

Comparative Study
For
CE802 Machine Learning and Data Mining

**Assignment: Design and Application of a Machine
Learning System for a Practical Problem**

Professor - Dr Luca Citi
Student Name - Karamjot Singh
Registration number(s): 2006510

Date- 20/01/21

Words count: 1701

Comparative Study

This is an Investigation report of a machine learning application to predict whether a customer will file a claim. The report is divided into 2 sections for different problem statements:

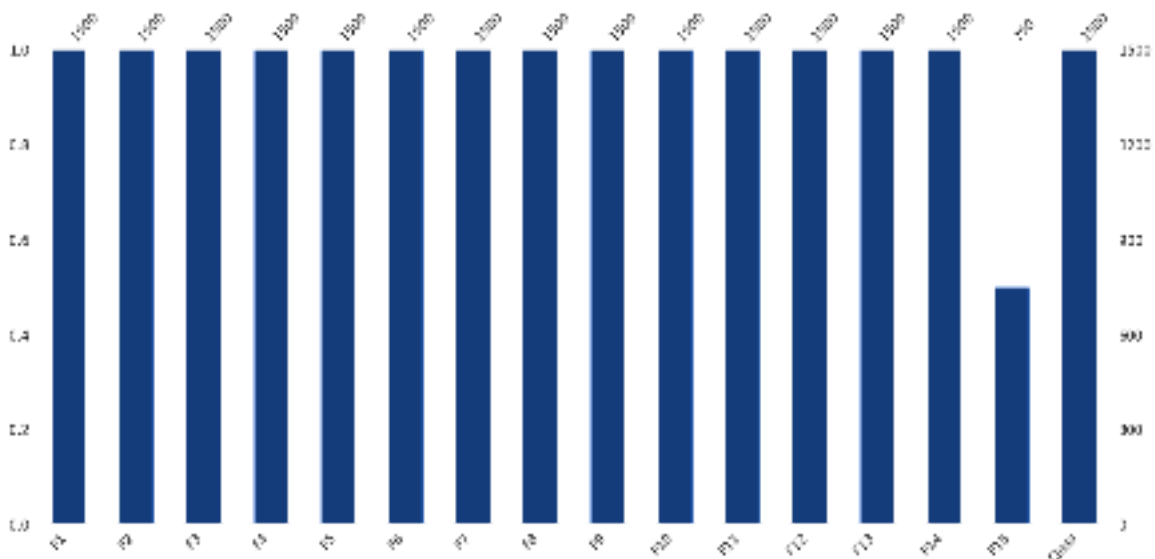
1. Classification where a system is made to predict the target class of the insured.
2. Regression where a system is made to predict the amount of the value of the claim.

Classification

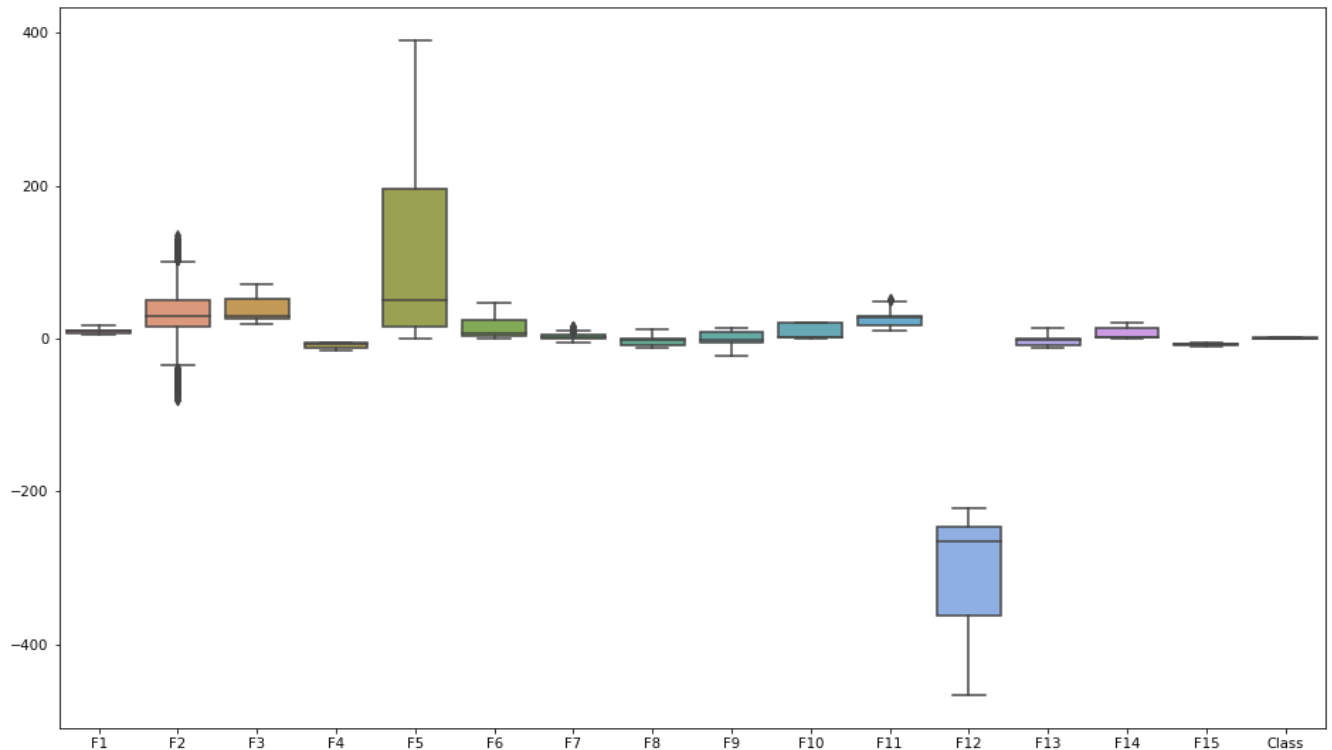
Data Preparation is the most important step in which data is transformed to run it through machine learning algorithms. There are issues with the data such as missing records, outliers, categorical variables.

In this process, data visualization techniques are used to identify the issues.

