FinalReport on

Restaurant Revenue Prediction System

(Mini-Project of CSE4007 - Data Analytics)

Submitted to

Dr. Divya Meena Sundaram

School of Computer Science and Engineering

Vellore Institute Of technology - Amaravthi

on

12 october 2021

Submitted by

Name	Reg.No	School
Vignesh Bandla	19BCE7085	SCOPE
Hemanth Kongara	19BCE7549	SCOPE
Balaram Naga Srinivas Borigala	19BCD7113	SCOPE
Prakyath Kommineni	19BCE7224	SCOPE

Table 1: Team-3

Contents

Abstract

Currently, making a decision about when and where to open new restaurant outlets is subjective in nature based on personal judgement and development teams' experience. This subjective data is difficult so We will be predicting the annual revenue of a new restaurant using Machine Learning algorithms like SVM, Linear Regression, Random Forest and with a few more which would help food chains to determine the feasibility of a new outlet

Key Words : Machine Learning; Random Forest; Support Vector Machines(SVM); Restaurant; Revenue; Prediction

1 Introduction

New restaurant outlets incur huge time and capital investments to establish. When the new outlet fails to break even, the site closes within a short time and operating losses are incurred. Finding an algorithmic model to increase the return on investments in new restaurant sites would facilitate businesses to direct their investments in other important business areas, like innovation, and training for new employees.Our supervised learning algorithm will construct complex features using simple features such as opening date for a restaurant, city that the restaurant is in, type of the restaurant (Food Court, Inline, Drive Thru, Mobile), Demographic data (population in any given area, age and gender distribution, development scales), Real estate data (front facade of the location, car park availability), and points of interest including schools, banks. Applying concepts of machine learning such as support vector machines and random

2 Objective

The primary objective of Restaurant Revenue Prediction using Machine Learning is to help restaurants make a more informed and optimal decision about opening new outlets.

One of the biggest features of the proposed application is that it aims to predict the revenue of new outlets of existing restaurant chains. Analytical prediction of data has proven more effective than by human judgement. Further, it can allow analysis and comparison of multiple new sites. Thus human errors can be avoided and operations can be performed faster than previous methods. All in all, this revenue prediction system will compute an accurate forecast of a restaurant outlet's future revenues

3 Literature Review

Author	Year Of Publication	Paper Title	Methodology	Algorithm	Advantage	Disadvantage	Remarks
Khushbu Kumari, Suniti Yadav	2018	Linear Regression Analysis Study	Quantity Rasearch	Linear Regression Algorithm	performs exceptionally well for linearly separable data	It is often quite prone to noise and overfitting	powerful method often used to study the linear relation between variables

Houtao Deng	2013	Guided Random Forest in the RRF Package	ensemble classifier	Random Forest	Flexible to Both Regression and Classification Problem	Requires Much more Resources	uses multiple models of several DTs to obtain a better prediction performance
I. Rish	2001	Emperical study of the naive bayes classifier	classification technique based on Bayes' Theorem	Naive Bayes	It handles both continuous and discrete data	faces the 'zero-frequency problem'	It is highly scalable with the number of predictors and data points.

S.V.M Vish-wanathan; M. Narasimha Murthy	2002	SSVM : Asimple SVM algorithm	linear model for classification and regression problems	Support Vector Machine (SVM)	SVM's are very good when we have no idea on the data.	Choosing a "good" kernel function is not easy	SVM can create the best decision boundary that can segregate n-dimensional space into classes
I.Taunk S.De; S.Verma; A.Swetap	2019	A Brief Review of Nearest Neighbor Algorithm for Learning and Classification	works by finding the distances between a query and all the data points	KNN	High accuracy,Versatile	Require high memory	obust to noisy data if the training set is large enough..

