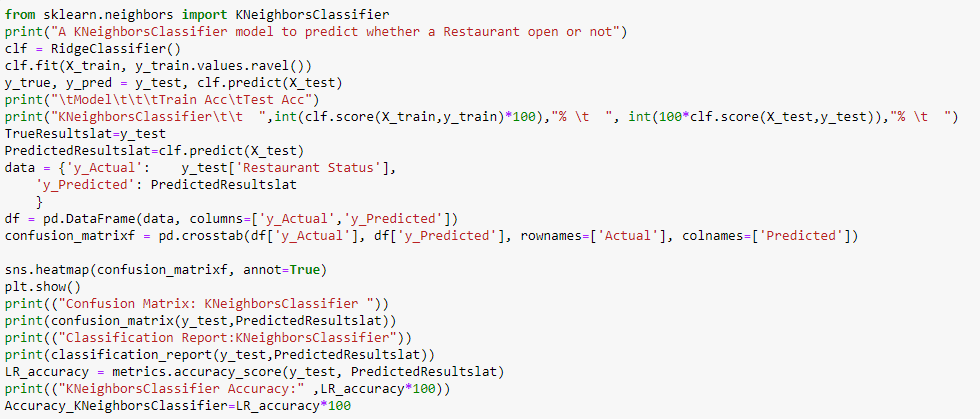


Figure 15: K Neighbor Classifier for restaurant status prediction

The above code is showing the implementation of K Neighbor Classification for the restaurant status prediction.

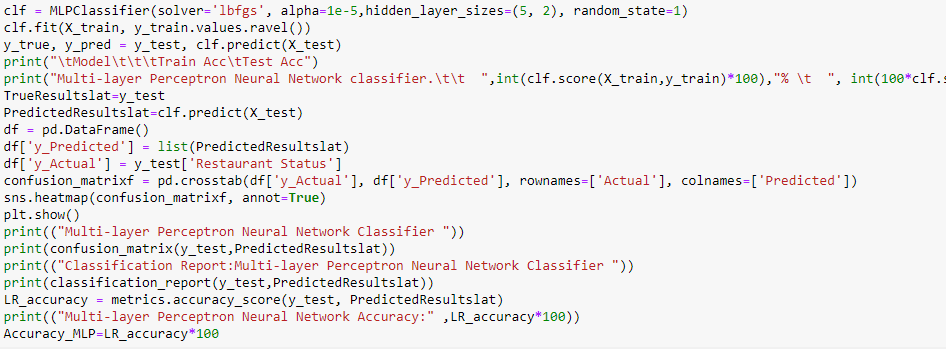


Figure 16: Multi-layer perception neural network for restaurant status prediction

The above code is showing the implementation of MLP on for the restaurant status prediction.

### Deep Sequential Neural Network

We used keras library for Neural Network. We create a [Sequential model](https://keras.io/models/sequential/) and add layers one at a time.

The first thing we ensure the input layer has the right number of input features. This is specified when creating the first layer with the **input\_dim** argument and setting it to 8 for the 8 input variables.

We need a network large enough to capture the structure of the problem.We use a fully-connected network structure with three layers.

Fully connected layers are defined using the [Dense class](https://keras.io/layers/core/). We specify the number of neurons or nodes in the layer as the first argument and specify the activation function using the **activation** argument.

We use the [rectified linear unit activation function](https://machinelearningmastery.com/rectified-linear-activation-function-for-deep-learning-neural-networks/) referred to as ReLU on the first two layers and the Sigmoid function in the output layer.

It used to be the case that Sigmoid and Tanh activation functions Oure preferred for all layers. We used better performance using the ReLU activation function. We use a sigmoid on the output layer to ensure our network output is between 0 and 1 and easy to map to either a probability of class 1 or snap to a hard classification of either class with a default threshold of 0.5.

We piece it all together by adding each layer:

* The model expects rows of data with 8 variables (the input\_dim=8 argument)
* The first hidden layer has 12 nodes and uses the relu activation function.
* The second hidden layer has 8 nodes and uses the relu activation function.
* The output layer has one node and uses the sigmoid activation function.

The shape of the input to the model is defined as an argument on the first hidden layer. This means that the line of code that adds the first Dense layer is doing 2 things, defining the input or visible layer and the first hidden layer.

Compiling the model uses the efficient numerical libraries under the covers (the so-called backend) such as Theano or TensorFlow. The backend automatically chooses the best way to represent the network for training and making predictions to run on hardware, such as CPU or GPU or even distributed.

When compiling, we specify some additional properties required when training the network. Training a network means finding the best set of weights to map inputs to outputs in our dataset.

We specify the loss function to use to evaluate a set of weights, the optimizer is used to search through different weights for the network and any optional metrics we would like to collect and report during training.

We use cross entropy as the **loss** argument. This loss is for a binary classification problems and is defined in Keras as “**binary\_crossentropy**“.

We define the **optimizer** as the efficient stochastic gradient descent algorithm “**adam**“. This is a popular version of gradient descent because it automatically tunes itself and gives good results in a wide range of problems.

It is a classification problem, we collect and report the classification accuracy, defined via the **metrics** argument.

We have defined our model and compiled it ready for efficient computation.

We execute the model on some data.

We train or fit our model on our loaded data by calling the **fit()** function on the model.

Training occurs over epochs and each epoch is split into batches.

* **Epoch**: One pass through all of the rows in the training dataset.
* **Batch**: One or more samples considered by the model within an epoch before weights are updated.

One epoch is comprised of one or more batches, based on the chosen batch size and the model is fit for many epochs. For more on the difference between epochs and batches, see the post:

The training process run for a fixed number of iterations through the dataset called epochs, that we specify using the **epochs** argument. We also set the number of dataset rows that are considered before the model weights are updated within each epoch, called the batch size and set using the **batch\_size** argument.

For this problem, we run for a small number of epochs (150) and use a relatively small batch size of 10.

We want to train the model enough so that it learns a good (or good enough) mapping of rows of input data to the output classification. The model always has some error, but the amount of error level out after some point for a given model configuration. This is called model convergence.

We have trained our neural network on the train dataset and we evaluate the performance of the network on the test dataset.

We evaluate Our model on Our testing dataset using the **evaluate ()** function on Our model and pass it the same input and output used to train the model.

This generate a prediction for each input and output pair and collect scores, including the average loss and any metrics We have configured, such as accuracy.

The **evaluate ()** function return a list with two values. The first be the loss of the model on the dataset and the second be the accuracy of the model on the dataset. We are only interested in reporting the accuracy, so we ignore the loss value.

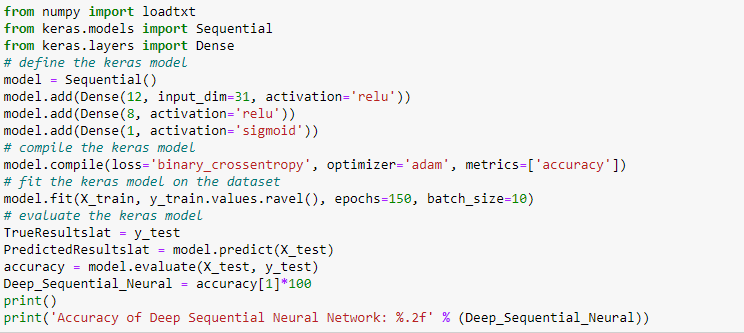


Figure 17: Deep Sequential neural network for restaurant status prediction

The above code is showing the implementation of deep sequential neural network classifier for the restaurant status prediction.

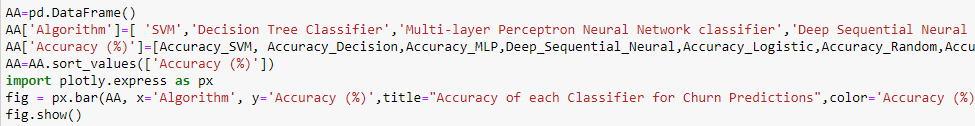


Figure 18: Combined Plot for accuracy of each classifier

Figure 18 is the code for combined plot for each Classifier.



Figure 19: K-fold cross validation on random forest

From visualizations it was checked that Random Forest Classification is the best algorithm for this project. So, K Fold cross validation is done using sklearn K Fold Model.

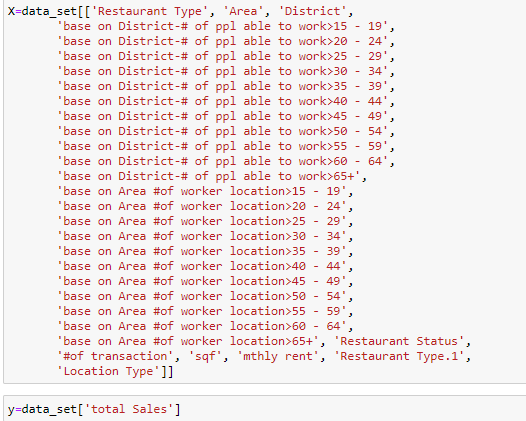


Figure 20: Selection of Variables for Regression Algorithms

The above code is showing the variable selection for the total sales prediction.



Figure 21: Train Test split on regression algorithms

The above code is showing the train and test splits of each algorithm.