- Missingno library is used to explore the missing values in the dataset with the help of msno.bar



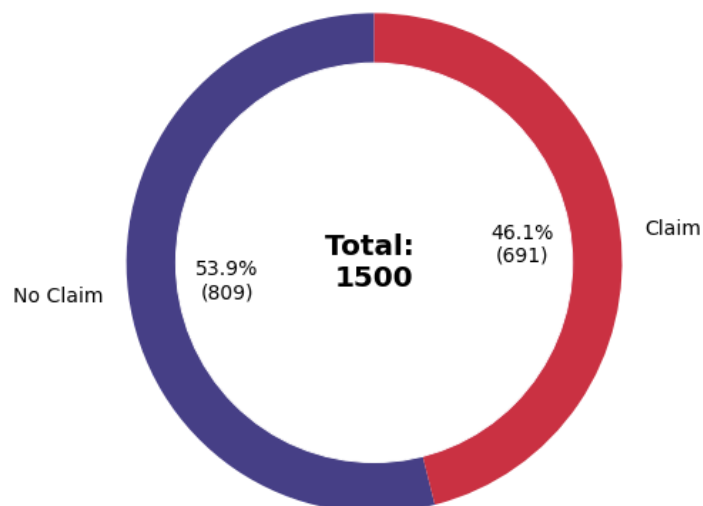
In the above graph, it is clear that half of the values in the F15 column are missing which would be handled by 2 approaches i.e. by dropping the column or Imputing using different techniques.



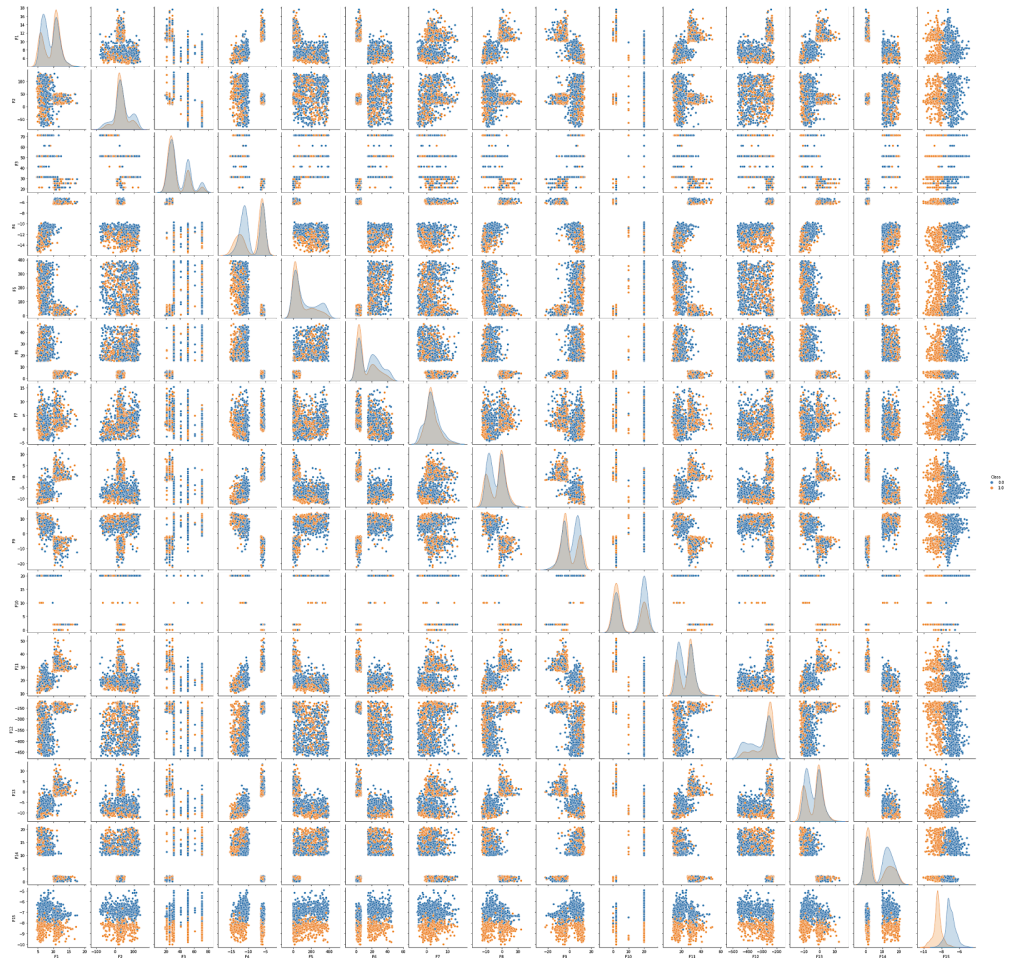
A box plot is created with the Seaborn library, where we can visualize the range of all the features of the data with the occurrence of outliers.

- The donut_chart of the target variable depicts that it contains a balanced distribution of the class. Smote Balancing is used to balance the train data completely before training it on the model.

Total & Percentage of Customers



The pairplot gives information about the type of algorithm which can solve this classification problem like we can see that many points are overlapping and they make a non-linear formation, from which we can say that SVM would work fine on it as compared to Logistic Regression.



Data Normalization

Many algorithms are sensitive to the scale of the data so we normalize the data to bring all the variables to the same range. This also improves the accuracy of the model. Scaling techniques which were used in the code are:

1. MinMax Scaler: it translates each feature individually within a specifically defined range
2. Standard Scaler: it scales the data so that the distribution centred around zero.
3. Quantile Scaler: it transforms the data into Gaussian or uniform probability distribution.