T.Chen, C.Guestrin	2016	XGBoost: A Scalable Tree Boosting System	XGBoost is a decision-tree-based ensemble ML algorithm that uses a gradient boosting framework	XG Boost	uses the power of parallel processing.	does not perform so well on sparse and unstructured data	library is laser focused on computational speed and model performance, as such there are few frills

Y.Ren, L.Zhang,P.NSuganthan	2016	Ensemble Classification and Regression-Recent Developments, Applications and Future Directions	Ensemble methods use multiple models to get better performance	Regression Ensembler	can create lower variance and lower bias	ensembles cannot help unknown differences between sample and population	several models are combined in order to improve the prediction accuracy in learning problems with a numerical target variable

4 Reasearch Gap

- **Linear Regression**

  When we fit a linear regression model to a particular data set, the main problem occurs and where it was less efficient is when there exists **"Non-linearity of the response-predictor relationships"** this makes us that virtually all of the conclusions that we draw from the fit are suspect. In addition, the prediction accuracy of the model can be significantly reduced which can be reduced using **Non-Linear Transformations**

- **Naive Bayes**

  assumption of independent predictor features. Naive Bayes implicitly assumes that all the attributes are mutually independent. In real life, it's almost impossible that we get a set of predictors that are completely independent or one another.

- **Random Forest**

  The main limitation of random forest is that a large number of trees can make the algorithm too slow and ineffective for real-time predictions. In general, these algorithms are fast to train, but quite slow to create predictions once they are trained. A more accurate prediction requires more trees, which results in a slower model. In most real-world applications, the random forest algorithm is fast enough but there can certainly be situations where run-time performance is important and other approaches would be preferred.

- **Support Vetor Machines**

  SVM algorithm is not suitable for large data sets. SVM does not perform very well when the data set has more noise i.e. target classes are overlapping. In cases where the number of features for each data point exceeds the number of training data samples, the SVM will underperform.

- **K Nearest Neighbors**

  There are many limitations in the practical usage of KNN Algoeithm like Accuracy depends on the quality of the data. With large data, the prediction stage might be slow. Sensitive to the scale of the data and irrelevant features. Require high memory – need to store all of the training data and also compuataionally expensive

- **XG Boost**

  XG Boost does not perform so well on sparse and unstructured data. A common thing often forgotten is that Gradient Boosting is very sensitive to outliers since every classifier is forced to fix the errors in the predecessor learners. The overall method is hardly scalable

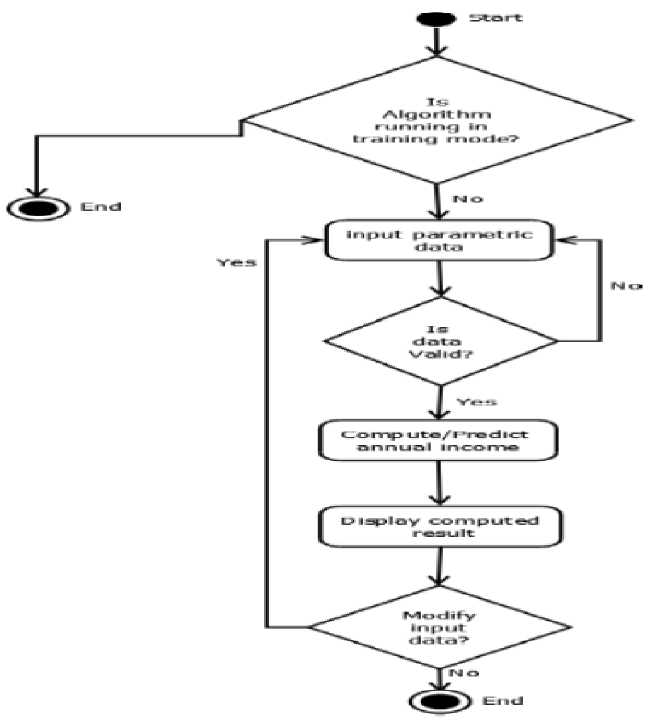
- **Regression Ensemblers**

  ensembles are not always better. New observations can still confuse. That is, ensembles cannot help unknown differences between sample and population. Ensembles should be used carefully.