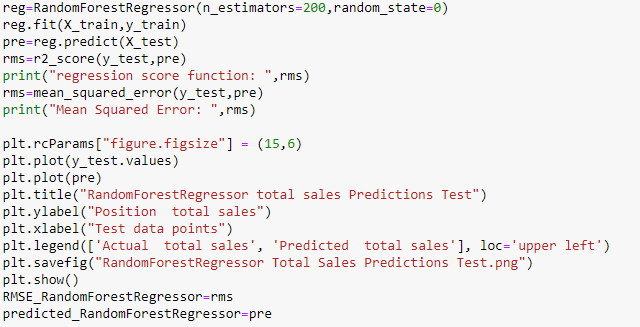


Figure 22: Random forest regression for total sales prediction

The above code is showing the implementation of Random Forest Regression for total sales predcition.

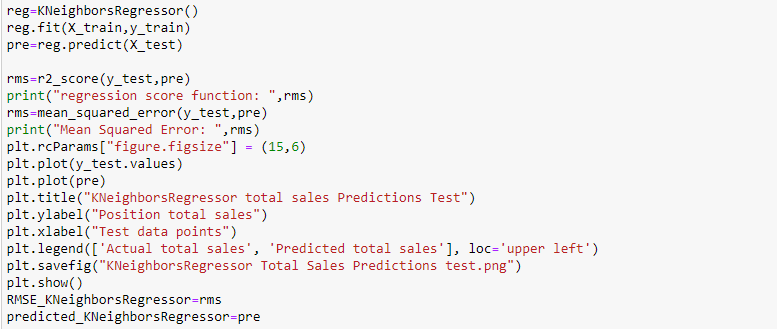


Figure 23: K Neighbor Regression for total sales Prediction

The above code is showing the implementation of K Neighbor Regression for total sales predcition.

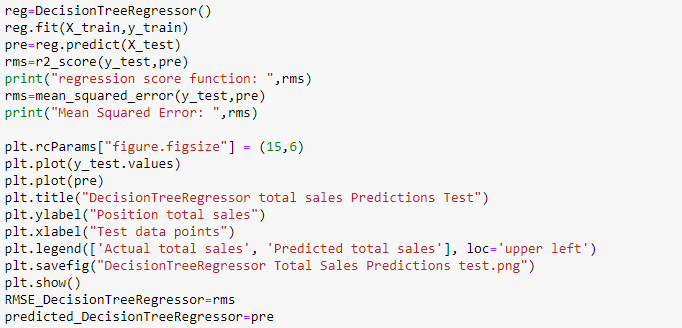


Figure 24: Decision Tree Regression for Total sales prediction

The above code is showing the implementation of Decision Tree Regression for total sales predcition.

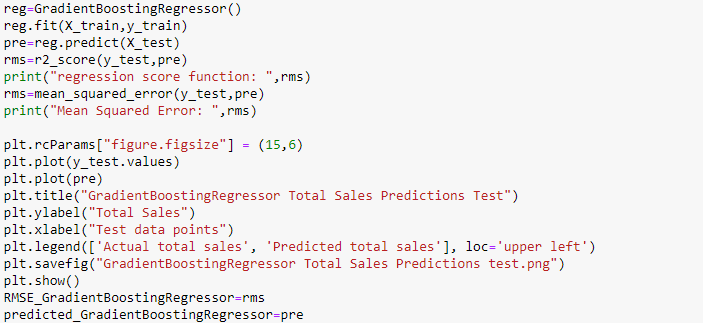


Figure 25: Gradient Boosting Regression for Total Sales Predictions

The above code is showing the implementation of Gradient Boosting Regression for total sales predcition.

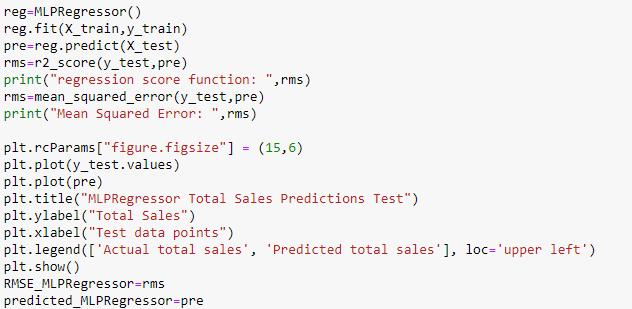


Figure 26: MLP Regression for total sales prediction

The above code is showing the implementation of MLP Regression for total sales predcition.

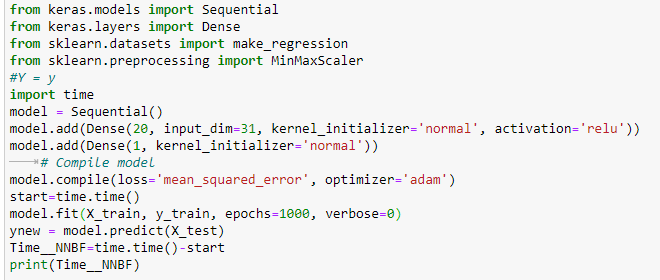


Figure 27: Neural Networks for total sales predictions

The above code is showing the implementation of Deep Sequential Neural Network Regression for total sales predcition.

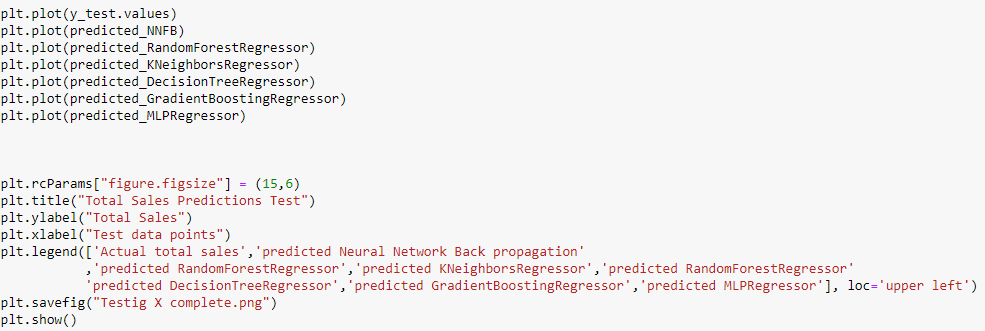


Figure 28: Code for combined plot of origional and predicted total sales for all algorithms

The above code is showing the combined plot for each algorithm.

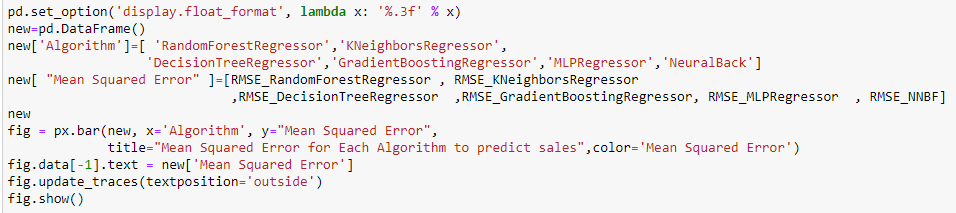


Figure 29: Code for combined mean square error plot for regression algorithms

The combined plot for mean squared error for each algorithm was plotted using this code.

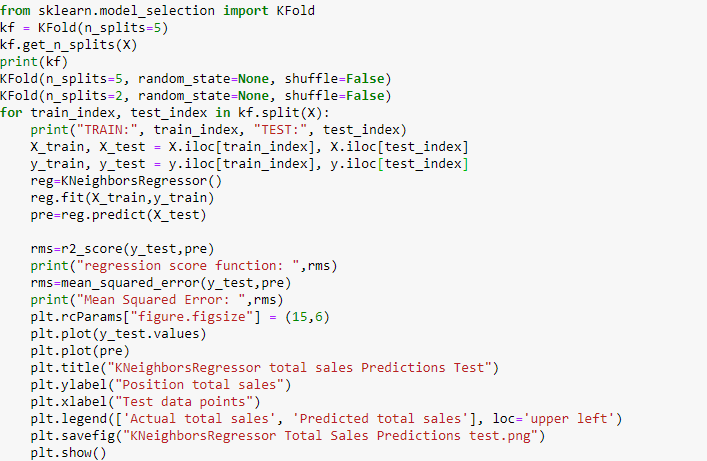


Figure 30: Cross Validation for K Neighbor Regression for total sales prediction

The last step I methodology was to apply cross validation techniques.

## **Procedures and Data Definitions**

## **Evaluation Methods**

### Accuracy Measures techniques used for Classification

- Mainly 4 techniques are used

#### Precision

Precision talks about how precise/accurate our model is out of those predicted positive, how many of them are actual positive. Precision is a good measure to determine, when the costs of False Positive is high.

#### Recall

Recall shall be the model metric we use to select our best model when there is a high cost associated with False Negative.

#### F1- Score

F1 Score might be a better measure to use if we need to seek a balance between Precision and Recall AND there is an uneven class distribution (large number of Actual Negatives).

#### Accuracy

Accuracy shows the combined effect of the precision and recall when both factors are important

Since it is important to know which factors will affect restaurant open and what factors do not, we focused on combined accuracy for the final purpose.

### Accuracy Measures techniques used for Regression

#### Mean Squared Error

For accuracy measurement of regression algorithms mean squared error was calculated.

## **Chapter Summary**

This chapter has explained the methodology of this research, the development from problem statements to model building and evaluation. Also, this development plan is consistent with data product development lifecycle. Especially the neural networks models produces different results compared to ML model. The next chapter will be discussed about the model results.