Classifiers

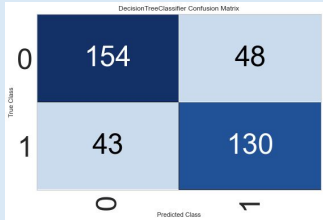
1. **Decision Tree** - It transforms the data into a tree representation. Each internal node of the tree representation denotes an attribute and each leaf node denotes a class label[2].
2. **Support Vector Classifier (SVC)** - the model creates a hyperplane to classify data points with the maximum possible margin. Because the nature of our data i.e. non-linear and SVC uses kernel trick to handle non-linear points, it is giving the highest accuracy.
3. **Random Forest** - consists of a large number of individual decision trees that operate as an ensemble. It is prone to overfitting as compared to decision trees [3]. It was the second-best model in our system after SVM

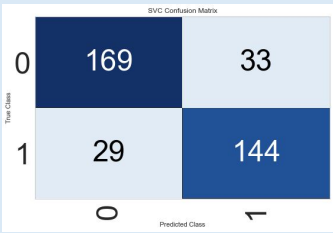
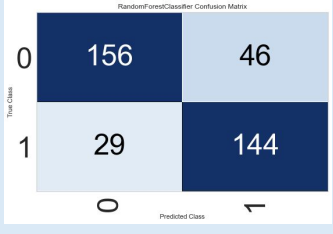
Model Comparison

The basic approach to handle the missing values in the dataset is to drop the F15 column. After that 2 different imputation methods, were applied:

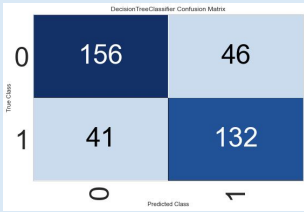
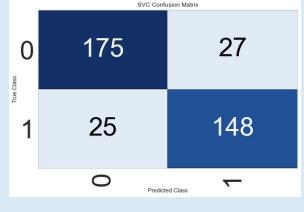
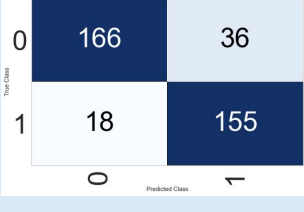
1. KNN Imputation
2. Median Imputation

❑ Dropping F15

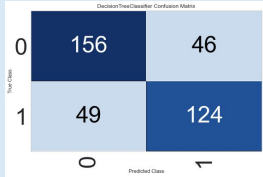
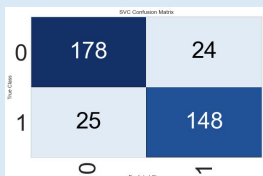
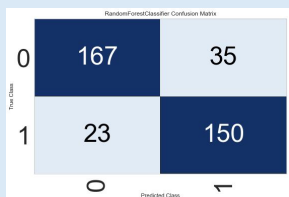
Classifiers	Confusion Matrix	Grid Search Parameters	Accuracy
Decision Tree	 <p>A confusion matrix for a Decision Tree Classifier. The y-axis is labeled 'True Class' with values 0 and 1. The x-axis is labeled 'Predicted Class' with values 0 and 1. The matrix values are: True Class 0, Predicted Class 0: 154; True Class 0, Predicted Class 1: 48; True Class 1, Predicted Class 0: 43; True Class 1, Predicted Class 1: 130.</p>	Max depth - 8 Criterion - 'entropy'	75.8

SVC		C - 1 gamma - 0.1 kernel - poly	83.4
Random Forest		N_estimators - 250	80.2

❑ KNN Imputation

Classifiers	Confusion Matrix	Grid Search Parameters	Accuracy
Decision Tree		Max depth - 8 Criterion - 'entropy'	79.8
SVC		C - 3.5 gamma - 0.6 kernel - rbf	89.4
Random Forest		N_estimators - 500	85.3

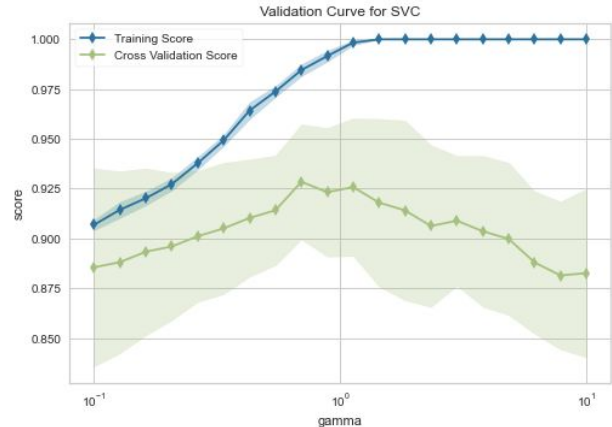
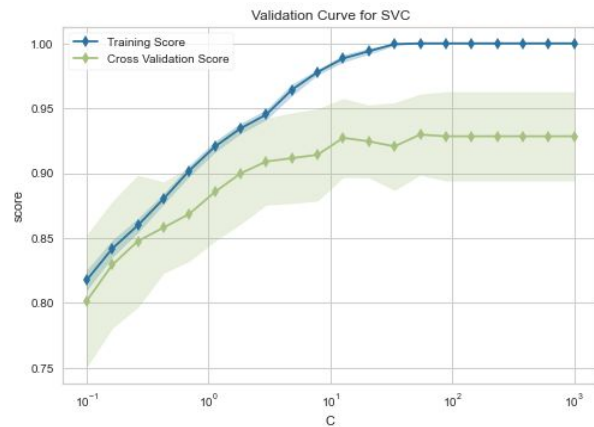
❏ Imputation with Median

Classifiers	Confusion Matrix	Grid Search Parameters	Accuracy												
Decision Tree	 <table><tr><th colspan="2">DecisionTreeClassifier Confusion Matrix</th></tr><tr><th rowspan="2">True Class</th><th>0</th><th>1</th></tr><tr><td>156</td><td>46</td></tr><tr><th rowspan="2">Predicted Class</th><th>0</th><th>1</th></tr><tr><td>49</td><td>124</td></tr></table>	DecisionTreeClassifier Confusion Matrix		True Class	0	1	156	46	Predicted Class	0	1	49	124	Max depth - 8 Criterion - 'entropy'	77.8
DecisionTreeClassifier Confusion Matrix															
True Class	0	1													
	156	46													
Predicted Class	0	1													
	49	124													
SVC	 <table><tr><th colspan="2">SVC Confusion Matrix</th></tr><tr><th rowspan="2">True Class</th><th>0</th><th>1</th></tr><tr><td>178</td><td>24</td></tr><tr><th rowspan="2">Predicted Class</th><th>0</th><th>1</th></tr><tr><td>25</td><td>148</td></tr></table>	SVC Confusion Matrix		True Class	0	1	178	24	Predicted Class	0	1	25	148	C - 21.5 gamma - 0.3 kernel - rbf	86.9
SVC Confusion Matrix															
True Class	0	1													
	178	24													
Predicted Class	0	1													
	25	148													
Random Forest	 <table><tr><th colspan="2">RandomForestClassifier Confusion Matrix</th></tr><tr><th rowspan="2">True Class</th><th>0</th><th>1</th></tr><tr><td>167</td><td>35</td></tr><tr><th rowspan="2">Predicted Class</th><th>0</th><th>1</th></tr><tr><td>23</td><td>150</td></tr></table>	RandomForestClassifier Confusion Matrix		True Class	0	1	167	35	Predicted Class	0	1	23	150	N_estimators - 500	85.5
RandomForestClassifier Confusion Matrix															
True Class	0	1													
	167	35													
Predicted Class	0	1													
	23	150													