5 Proposed Methadology

The problem can be defined as: design an automated approach to decide the task environment for new restaurant by applying concepts of Support Vector Machines, Gaussian Naive Bayes and Random Forest on certain parameters, it will predict the annual revenue of a new restaurant outlet which would help food chains to determine its feasibility.

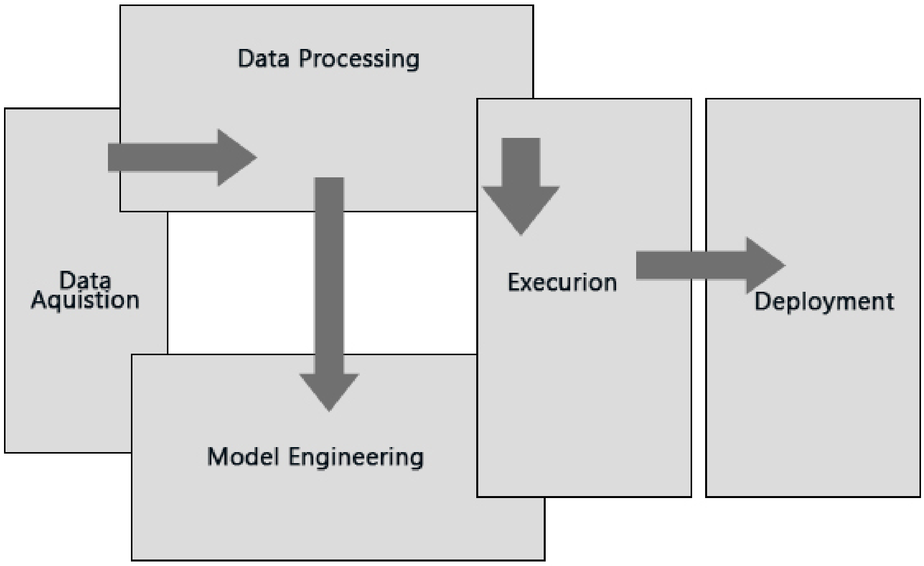
5.1 Flow Chart



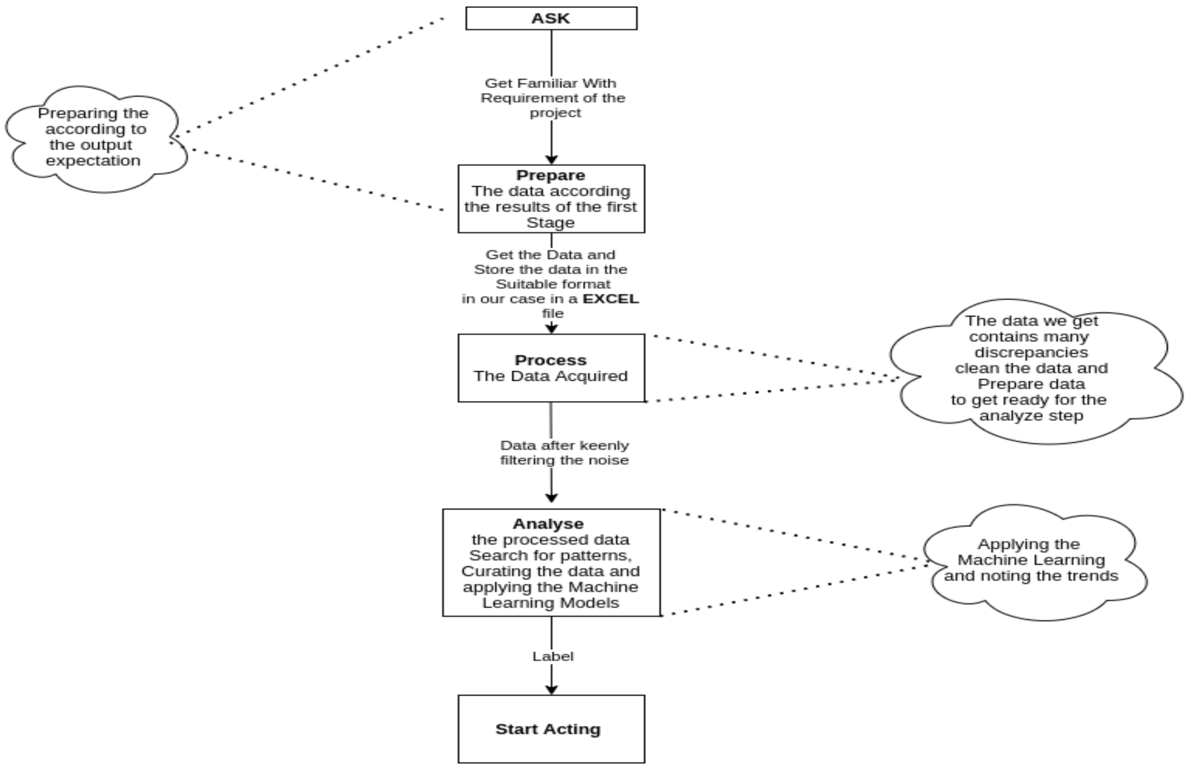
Proposed Idea The system checks if the system is running in training mode. If not, it lets the user input parametric data. The data is checked for validation. In case of invalid data, the user is asked to reenter data. Otherwise our algorithm, computes the annual income and displays the predicted result. Restaurant Revenue Prediction