# **CHAPTER 4:**

# **RESULTS AND EVALUATIONS**

## **Results**

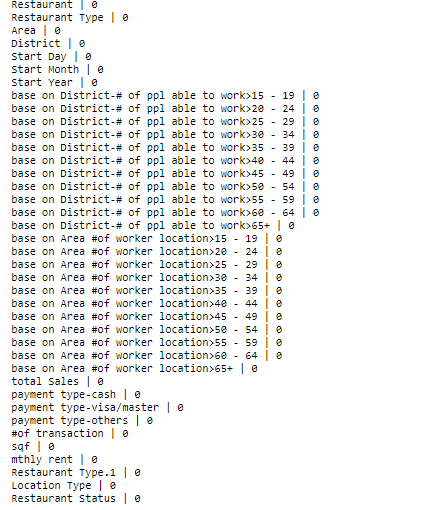


Figure 31: Null Values of each columns

Figure 31 is showing that there is no null value in the dataset.

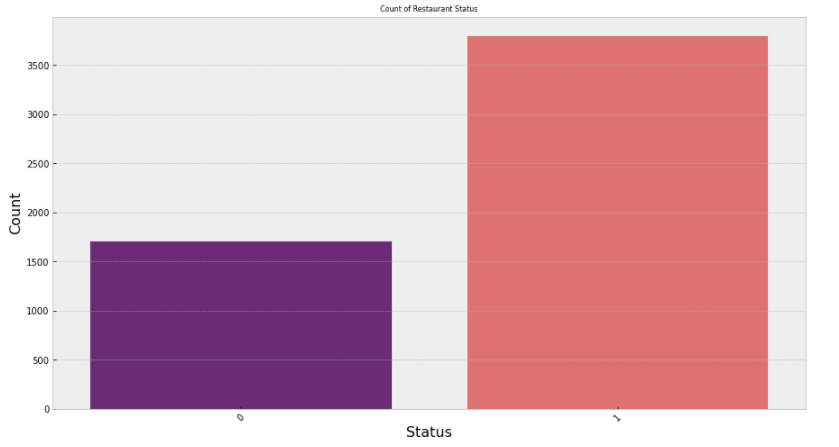


Figure 32: Count of Restaurant Status

Figure 30 is showing that Almost more then 1500 restaurants were not open but more than 3500 restaurants were open.

### Boxplot for distributions of each feature

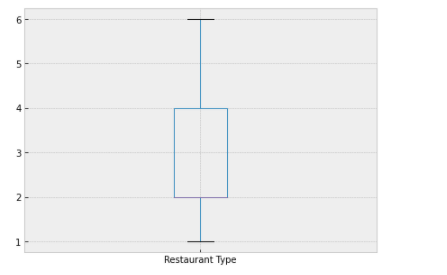


Figure 33 Boxplot visualization for restaurant type

It can be seen that mostly data belongs to first 4 types of restaurants.

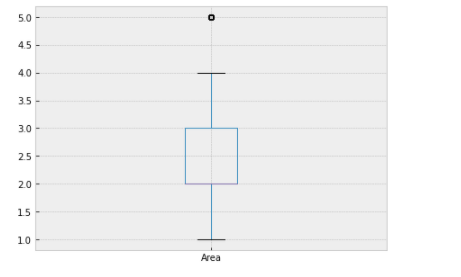


Figure 34: Boxplot visualization for restaurant area

It can be seen that most of the values belongs to first 4 areas instead of the fifth area.

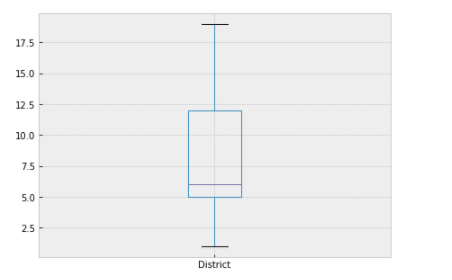


Figure 35: Boxplot visualization for Districts

It can be seen that 80% of the data lies in first 12 districts.

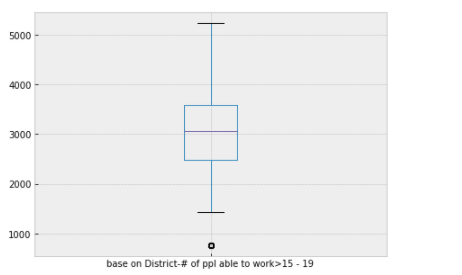


Figure 36: Boxplot of Number of people able to work from age 15 to 19 base on District

It can be seen that in districts there are mostly 1500 to 5500 people are available to work under 19.

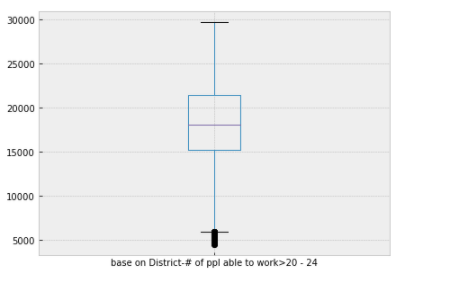


Figure 37: Boxplot of Number of people able to work from age 20 to 24 base on District

It can be seen that in districts there are mostly 6000 to 30000 people are available to work for age 20 to 24.

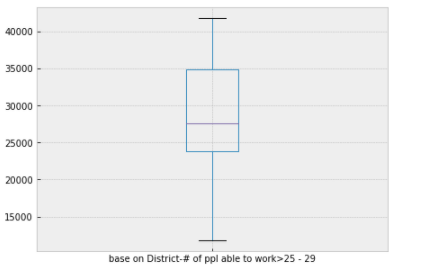


Figure 38: Boxplot of Number of people able to work from age 25 to 29 base on District

It can be seen that in districts there are mostly 12000 to 40000 people are available to work for age 25 to 29.

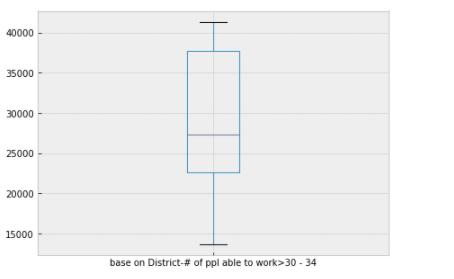


Figure 39: Boxplot of Number of people able to work from age 30 to 34 base on District

It can be seen that in districts there are mostly 12000 to 42000 people are available to work for age 30 to 34.

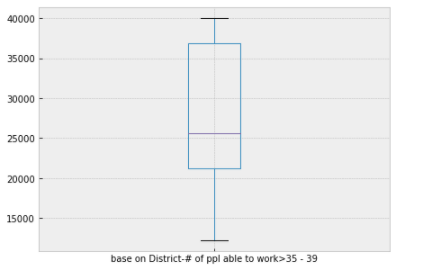


Figure 40: Boxplot of Number of people able to work from age 35 to 39 base on District

It can be seen that in districts there are mostly 12000 to 40000 people are available to work for age 35 to 39.

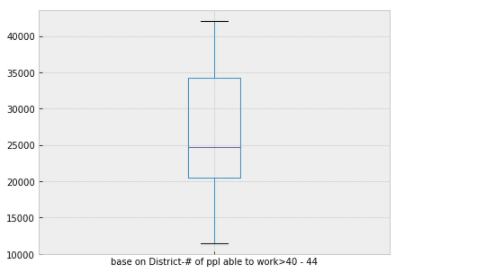


Figure 41: Boxplot of Number of people able to work from age 40 to 44 base on District

It can be seen that in districts there are mostly 12000 to 43000 people are available to work for age 40 to 44.

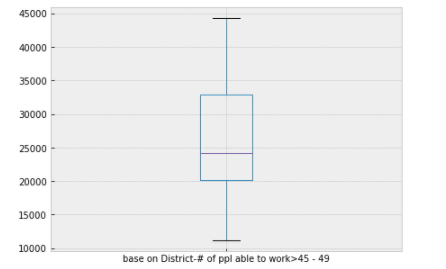


Figure 42: Boxplot of Number of people able to work from age 45 to 49 base on District

It can be seen that in districts there are mostly 11000 to 45000 people are available to work for age 45 to 49.

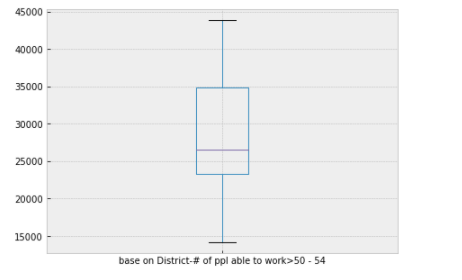


Figure 43: Boxplot of Number of people able to work from age 50 to 54 base on District

It can be seen that in districts there are mostly 14000 to 43000 people are available to work for age 50 to 54.

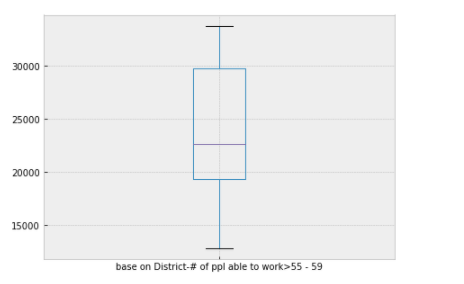


Figure 44: Boxplot of Number of people able to work from age 55 to 59 base on District

It can be seen that in districts there are mostly 12000 to 34000 people are available to work for age 55 to 59.