Interpretation of the Results

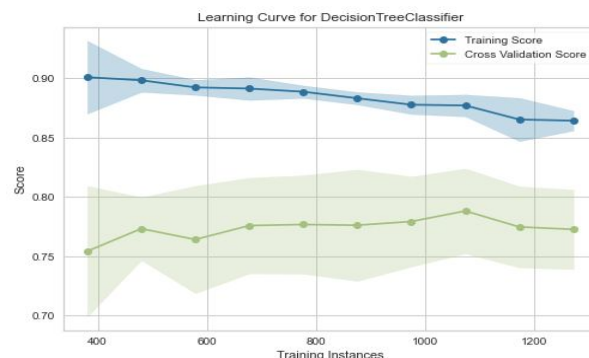
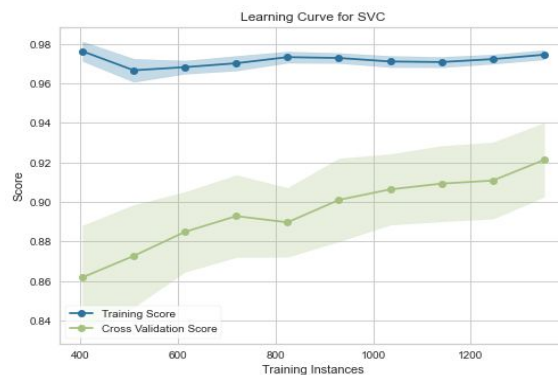
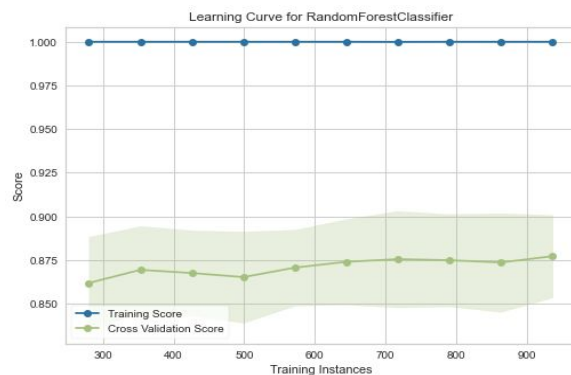
After comparing models with different imputation and scaling techniques, KNN with quantile scaling gives the best results. Quantile transforms the features in such a way that numerical features have a standard probability distribution, such as a uniform distribution on which many algorithms perform better. [6]

KnnImputer is the most effective imputation in our case as there were 750 missing values, it uses Euclidean distance to find nearest neighbours to impute the missing value. The number of neighbours were selected with the help of the following validation curves comparing gamma and C values of SVC. The number of neighbours which were giving the highest validation score was selected.



Learning curves

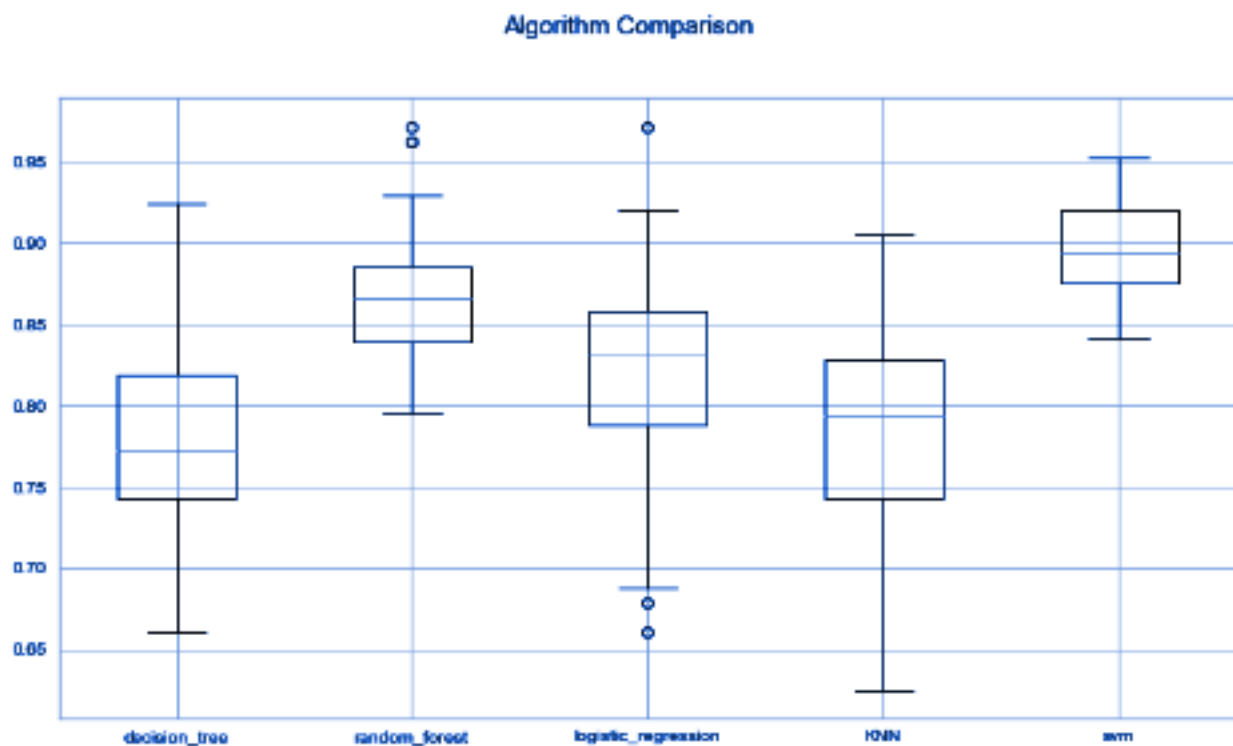
From the following learning curves, we can observe that in SVC, when the training instances increase, the score and validation score curve converges. So adding more data may be useful in this case. As compared to others, SVC is showing the least bias and variance. The gap between curves is more in the Decision tree and Random Forest which says that the model suffers overfitting problem i.e. High Variance. Whereas, Decision Tree is a greedy algorithm which achieves local minima. This means that Decision Tree built is typically locally optimal and not globally optimal or best.[4] That's why decision tree & Random Forest are not good options in this case.