We here analyze different models and carry out the comparitive study where finally we will sort out the one best performing algorithms

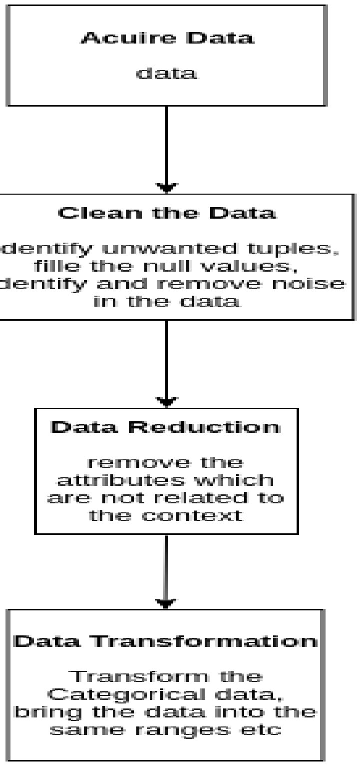
5.2 Architecture Diagram



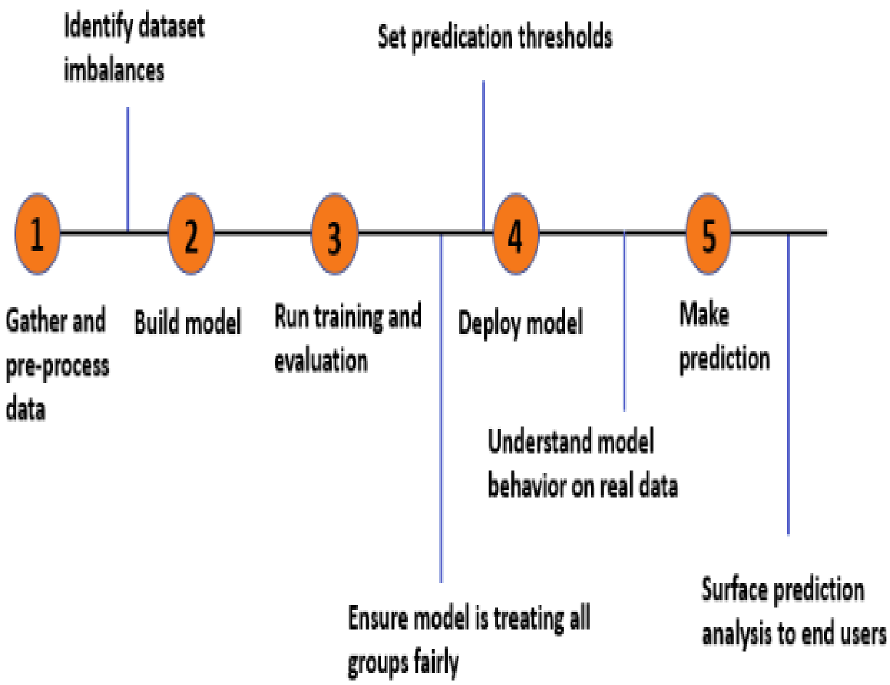
5.3 The Data Life Cycle



5.4 Processing Preparing Stage



5.5 Putting things together



6 Experimental Framework

6.1 Implementation Details

6.1.1 Languages Used

- Python
- R

6.1.2 Linear Regression

Splitting Data set

```
x_train, x_test, y_train, y_test = train_test_split(X,y, test_size=0.2,
random_state=1)
```

Model Fitting

```
from sklearn.linear_model import LinearRegression
from sklearn import metrics
from sklearn.linear_model import Lasso
lm = LinearRegression()
lm.fit(x_train, y_train)
predictions = lm.predict(x_test)
```

6.1.3 Random Forest

Splitting Data set

```
X_train <- panel[1:nrow(train),-1]
X_test <- panel[(nrow(train)+1):nrow(panel),]
```

Model Fitting

```
model_rf_1 <- RandomForestRegression_CV(X_train, result, X_test, cv=5,
ntree=25,nodesize=5,seed=235,metric="rmse")
model_rf_2 <- RandomForestRegression_CV(X_train, result, X_test, cv=5,
ntree=25,nodesize=5,seed=235,metric="rmse")
model_rf_3 <- RandomForestRegression_CV(X_train, result, X_test, cv=5,
```

```
ntree=25,nodesize=5,seed=235,metric="rmse")
model_rf_4 <- RandomForestRegression_CV(X_train,result,X_test,cv=5,
ntree=25,nodesize=5,seed=235,metric="rmse")
model_rf_5 <- RandomForestRegression_CV(X_train,result,X_test,cv=5,
ntree=25,nodesize=5,seed=235,metric="rmse")
```

6.1.4 KNN

Splitting Data set

```
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.20, random_state=118)
```

Model Fitting

```
knn_model = KNeighborsRegressor(n_neighbors=knn_regressor.best_params_["n_
knn_model.fit(X_train, y_train)
y_train_pred = knn_model.predict(X_train)
y_pred = knn_model.predict(X_test)
```

6.1.5 XG Boost

Splitting Data set

```
x_train, x_test, y_train, y_test= train_test_split(x, y,
test_size=0.20, random_state=42)
```

Model Fitting

```
import xgboost as xgb
from xgboost import XGBRegressor
from sklearn.metrics import mean_squared_error

data_dmat= xgb.DMatrix(data= x, label= y)
xg_reg= XGBRegressor(objective ='reg:linear', colsample_bytree = 0.3, learn
                max_depth = 5, alpha = 10, n_estimators = 10)
xg_reg.fit(x_train, y_train)
pred=xg_reg.predict(x_test)
```

6.1.6 Regressor Ensembling

Splitting Data set

```
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.20, random_state=118)
```

Model Fitting

```
rf_model_en = RandomForestRegressor(max_depth=200, max_features=0.4, min_s
                                    min_samples_split=6, n_estimators=30, n_jo
rf_model_en.fit(X_train, y_train)
```

6.2 Dataset Description

6.2.1 Dimension and Null count

```
Training set Dimension :  137 43
Test set Dimension :  100000 42
Null values in train set :  0
Null values in test set :  0
```