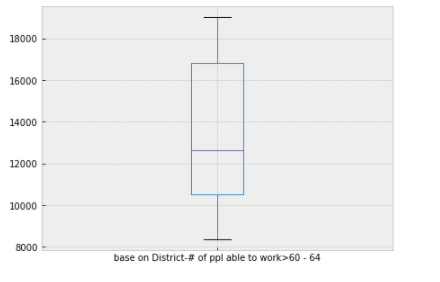


Figure 45: Boxplot of Number of people able to work from age 60 to 64 base on District

It can be seen that in districts there are mostly 8000 to 19000 people are available to work for age 60 to 64.

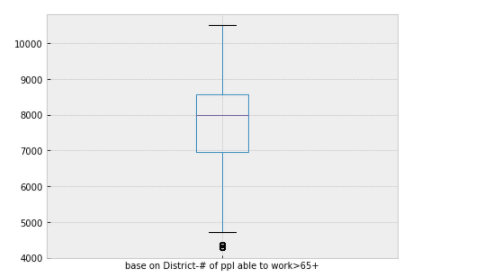


Figure 46: Boxplot of Number of people able to work from age 65 and above base on District

It can be seen that in districts there are mostly 5000 to 11000 people are available to work for age 65 and above.

Overall, It can also be concluded that mostly available for work are of age group are 20 to24.



Figure 47: Boxplot of Number of people able to work from age 15 to 19 base on Area

It can be seen that mostly people below 19 are 7500 to 10500 in any restaurant base on Hong-Kong areas.

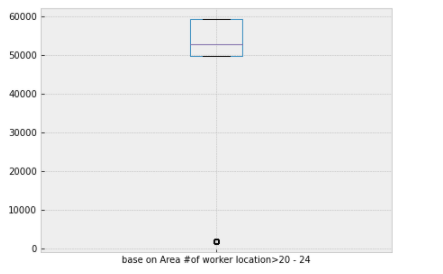


Figure 48: Boxplot of Number of people able to work from age 20 to 24 base on Area

It can be seen that mostly people from age 20 to 24 are 50000 to 60000 in any restaurant base on Hong-Kong areas.

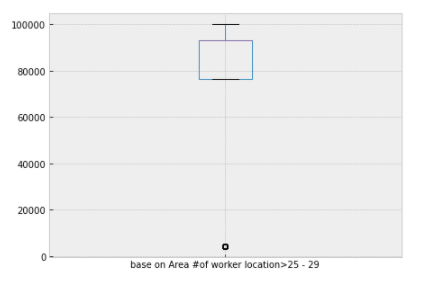


Figure 49:Boxplot of Number of people able to work from age 25 to 29 base on Area

It can be seen that mostly people from age 25 to 29 are 80000 to 95000 in any restaurant base on Hong-Kong areas.

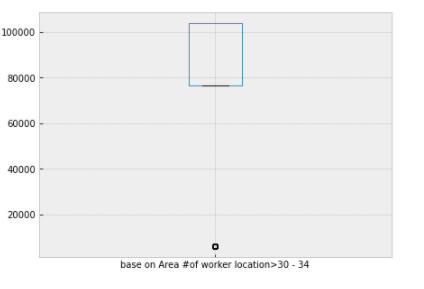


Figure 50: Boxplot of Number of people able to work from age 30 to 34 base on Area

It can be seen that mostly people from age 30 to 34 are 80000 to 100000 in any restaurant base on Hong-Kong areas.

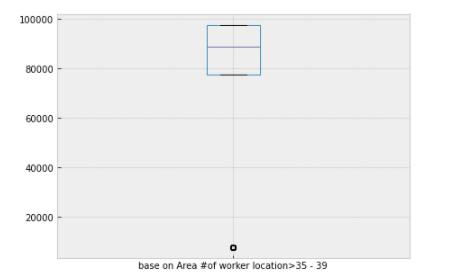


Figure 51: Boxplot of Number of people able to work from age 35 to 39 base on Area

It can be seen that mostly people from age 35 to 39 are 80000 to 980000 in any restaurant base on Hong-Kong areas.

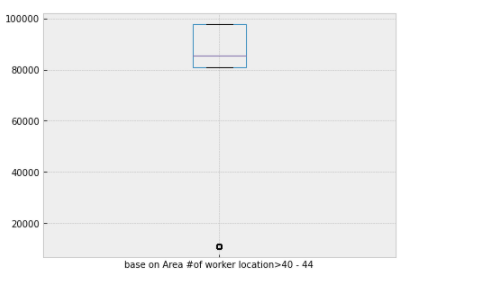


Figure 52: Boxplot of Number of people able to work from age 40 to 44 base on Area

It can be seen that mostly people from age 40 to 44 are 80000 to 980000 in any restaurant base on Hong-Kong areas.



Figure 53: Boxplot of Number of people able to work from age 45 to 49 base on Area

It can be seen that mostly people from age 45 to 49 are 80000 to 980000 in any restaurant base on Hong-Kong areas.

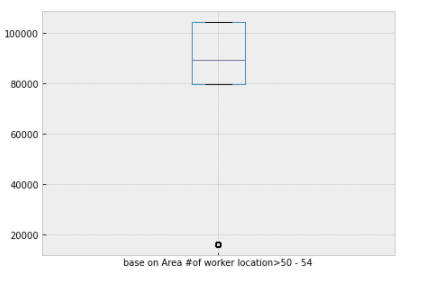


Figure 54: Boxplot of Number of people able to work from age 50 to 54 base on Area

It can be seen that mostly people from age 50 to 54 are 80000 to 101200 in any restaurant base on Hong-Kong areas.

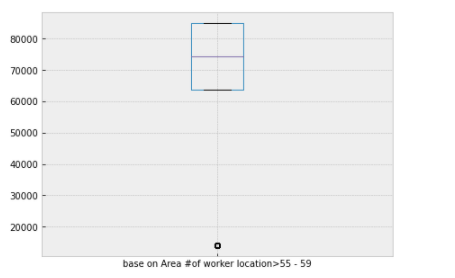


Figure 55: Boxplot of Number of people able to work from age 55 to 59 base on Area

It can be seen that mostly people from age 55 to 59 are 65000 to 85000 in any restaurant base on Hong-Kong areas.

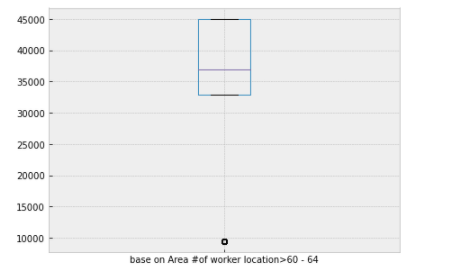


Figure 56: Boxplot of Number of people able to work from age 60 to 64 base on Area

It can be seen that mostly people from age 60 to 64 are 33000 to 45000 in any restaurant base on Hong-Kong areas.

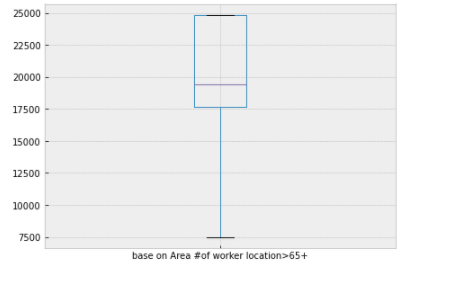


Figure 57: Boxplot of Number of people able to work from age 65 and above base on Area

It can be seen that mostly people from age 65 and above are 17500 to 25000 in any restaurant base on Hong-Kong areas.

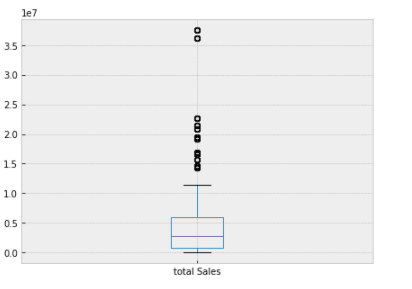


Figure 58: Boxplot of total sales

It can be seen that the total sales are mostly below 10,000,000

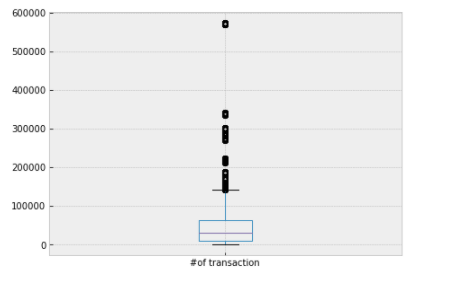


Figure 59: Boxplot of Number of transactions

It can be seen that the number of transactions are mostly below 150000.

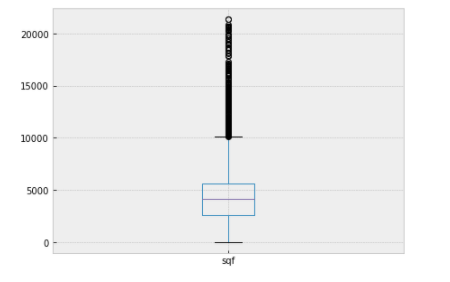


Figure 60: Boxplot of sqf

It can be seen that mostly values of sqf are below 10,000.

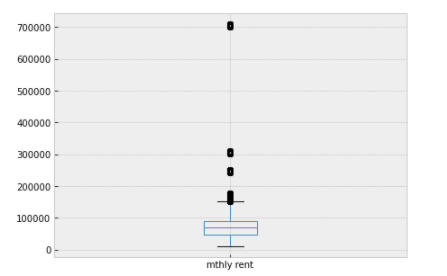


Figure 61: Boxplot of monthly rent

It can be seen that mostly values of monthly rent are below 180,000.