Classifier Selection

From all above Evaluation, the model with the best accuracy is SVC with around 90% accuracy on unseen data with kernel 'rbf'. As shown in the pairplot, the nature of our data is non-linear which SVC can handle easily with the kernel trick. Its main objective is to find a maximum marginal hyperplane that classifies the dataset as perfect as possible.[7]

The final prediction on the test data will be done by using Support Vector Classifier. The following are the Algorithm Comparison, Confusion Matrix and ROC curve of the SVC with Quantile Scalar.



Apart from the above-mentioned classifiers, KNN and logistics regression were also used in the code. Their performance was not as good as others, hence they were not considered for the study but can be found in the code.

Regression

Data Preparation

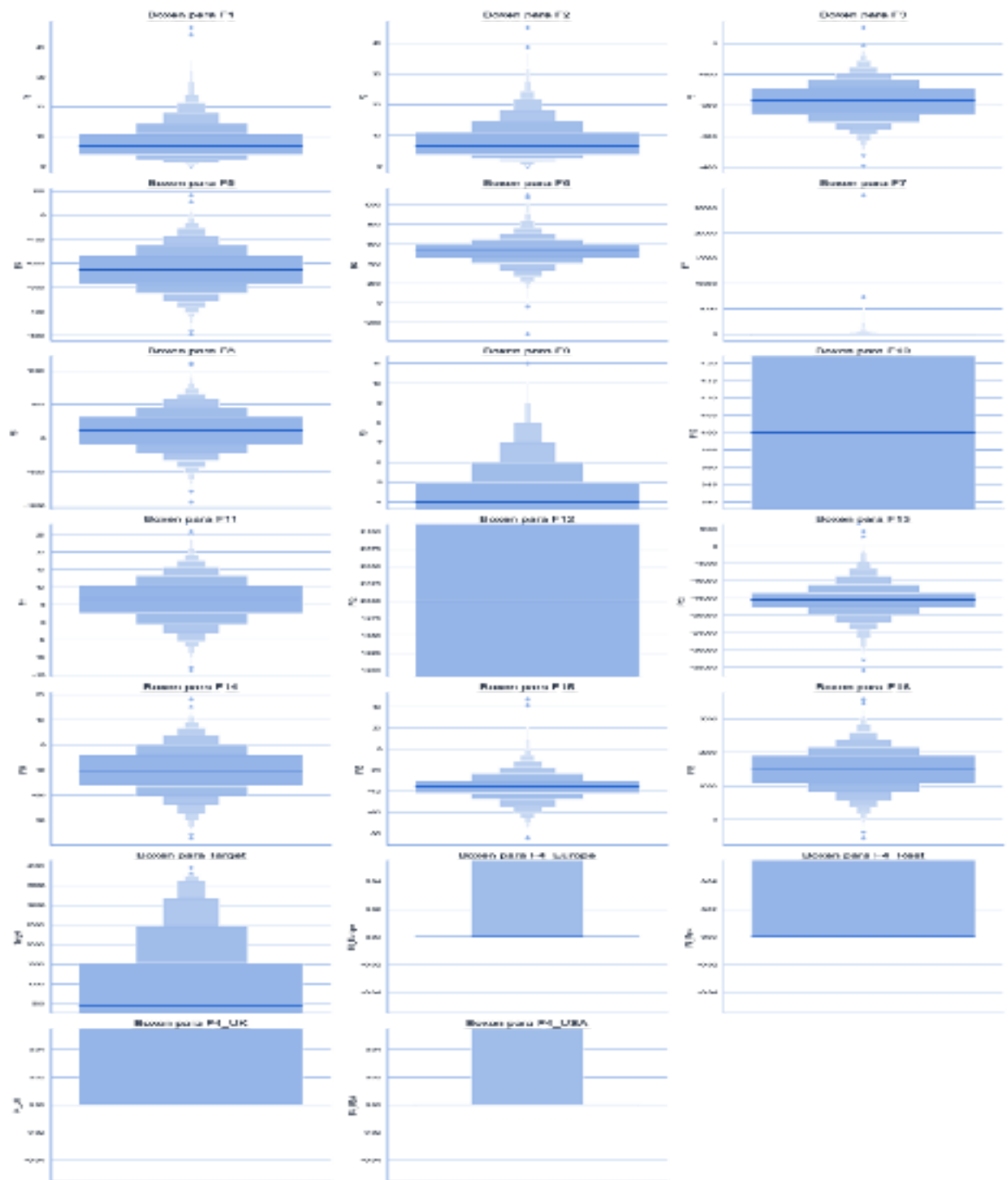
	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F12	F13	F14	F15	F16	Target
0	16.56	12.42	-236.06	Rest	-98.88	529.56	4.54	379.54	1	1	7.30	High	-15085.87	-12.93	-39.42	1734.58	3616.82
1	11.72	12.46	-190.06	Rest	-59.22	493.11	0.05	402.78	5	3	-1.28	Very low	-15782.44	-8.55	-35.61	1672.70	3342.88
2	4.34	2.74	-201.20	UK	-228.48	563.79	1.22	147.35	4	4	8.28	Low	-10526.01	-9.66	-29.10	1462.86	0.00
3	12.76	2.58	-282.26	UK	-173.28	536.94	0.25	113.49	4	3	6.26	Low	-8327.14	-19.23	-34.59	809.46	1742.65
4	11.10	9.82	-242.86	USA	-193.14	617.52	9.15	343.64	8	6	-6.88	Very low	-14434.13	-9.45	-46.14	1435.90	373.56