6.2.2 Structural Info of Data set

```
'data.frame':   137 obs. of  43 variables:
 $ Id        : int  0 1 2 3 4 5 6 7 8 9 ...
 $ Open.Date : Factor w/ 134 levels "01/03/2014","01/07/2000",..: 60 17 21 11 42 16 94 53 78 114 ...
 $ City      : Factor w/ 34 levels "Adana","Afyonkarahisar",..: 17 4 11 32 15 4 17 17 2 12 ...
 $ City.Group: Factor w/ 2 levels "Big Cities","Other": 1 1 2 2 2 1 1 1 2 2 ...
 $ Type      : Factor w/ 3 levels "DT","FC","IL": 3 2 3 3 3 2 3 3 3 3 ...
 $ P1        : int  4 4 2 6 3 6 2 4 1 6 ...
 $ P2        : num  5 5 4 4.5 4 6 3 5 1 4.5 ...
 $ P3        : num  4 4 2 6 3 4.5 4 4 4 6 ...
 $ P4        : num  4 4 5 6 4 7.5 4 5 4 7.5 ...
 $ P5        : int  2 1 2 4 2 8 1 2 1 6 ...
 $ P6        : int  2 2 3 4 2 10 5 3 2 4 ...
 $ P7        : int  5 5 5 10 5 10 5 5 1 10 ...
 $ P8        : int  4 5 5 8 5 8 5 4 5 10 ...
 $ P9        : int  5 5 5 10 5 8 5 4 5 10 ...
 $ P10       : int  5 5 5 10 5 8 5 4 5 10 ...
 $ P11       : int  3 1 2 8 2 10 2 4 1 2 ...
 $ P12       : int  5 5 5 10 5 8 5 3 5 10 ...
 $ P13       : num  5 5 5 7.5 5 6 5 4 5 7.5 ...
 $ P14       : int  1 0 0 6 2 0 3 0 1 0 ...
 $ P15       : int  2 0 0 4 1 0 4 0 1 0 ...
 $ P16       : int  2 0 0 9 2 0 4 0 2 0 ...
 $ P17       : int  2 0 0 3 1 0 3 0 1 0 ...
 $ P18       : int  4 0 0 12 4 0 4 0 4 0 ...
 $ P19       : int  5 3 1 20 2 5 2 3 1 25 ...
 $ P20       : int  4 2 1 12 2 6 4 5 1 3 ...
 $ P21       : int  1 1 1 6 1 3 1 2 1 3 ...
 $ P22       : int  3 3 1 1 2 1 2 4 1 1 ...
 $ P23       : int  3 2 1 10 1 5 1 2 1 10 ...
 $ P24       : int  1 0 0 2 2 0 5 0 4 0 ...
 $ P25       : int  1 0 0 2 3 0 4 0 4 0 ...
 $ P26       : num  1 0 0 2.5 3 0 4 0 4 0 ...
 $ P27       : num  4 0 0 2.5 5 0 5 0 2 0 ...
```