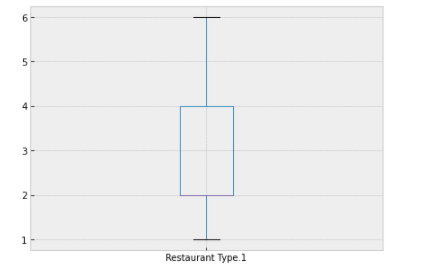


Figure 62: Boxplot of restaurant types

It can be seen that mostly monthly values belong to first 4 types.

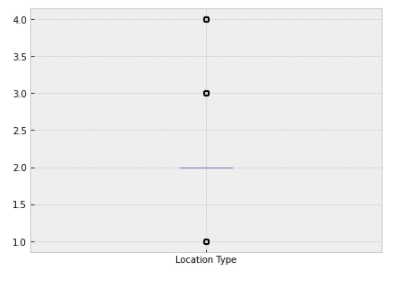


Figure 63: Boxplot of location types

It can be seen that mostly the 2nd location has more users.

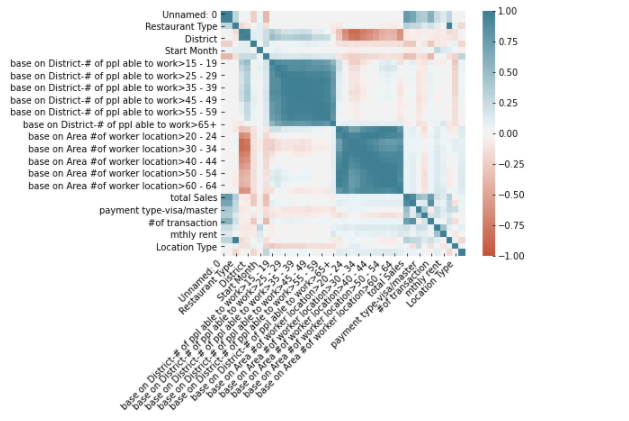


Figure 64: Co relation plot for all features

The above figure us showing that total sales only have good co relation with number of transitions.

### Classification Algorithms Results for Restaurant Status Prediction

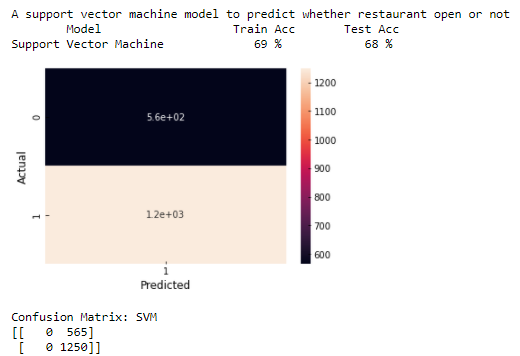


Figure 65: Results of Support Vector Machine Classifier for Restaurant Status Predictions

Figure 65 is showing the output of SVM model testing. It can be seen that model has wrongly predicted 565 not open status as open. Also, the accuracy becomes 68%. But this is showing that SVM isn’t the good algorithm for this purpose.

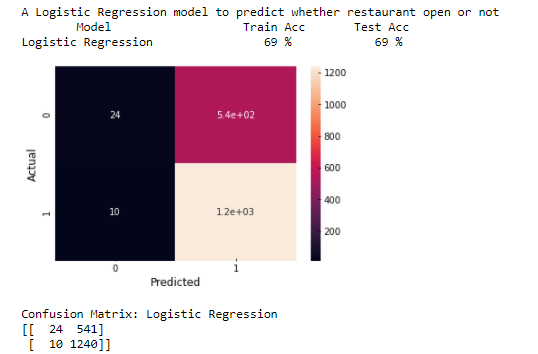


Figure 66: Results of Logistics Regression for Restaurant Status Predictions

Figure 66 is showing the output of Logistic model testing. It can be seen that model has wrongly predicted 542 not open status as open. Also, the accuracy becomes 69%. But this is showing that Logistic Regression isn’t the good algorithm for this purpose.

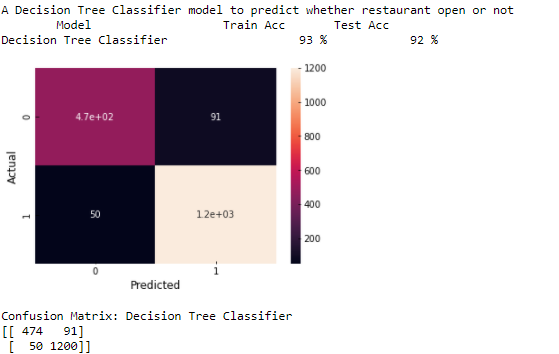


Figure 67: Result of decision tree Classifier for restaurant status prediction

Figure 67 is showing the output of Decision Tree Classifier testing. It can be seen that model has wrongly predicted 91 not open status as open. Also, the accuracy becomes 92%. But this is showing that Decision Tree classifier is a good algorithm for this purpose.

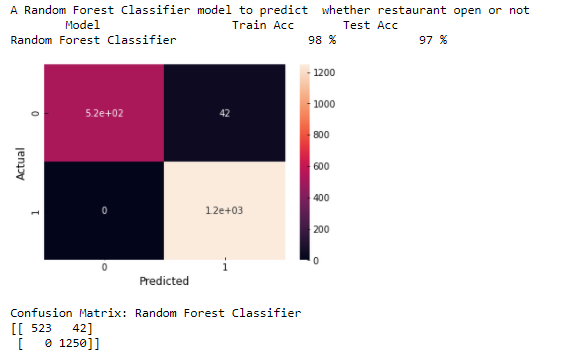


Figure 68: Results of random forest for restaurant status prediction

Figure 68 is showing the output of Random Forest Classifier testing. It can be seen that model has wrongly predicted 42 not open status as open. Also, the accuracy becomes 98%. But this is showing that Random Forest classifier is a good algorithm for this purpose.

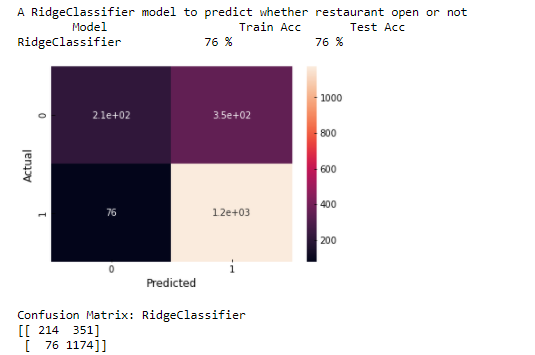


Figure 69: Results of ridge classifier for restaurant status Prediction

Figure 69 is showing the output of Ridge Classifier testing. It can be seen that model has wrongly predicted 351 not open status as open. Also, the accuracy becomes 76%. But this is showing Ridge Classifier is an average algorithm for this purpose.

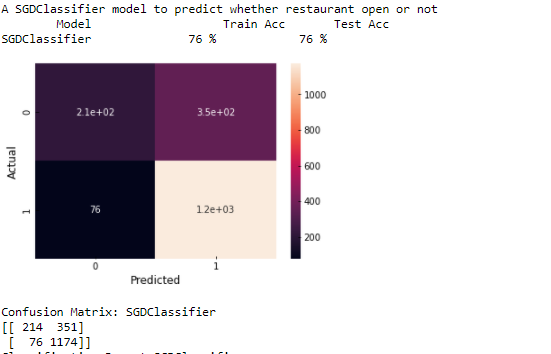


Figure 70: Results of SGD Classifier for restaurant status prediction

The above figure is showing the output of SDG Classifier testing. It can be seen that model has wrongly predicted 351 not open status as open. Also, the accuracy becomes 76%. But this is showing SDG Classifier is an average algorithm for this purpose.

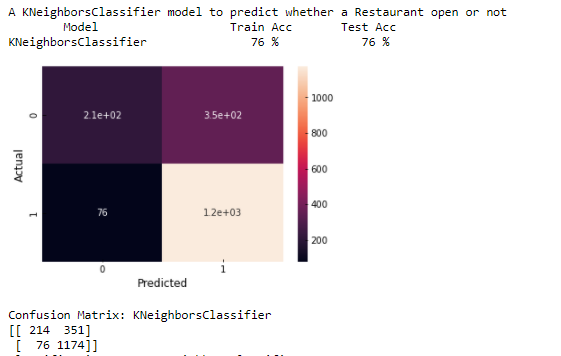


Figure 71: Results of K Neighbor classifier for restaurant status predictions

The above figure is showing the output of KNN Classifier testing. It can be seen that model has wrongly predicted 351 not open status as open. Also, the accuracy becomes 76%. But this is showing KNN Classifier is an average algorithm for this purpose.

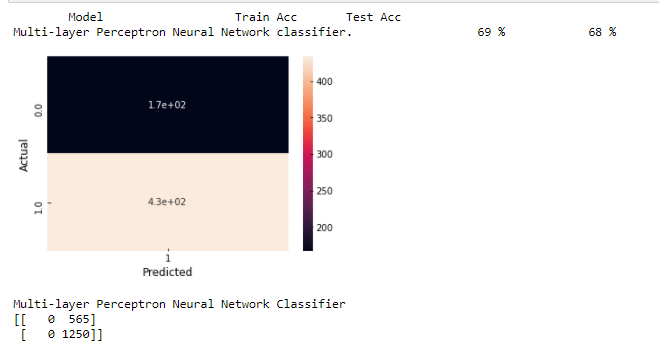


Figure 72: Results MLP classifier for restaurant status predictions

The above figure is showing the output of MLP Classifier testing. It can be seen that model has wrongly predicted 565 not open status as open. Also, the accuracy becomes 69%. This is showing MLP Classifier is a bad algorithm for this purpose.