There are two categorical columns in the data F4 and F12 which need to be changed to numerical values. Different techniques are used to handle different categorical data. As seen there is no relationship between the categories of the F4 column and so we OneHotEncode that column which means making a dummy column for each category as shown in the image. But the categories in F12 have a relationship between so we gave different numbers to them in ascending order:

F4_Europe	F4_Rest	F4_UK	F4_USA
0	1	0	0
0	1	0	0
0	0	1	0
0	0	1	0
0	0	0	1

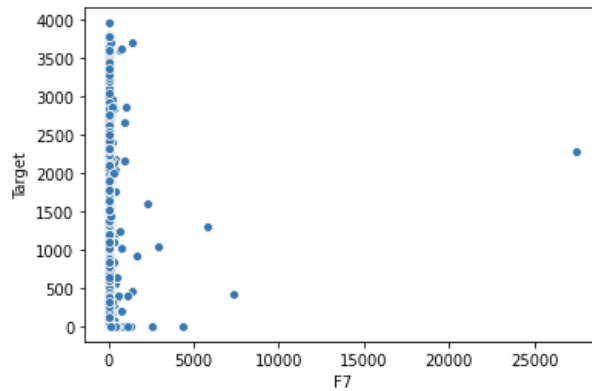
- Very low - 0
- Low - 1
- Medium - 2
- High - 3
- Very High - 4

These multiple boxplots are created with pycomp.viz library which clearly shows the data is highly skewed in F1, F2, F7 and F17 columns which need to be handled before running through algorithms as it can affect the performance of the model. It also shows the occurrence of outliers which would be managed with the outlier function in the code.

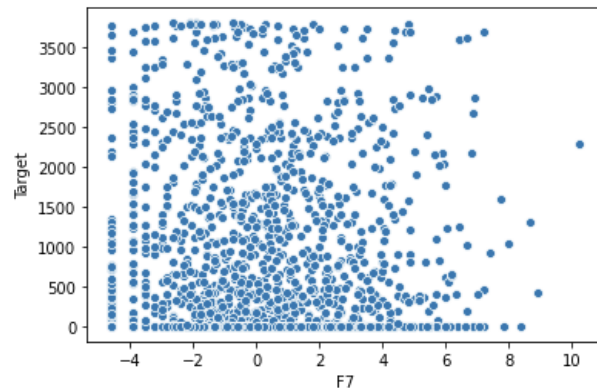


Handling Skewed Data

Log transformation is used in the code to handle the skewed data as it transforms the data into normal distribution which helps to find patterns in the data, for example, these are the scatterplot of F7 columns with Target before and after the transformation.



Before

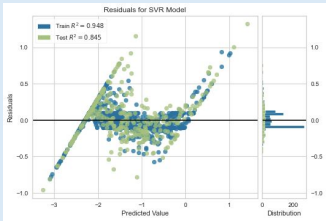
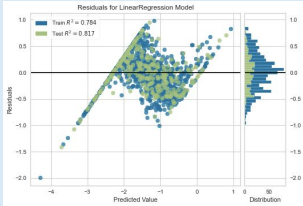


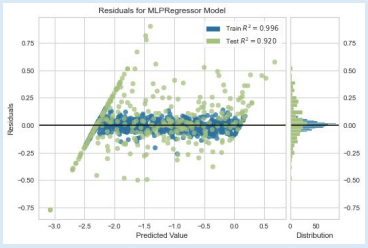
After log transform

Models Evaluation

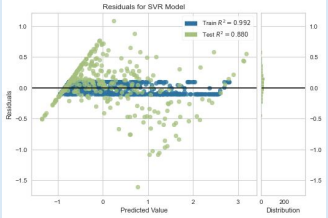
For the study, 3 different regressors are used with the following scaling techniques:

MinMax Scaler

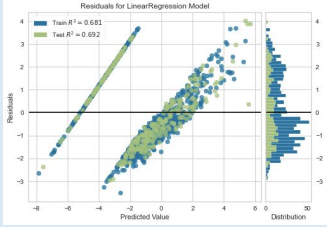
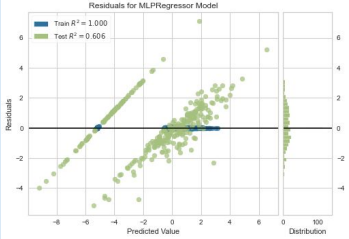
Regressor	Residual Plot	Test Score	RMSE
SVR		88	0.309
Linear Regression		83.3	0.351733

MLP		91	0.232803
-----	---	----	----------

Standard Scalar

Regressor	Residual Plot	Test Score	RMSE
SVR		0.89	0.2797
Linear Regression		0.76	0.451733
MLP		0.99	0.09946