6.2.3 5 Point Summary OF Data

```
       Id            Open.Date          City           City.Group      Type          P1              P2              P3       
 Min.   :    0   07/10/2013:  645   İstanbul:34087   Big Cities:49272   DT: 2244   Min.   : 1.000   Min.   :1.000   Min.   :0.000  
 1st Qu.:25000   05/06/2013:  635   Ankara  : 8720   Other     :50728   FC:57019   1st Qu.: 2.000   1st Qu.:3.750   1st Qu.:4.000  
 Median :50000   07/04/2011:  635   İzmir   : 6465                      IL:40447   Median : 3.000   Median :5.000   Median :4.000  
 Mean   :50000   09/20/2013:  632   Antalya : 5911                      MB:  290   Mean   : 4.088   Mean   :4.428   Mean   :4.215  
 3rd Qu.:74999   03/05/1996:  631   Kocaeli : 4364                                 3rd Qu.: 4.000   3rd Qu.:5.000   3rd Qu.:4.000  
 Max.   :99999   04/17/2009:  625   Mersin  : 2735                                 Max.   :15.000   Max.   :7.500   Max.   :6.000  
                 (Other)   :96197   (Other) :37718                                                                                 
       P4              P5             P6               P7               P8               P9              P10        
 Min.   :2.000   Min.   :1.00   Min.   : 1.000   Min.   : 1.000   Min.   : 1.000   Min.   : 4.000   Min.   : 4.000  
 1st Qu.:4.000   1st Qu.:1.00   1st Qu.: 2.000   1st Qu.: 5.000   1st Qu.: 4.000   1st Qu.: 4.000   1st Qu.: 5.000  
 Median :4.000   Median :2.00   Median : 2.000   Median : 5.000   Median : 5.000   Median : 5.000   Median : 5.000  
 Mean   :4.396   Mean   :1.99   Mean   : 2.882   Mean   : 5.301   Mean   : 4.931   Mean   : 5.251   Mean   : 5.459  
 3rd Qu.:5.000   3rd Qu.:2.00   3rd Qu.: 4.000   3rd Qu.: 5.000   3rd Qu.: 5.000   3rd Qu.: 5.000   3rd Qu.: 5.000  
 Max.   :7.500   Max.   :6.00   Max.   :10.000   Max.   :10.000   Max.   :10.000   Max.   :10.000   Max.   :10.000  

      P11              P12              P13             P14              P15              P16              P17        
 Min.   : 1.000   Min.   : 2.000   Min.   :3.000   Min.   : 0.00   Min.   : 0.000   Min.   : 0.000   Min.   : 0.000  
 1st Qu.: 2.000   1st Qu.: 4.000   1st Qu.:5.000   1st Qu.: 0.00   1st Qu.: 0.000   1st Qu.: 0.000   1st Qu.: 0.000  
 Median : 3.000   Median : 5.000   Median :5.000   Median : 0.00   Median : 0.000   Median : 0.000   Median : 0.000  
 Mean   : 3.312   Mean   : 5.061   Mean   :5.087   Mean   : 1.28   Mean   : 1.306   Mean   : 1.747   Mean   : 1.157  
 3rd Qu.: 4.000   3rd Qu.: 5.000   3rd Qu.:5.000   3rd Qu.: 2.00   3rd Qu.: 2.000   3rd Qu.: 3.000   3rd Qu.: 2.000  