Figure 73: Restaurants Recommendations accuracy comparisons

The above Figure is showing that random forest classification has achieved the best accuracy for restaurant status Predictions.

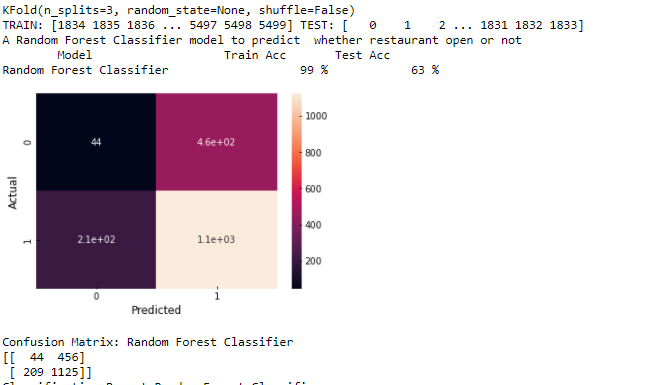


Figure 74: K fold validation step-1 results

K- Fold Validation was applied on Random Forest Classification. But it can be seen the random forest isn’t a sensible for the case where we are predicting based on increasing value.



Figure 75: K Fold validation step 2 results

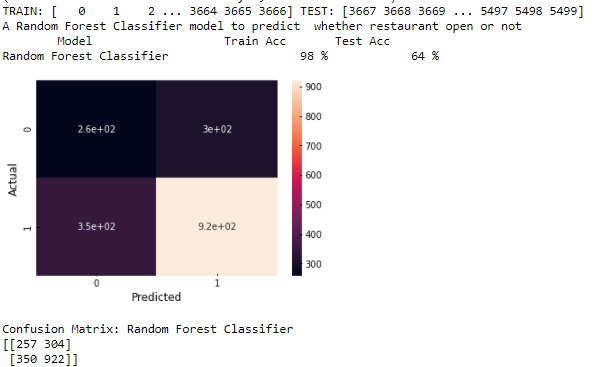


Figure 76: K Fold Validating step 3 results

Figure 75 and 76 are also showing that K Fold Cross Validation technique has proven the random forest is also so much not suitable for the purpose but it is also showing that selecting the training and testing randomly is very important.

### Regression Algorithms Results for Total Sales Prediction

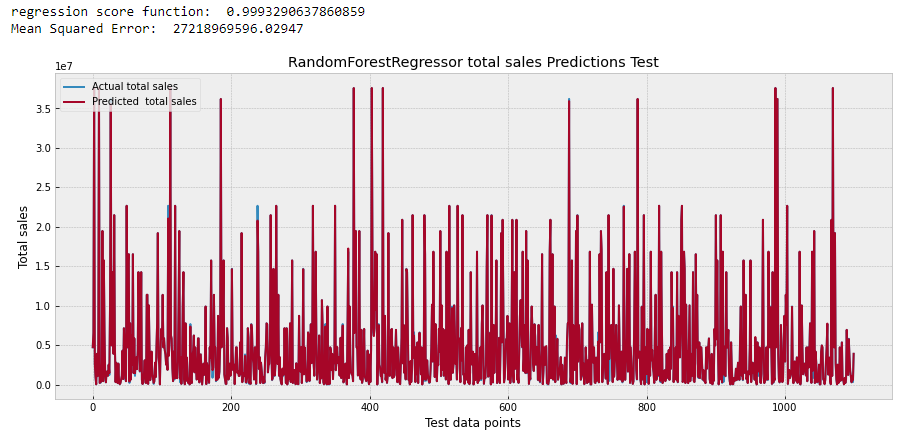


Figure 77: Random Forest Regression total sales predictions original and Predicted results for test data points

The above figure is showing that random forest has mostly high mean square root but it has good regression score function.

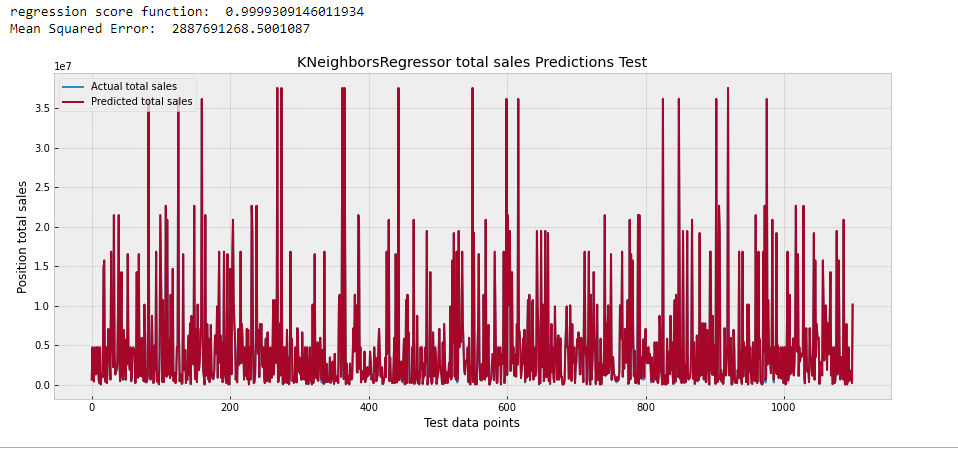


Figure 78: K Neighbor Regression total sales predictions original and Predicted results for test data points

The above figure is showing that K Neighbor regression is also mostly high mean square root but it has good regression score function.

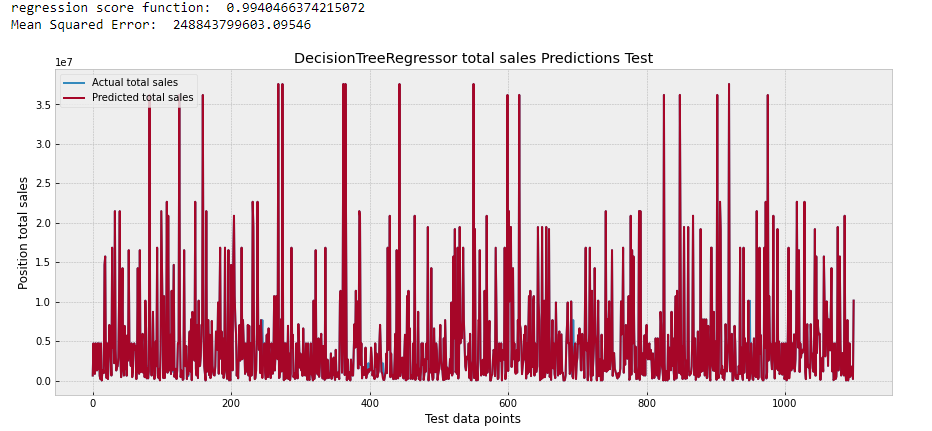


Figure 79: Decision Tree Regression for total sales predictions original and Predicted results for test data points

The above figure is showing that Decision Tree regression is also mostly high mean square root but it has good regression score function.

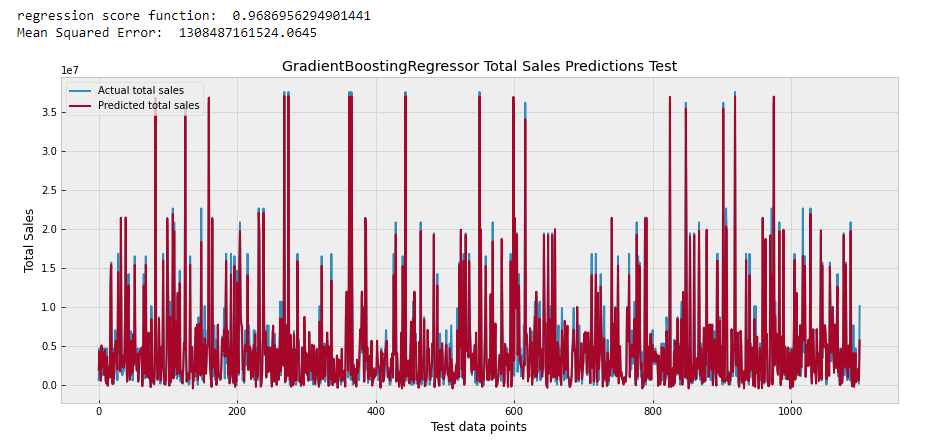


Figure 80: Gradient Boosting Regression for total sales predictions original and Predicted results for test data points

The above figure is showing that Decision Tree regression is also mostly high mean square root and also it does not have a good regression score function.