Quantile Scalar

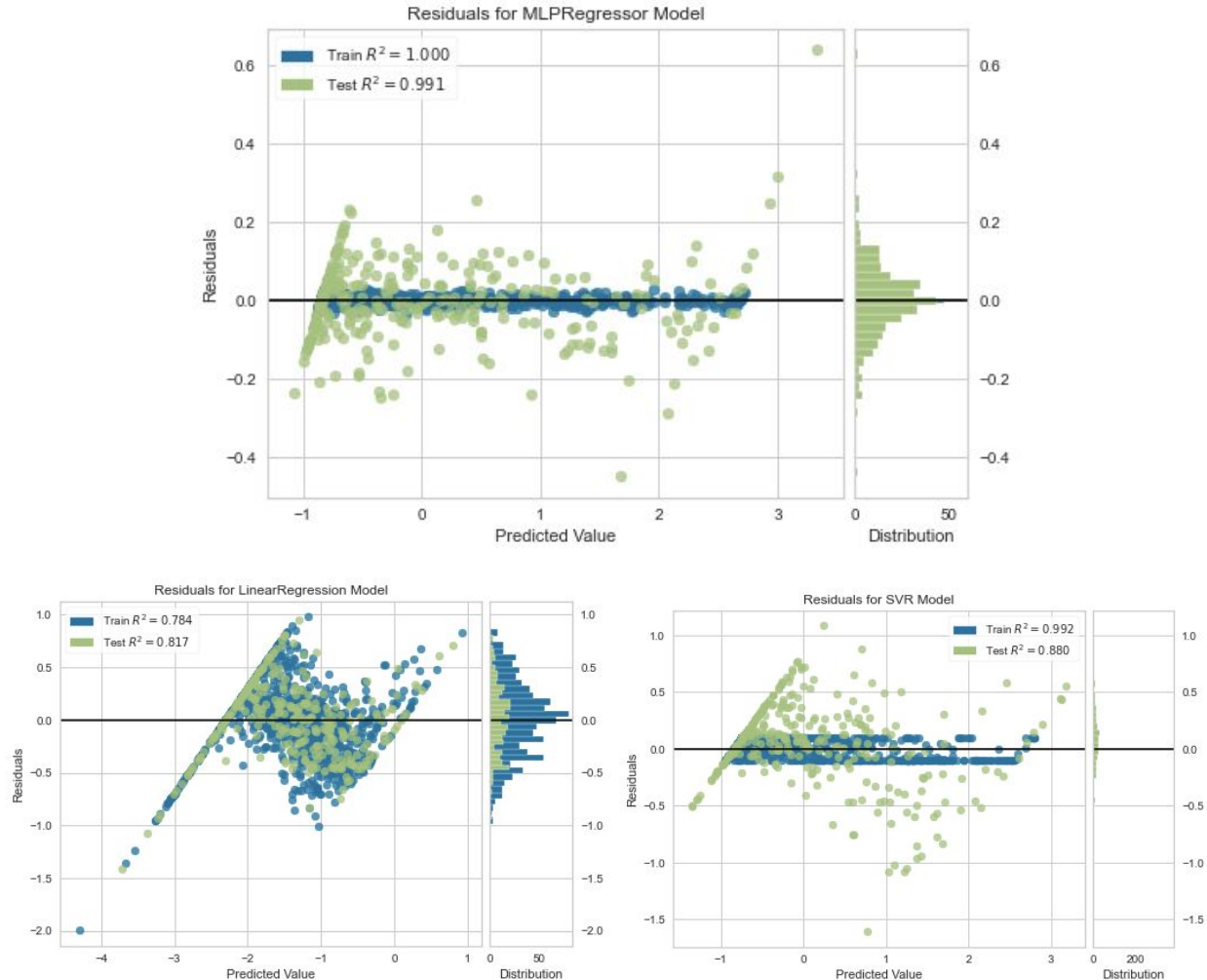
Regressor	Residual Plot	Test Score	RMSE
SVR	 The residual plot for the SVR model shows a clear linear pattern in the residuals, indicating a poor fit. The training R-squared is 0.816 and the test R-squared is 0.679. The residuals are not randomly distributed around the zero line.	0.65	1.5
Linear Regression	 The residual plot for the Linear Regression model shows a clear linear pattern in the residuals, indicating a poor fit. The training R-squared is 0.581 and the test R-squared is 0.692. The residuals are not randomly distributed around the zero line.	0.7	1.351
MLP	 The residual plot for the MLPRegressor model shows a random distribution of residuals around the zero line, indicating a good fit. The training R-squared is 1.000 and the test R-squared is 0.666. The residuals are well-distributed around the zero line.	60.5	1.67

Interpretation of the Results

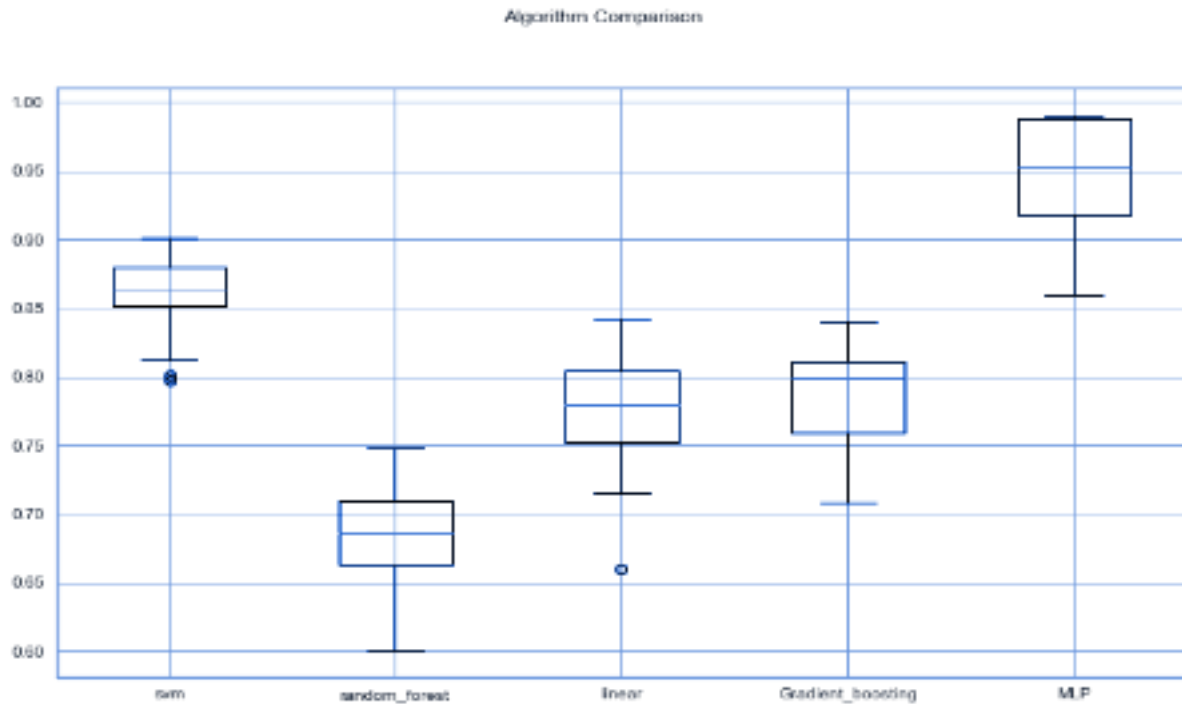
$$\text{Residual} = \text{Actual} - \text{Predicted}$$

1. SVR: In the following residual plots, the points show a linear pattern in the starting but after that distributes near the horizontal line which makes it a fine predictor. But compared to MLP, train and test points are not aligned and due to linear pattern, the rmse values are high.
2. MLPRegressor: An ideal Residual Plot is distributed towards the horizontal line and randomly, where residuals are zero. It's clear from the plot that MLP with Standard Scalar is a well-fitted model.