 Max.   :10.000   Max.   :10.000   Max.   :7.500   Max.   :15.00   Max.   :10.000   Max.   :15.000   Max.   :15.000  

      P18              P19              P20              P21              P22             P23              P24        
 Min.   : 0.000   Min.   : 1.000   Min.   : 1.000   Min.   : 1.000   Min.   :1.00   Min.   : 1.00   Min.   : 0.000  
 1st Qu.: 0.000   1st Qu.: 2.000   1st Qu.: 2.000   1st Qu.: 1.000   1st Qu.:1.00   1st Qu.: 1.00   1st Qu.: 0.000  
 Median : 0.000   Median : 3.000   Median : 4.000   Median : 2.000   Median :2.00   Median : 2.00   Median : 0.000  
 Mean   : 1.708   Mean   : 5.191   Mean   : 4.571   Mean   : 2.542   Mean   :2.43   Mean   : 3.64   Mean   : 1.234  
 3rd Qu.: 4.000   3rd Qu.: 5.000   3rd Qu.: 5.000   3rd Qu.: 3.000   3rd Qu.:3.00   3rd Qu.: 4.00   3rd Qu.: 2.000  
 Max.   :15.000   Max.   :25.000   Max.   :15.000   Max.   :15.000   Max.   :5.00   Max.   :25.00   Max.   :10.000  
```

```
      P25              P26             P27              P28              P29              P30              P31        
 Min.   : 0.000   Min.   : 0.00   Min.   : 0.000   Min.   : 1.000   Min.   : 0.000   Min.   : 0.000   Min.   : 0.000  
 1st Qu.: 0.000   1st Qu.: 0.00   1st Qu.: 0.000   1st Qu.: 2.000   1st Qu.: 2.000   1st Qu.: 0.000   1st Qu.: 0.000  
 Median : 0.000   Median : 0.00   Median : 0.000   Median : 3.000   Median : 3.000   Median : 0.000   Median : 0.000  
 Mean   : 1.244   Mean   : 1.28   Mean   : 1.164   Mean   : 3.234   Mean   : 3.084   Mean   : 2.083   Mean   : 1.193  
 3rd Qu.: 2.000   3rd Qu.: 2.00   3rd Qu.: 2.000   3rd Qu.: 4.000   3rd Qu.: 3.000   3rd Qu.: 3.000   3rd Qu.: 1.000  
 Max.   :10.000   Max.   :12.50   Max.   :12.500   Max.   :12.500   Max.   :10.000   Max.   :25.000   Max.   :15.000  

      P32              P33               P34              P35              P36              P37         
 Min.   : 0.000   Min.   :0.0000   Min.   : 0.000   Min.   : 0.000   Min.   : 0.000   Min.   :0.0000  
 1st Qu.: 0.000   1st Qu.:0.0000   1st Qu.: 0.000   1st Qu.: 0.000   1st Qu.: 0.000   1st Qu.:0.0000  
 Median : 0.000   Median :0.0000   Median : 0.000   Median : 0.000   Median : 0.000   Median :0.0000  
 Mean   : 1.943   Mean   :0.9874   Mean   : 2.109   Mean   : 1.833   Mean   : 1.969   Mean   :0.9735  
 3rd Qu.: 3.000   3rd Qu.:2.0000   3rd Qu.: 3.000   3rd Qu.: 4.000   3rd Qu.: 3.000   3rd Qu.:2.0000  
 Max.   :25.000   Max.   :6.0000   Max.   :30.000   Max.   :15.000   Max.   :20.000   Max.   :8.0000  
```