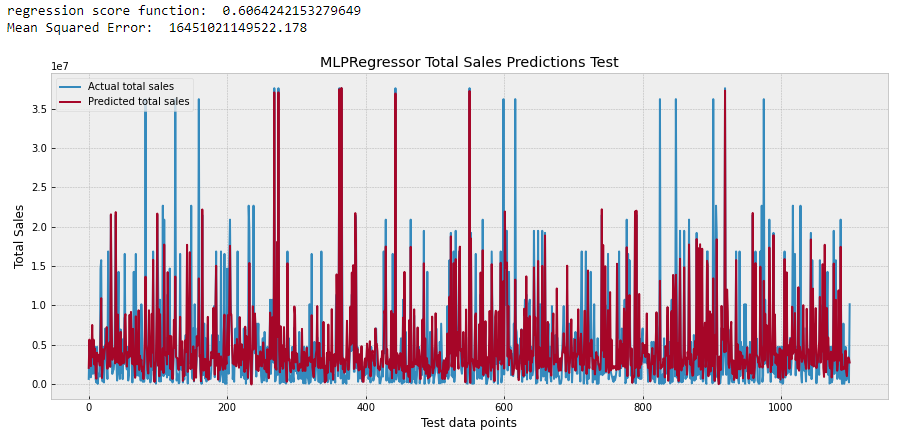


Figure 81: MLP Regression for total sales predictions original and Predicted results for test data points

The above figure is showing that MLP is also mostly high mean square root and also it does not have a good regression score function.

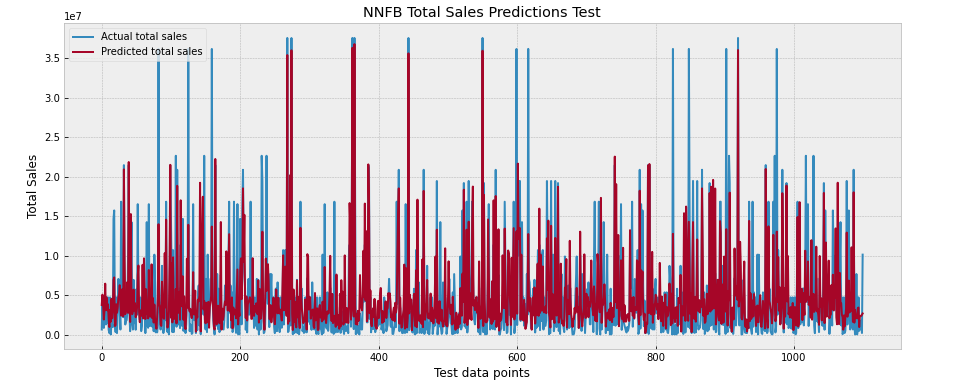


Figure 82: Neural Network Regression for total sales predictions original and Predicted results for test data points

The above figure is showing that Neural Network is also mostly high mean square root and also it does not have a good regression score function.

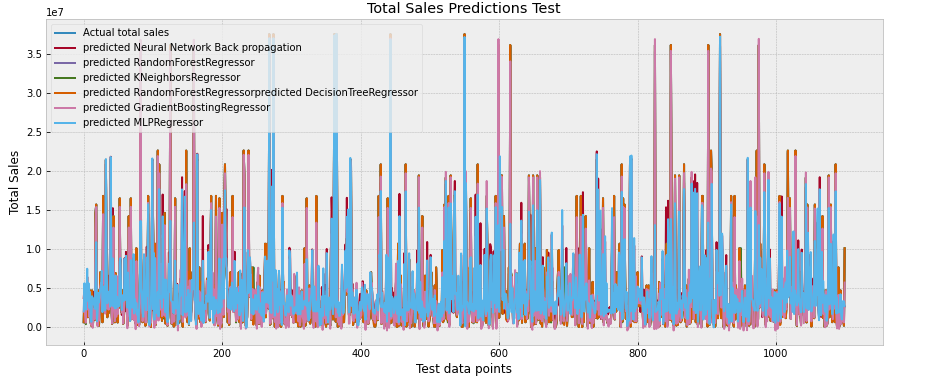


Figure 83: Combined plot for regression results

The combined plot is showing that there are differences in prediction of each algorithm.

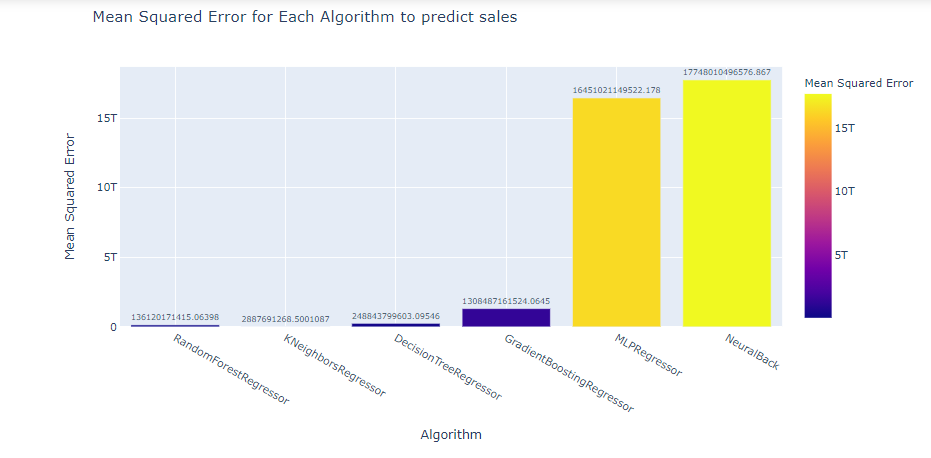


Figure 84: Mean Squared Algorithm for each algorithm to predict sales

The above figure is showing that K Neighbor Regression has the minimum mean squared error.

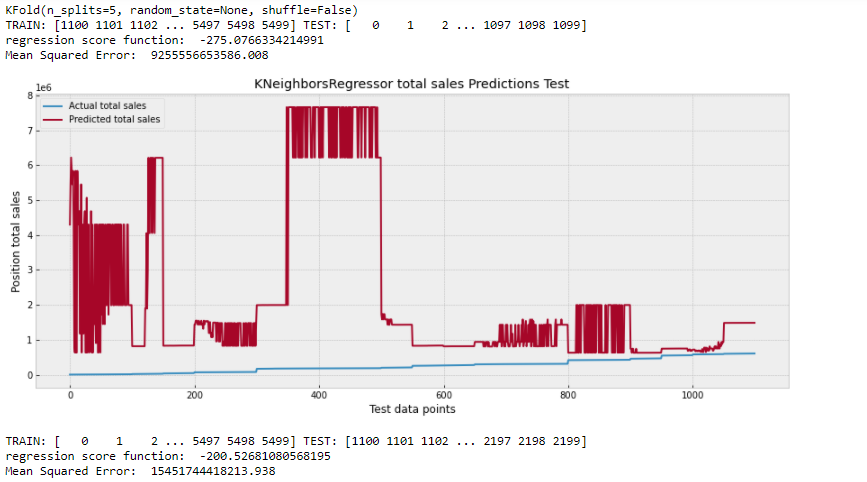


Figure 85: K Fold Validation Split 1 results

Cross Validation using K Neighbor Regression was done.