- Linear Regression: the residual plot also shows the linear pattern which makes it an unfit model for the data. The rmse values are also high in the above scalar table because of the linear residual pattern and skewed distribution of points.



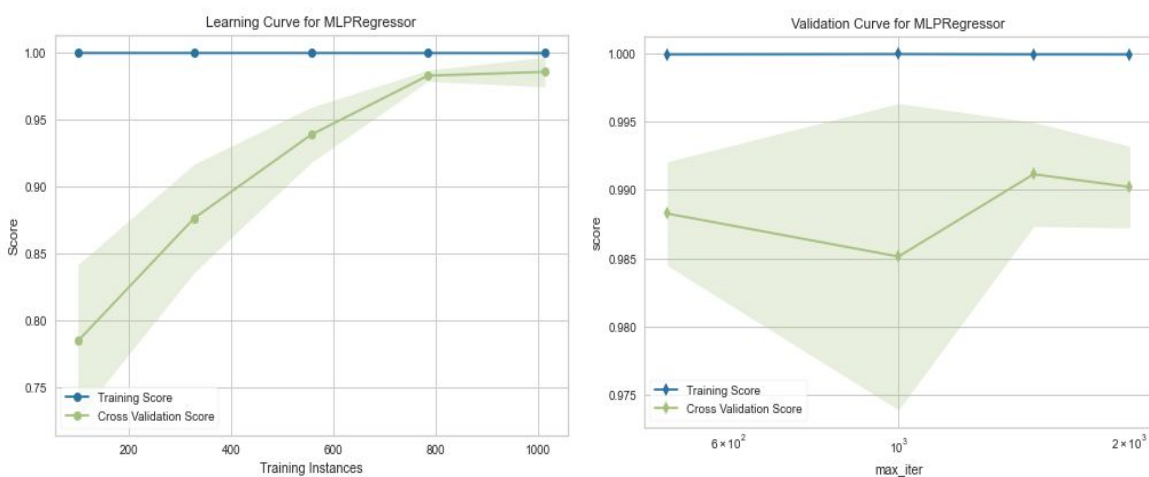
MLP was giving the highest accuracy near 96%. SVR is the second-best regressor with a mean score of 86%. MLP and SVM both are good in distinguishing non-linear dataset. Due to the architecture of the MLP, it can handle multidimensional dataset and uses the Back-propagation and loss function, which trains the model for a specified number of iterations to learn complex relationships between data. The following boxplot shows the performance of all the regressors used on the data with different scaling techniques.



Apart from the above-mentioned regressors, random forest and gradient boosting was also used in the code. Their performance was not as good as others, hence they were not considered for the study but can be found in the code.

MLP Parameter Selection

The following learning curve shows high variability until 600 instances which say the model suffers from variance but with the increase in data both the curve converges



and variance decreases. So adding more data may be useful. From 800-1000 instances, both variance and bias stabilized which makes it the best fit. Whereas in the validation curve, with the max iter 1500 we are getting the highest score. Above evaluations from boxplot, residual plot, learning and validation curves conclude that MLPRegressor is the best model for our data.

Predictions

After the prediction of the test data, there are some negative values in the predicted Target column. The following is the description of the same column.

```
count    1500.000000
mean      901.515089
std       1039.591503
min       -250.960427
25%        49.626754
50%       490.534852
75%      1557.201493
max       4167.824085
Name: Target, dtype: float64
```

This Target represents the discount to the customers and discount cannot be in negative. To reduce the error of the model prediction, we need to change all the negative values to zero which represents NO discount.

Experiments & Learnings

Iterative Imputer is an excellent imputer as it takes all other columns in account before imputing the missing values but I didn't realize that we should not give Target column to the imputer as the iterative imputer will highly correlate the target column and during predictions, it performs terribly poor as there is no Target column to correlate.

Works Cited

1. Brownlee, Jason. 2020. "Train-Test Split For Evaluating Machine Learning Algorithms". <https://machinelearningmastery.com/train-test-split-for-evaluating-machine-learning-algorithms/>.
2. *Decision Tree Algorithm* (2020). Available at: <https://dhirajkumarblog.medium.com/top-5-advantages-and-disadvantages-of-decision-tree-algorithm-428ebd199d9a> (Accessed: 11 January 2021).
3. *Understanding Random Forest* (2019). Available at: <https://towardsdatascience.com/understanding-random-forest-58381e0602d2> (Accessed: 11 January 2021).
4. *Decision Trees – Disadvantages & methods to overcome them* (2015). Available at: <https://www.edupristine.com/blog/decision-trees-development-and-scoring#:~:text=Ap%20from%20overfitting%2C%20Decision%20Trees,are%20prone%20to%20sampling%20errors.> (Accessed: 16 January 2021).
5. Shamsaddini, A., Dadkhah, K. and Gillevet, P. (2020) "BiomMiner: An advanced exploratory microbiome analysis and visualization pipeline", PLOS ONE, 15(6), p. e0234860. doi: 10.1371/journal.pone.0234860.
6. Brownlee, J. (2020) *How to Use Quantile Transforms for Machine Learning, Machine Learning Mastery*. Available at: <https://machinelearningmastery.com/quantile-transforms-for-machine-learning/> (Accessed: 20 January 2021).
7. Ala'raj, M., Majdalawieh, M., & Abbod, M. (2020). Improving binary classification using filtering based on k-NN proximity graphs. *Journal Of Big Data*, 7(1). doi: 10.1186/s40537-020-00297-7