6.3 Performance Metrics

The Main metric we used here is **Root mean Squared Error**

Root mean square error or root mean square deviation is one of the most commonly used measures for evaluating the quality of predictions RMSE uses and needs true measurements at each predicted data point.

6.4 System Requirements

6.4.1 Hardware Requirements

- Processor with 64-bit, four-core, 2.5 GHz minimum per core (If your dataset size is significantly larger than the medium dataset
- RAM 16GB
- Hard Disk with 80 GB Of Free Space

6.4.2 Software Requirements

Software	Min Version	Recomended
R Software	4.1.2 / 1	4.0.5
R Studio	1.1.0.1-17	2021.09.0
Python Software	Python 1.6	Python 3.7
Jupyter Notebook	Jupyter 4.1.1	Jupyter 4.6

7 Results

S.No	Model	RMSE/Thousand
1	Linear Regression	1886.6
2	Random Forest	1744.92
3	KNN	1819
4	XG Boost	2008.04
5	Regressor Ensembling	1741.68

8 Conclusion

As we wen thorough cleaning process of the small training data and later fitted the model to the test set we can find that The **Root Mean Squared error** for **Regressor Ensembling** gives the minimum which shows us that it becomes a better choice for non-requent usage of the model since it can't find out the differenece between sample and population variable. Hence **Random Forest** gives a better alternative model for prctical frequent applications

References

[1] Graeth James, Daniela Witten, Trevor Hastie, Robert Tibshirani, "An Introduction to Statistical Learning"

[2] Khusbhu Kumari, Suniti Yadav "Linear Regression Anlaysis Study", 2018.

[3] Houtao Deng, "Guided Random Forest in the RRF Package", 2013.

[4] I.Rish, "Emperical study of the naive bayes Classifier", 2001.

[5] M.Vishwanatham, "SVM :AsimpleSVM algorithm", 2002.

[6] I.Tanuk S.De, "A BriefReview of Nearest Neighbor Algorithm for Learning and Classification", 2019.

[7] T.Chen, "XGBoost:A Scalable Tree Boosting System", 2016.

[8] Y.Ren, "Ensemble Classification and Regression Recent Developments,Applications and Future Directions", 2016.