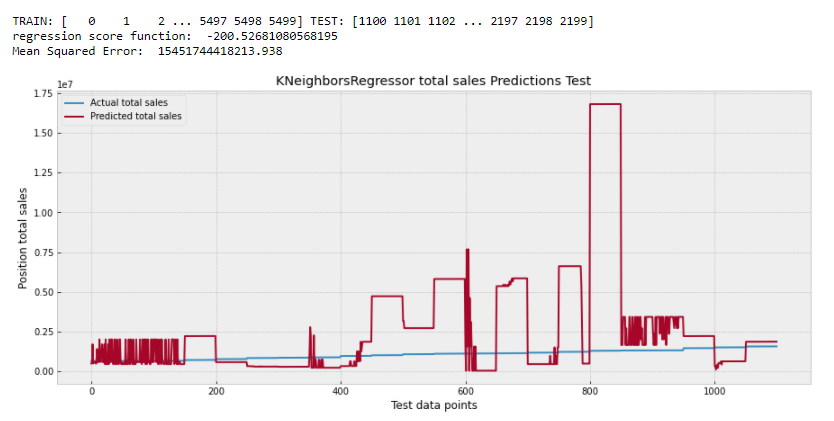


Figure 86: K Fold Validation Split 2 results

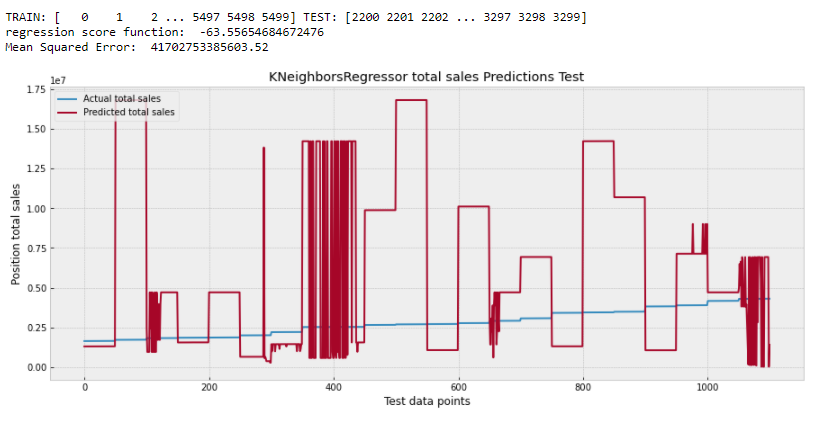


Figure 87: K Fold Validation split 3 results

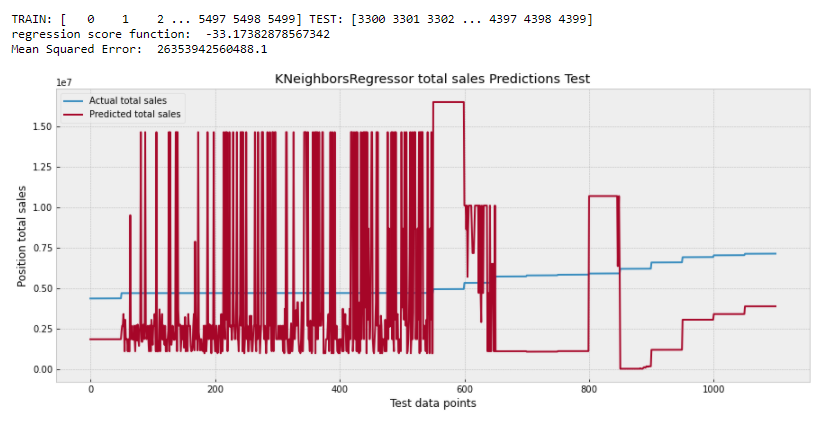


Figure 88: K Fold Validation split 4 results

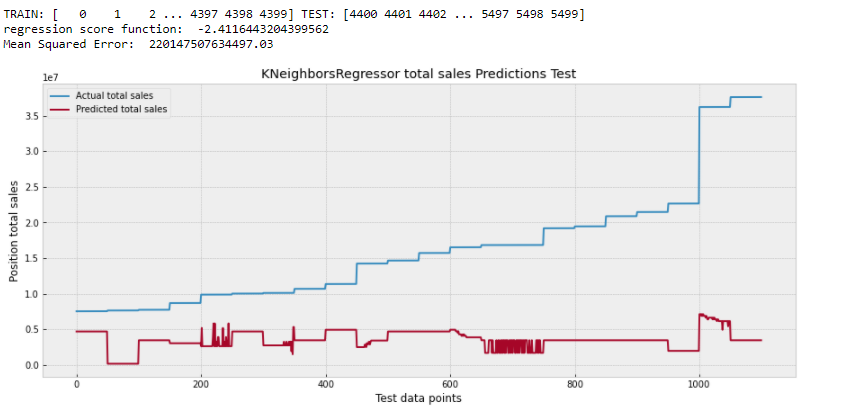


Figure 89: K Fold Validation split 5 results

Figure 85 to 89 are showing that it is difficult to predict the increasing and decreasing the price when we have 1125 continuous values are predicted. It can be seen that continues selection is not reasonable and hence the random selection should be done for the model to perform best.

## **Discussions**

From above results it can be seen that mostly people of age 20 to 24 are available in any District. In any area the age group 25 to 44 is very much active. It can be said that the continues data is difficult to predict and to tune the model random section of rows must be done.

# **CHAPTER 5:**

# **CONCLUSIONS**

## **Key Findings**

From this research it was checked that the data selection to fit in algorithms must be selected randomly for more accuracy. It was checked that Random Forest Classification is the best algorithm for the Restaurant Status Prediction. It was checked that K Neighbor Regression is one of the best algorithm for total sales predictions. It was also checked that mostly people available for working in Hong Kong are from age 20 to 40.

## **Limitations**

When K Fold Cross Validation was applied on the best algorithms for both cases the accuracy was dropped hence the random selection of data should be done to get the accurate results.

The dataset was relatively small then the total number of Restaurants in the Hong Kong hence in future more data is needed.

## **Recommendations and Further Studies**

In future more, data can be added to fit into the algorithms so that they can be more accuracte. In future parameter tuning can also be done to get maximum accuracy for each algorithm.

## **Social, Ethical, Professional and Legal Issues**

Client purposed to develop innovations of technology sides and kept close tracks on restaurants opening and sales and predictions. Though this research adopted hotels data with no personal privacy concerns.

5.4.1 **Social issues**

The insight provided from the predication model is purposed for internal use. The poor performance causes reputation risk and loss business shares to the client.

5.4.2 **Ethical issues**

The prediction model is a kind of tool for recommendation for experience user but still

required some knowledges and not blindly following the prediction. Misunderstanding or

misuses on the model may produce wrong result but some results may be downtrend due to significant information disclosure (profit warning).

5.4.2 **Professional issues**

The model is only applied to Hong Kong Restaurants, not universal model on various restaurants. Misuse on model may produce revenue loss and reputation risk. The model users should be well-known the model usage. The prediction has no guarantee accuracy.

5.4.3 **Legal issues**

Client must not directly provide the prediction result to its customers as the restaurant opening

advices. Wrong prediction may make revenue loss to client’s customers. Client should

be known that restaurant advice should be provided by licensed person and client has to make a clear internal announcement for the usage of prediction model. To conclude, these concerns are mainly related to the use of models. Sufficient instructions, user’s manual and user training can solve these concerns.

## **Putting it Together : Conclusions**

In Final, the research has fulfilled the client’s requirements in chapter 1.3 and Project Objectives in 1.4. The aim of this research is to use ML supervised classifications to predict restaurant’s status prediction and to use regression models to predict restaurant’s sales prediction. This research provided the literature review of related work and demonstrate the complete data product development methodology. Results evaluations and visualizations are also added. Te research shows the Random Forest Classification is best algorithm for restaurant’s status predcition and K Neighbor Regression is the best algorithm for restaurant’s sales predcition.

we can figure out how to approach a given problem. For instance, if we are trying to determine what neighborhood someone lives in, KNN is a pretty good choice, whereas Naive Bayesian Classification makes absolutely no sense. But Naive Bayesian Classification could determine sentiment or some other type of prob‐ ability. The SVM algorithm works well for problems such as finding a hard split between two pieces of data, and it doesn’t suffer from the curse of dimensionality nearly as much. So SVM tends to be good for word problems where there’s a lot of features. Neural networks can solve problems ranging from classifications to driving a car. HMMs can follow musical scores, tag parts of speech, and be used well for other system-like applications.

Clustering is good at grouping data together without any sort of goal. This can be useful for analysis, or just to build a library and store data effectively. Filtering is well suited for overcoming the curse of dimensionality. We saw it used predominantly in Chapter 3 by focusing on important attributes of mushrooms like cap color, smell, and the like.

What we didn’t touch on in the book is that these algorithms are just a starting point. The important thing to realize is that it doesn’t matter what you pick; it is what you are trying to solve that matters. That is why we cross-validate, and measure precision, recall, and accuracy. Testing and checking our work every step of the way guarantees that we at least approach better answers.

What’s Next for me?

This is just the beginning of my journey. The ML field is rapidly growing every single year. I was learning how to build ML product which publicly available for F&B industry to widely use it on Web. The future is bright for ML, and now that I learned this courses are better equipped to learn more about deeper subtopics like reinforcement learning, deep learning, artificial intelligence in general, and more complicated ML algorithms to be able to launch the data product publicly.

# **References**

Camp, C., n.d. *Plotly?.* [Online]   
Available at: https://comparecamp.com/plotly-review-pricing-pros-cons-features/#:~:text=The%20main%20benefits%20of%20Plotly,%2C%20scalability%2C%20and%20total%20customization.

Chang, Chih-Chung, and Chih-Jen Lin., 2001. "Libsvm: a library for support vector machines.

Deanne Larson, Victor Chang, 2016. A Review and Future Direction of Agile, Business Intelligence, Analytics and Data Science. *Elsevier, International Journal of Information Management.*

Educba, n.d. *Matplotlib in Python.* [Online]   
Available at: https://www.educba.com/matplotlib-in-python/

Fabian Pedregosa,Olivier Grisel ,Mathieu Blondel , 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research 12 (2011) 2825-2830.*

Fan, Rong-En, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin., 2008. LIBLINEAR: A library for large linear classification. *Journal of machine learning research 9, no. Aug (2008): 1871-1874..*

Jason Brownlee, August 20, 2020. *Gentle Introduction to the Adam Optimization Algorithm for Deep Learning.* [Online]   
Available at: https://machinelearningmastery.com/adam-optimization-algorithm-for-deep-learning/

Jérémie du Boisberranger, J. V. d. B. ,. L. E. T. J. F., 2007-2020. *scikit-learn Machine Learning in Python.* [Online]   
Available at: https://scikit-learn.org/stable/

Jupyter Notebook Documentation, n.d. Jupyter Notebook Documentation, Release 5.3.1.

K. Jarrod Millman , Michael Aivazis, 2011. Python for Scientists and Engineers. *Computing in Science & Engineering .*

Michael Hanke, Yaroslav O. Halchenko, Per B. Sederberg, Stephen José Hanson, James V. Haxby & Stefan Pollmann , 2009. PyMVPA: a Python Toolbox for Multivariate Pattern Analysis of fMRI Data. *Neuroinformatics volume 7, pages37–53(2009).*

Rickard Adolfsson, Eric Andersson, George Osipov, 2019. Improving sales forecast accuracy for restaurants. *Linköpings universitet.*

Suthaharan, S., 2014. Big Data Classification: Problems and Challenges in Network Intrusion Prediction with.

Tiziano Zito, Niko Wilbert, Laurenz Wiskott and Pietro Berkes, 2008. Modular toolkit for Data Processing (MDP): a Python data processing framework. *Front. Neuroinform., 08 January 2009.*

Tom Schaul, Justin Bayer ,Daan Wierstra,Yi Sun, 2010. PyBrain. *Journal of Machine Learning Research 11 (2010) 743-746.*

Wes McKinney, 2011. pandas: a Foundational Python Library for Data. *pyhpc2011\_submission\_9.*