University of Essex
School of Computer Science and Electronic Engineering
CE802 Machine Learning and Data Mining

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

Submitted By:
Gurleen Singh Oberoi
2007161

Submitted To:
Dr. Luca Citi

(1200 words)

Abstract

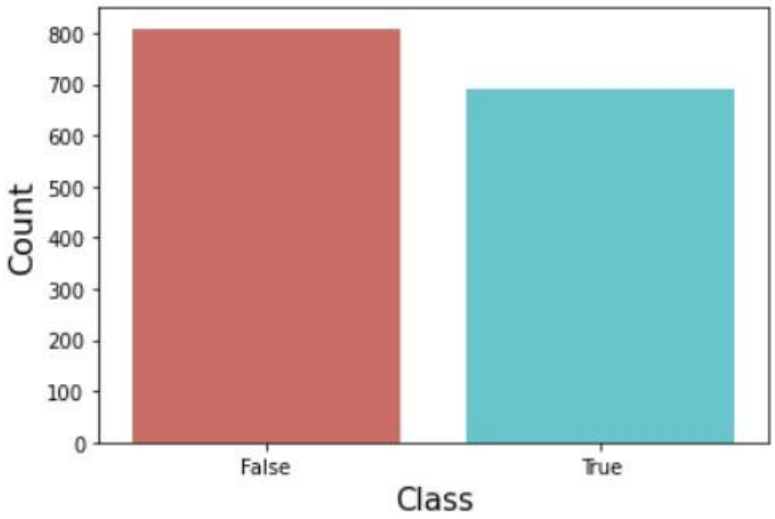
Assignment asked to perform a comparative study of different machine learning algorithms and test the algorithm with test data provided. Learning outcomes of this assignment are: a) to learn to identify machine learning techniques appropriate for a particular practical problem; and b) to undertake a comparative evaluation of several machine learning procedures when applied to specific problem. After performing the required tasks on the given dataset, herein lies my final report.

DATA ANALYSIS

	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F12	F13	F14	F15	Class
0	11.7	4.02	-4.34	9.90	29.79	89.58	0.63	23	10.35	158.56	-7.88	0.03	1	1.96	NaN	False
1	11.7	4.20	-3.68	10.98	17.46	179.58	0.05	11	8.30	110.56	-3.10	0.84	1	1.50	NaN	False
2	37.7	25.80	3.60	0.48	12.24	407.58	-0.29	230	4.06	254.56	6.68	21.60	10	7.63	NaN	True
3	7.7	5.40	0.30	9.42	19.86	119.58	0.29	12	7.61	66.56	-1.84	1.05	1	2.27	12.17	True
4	15.7	5.58	-2.58	16.34	17.49	146.58	-0.64	25	9.86	106.56	-4.36	1.68	1	1.28	NaN	False
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
1495	37.7	33.90	5.80	6.62	10.71	362.58	-1.52	165	5.52	444.56	-1.96	15.30	10	6.93	8.76	False
1496	17.7	29.40	8.00	-0.48	3.54	-102.42	1.17	100	3.76	304.56	6.78	29.25	10	7.53	12.19	True
1497	11.7	2.13	-0.92	12.12	22.65	95.58	-0.57	10	8.47	76.56	-4.76	2.34	1	1.89	NaN	True
1498	11.7	2.94	0.64	11.68	17.49	146.58	1.47	20	8.57	116.56	-5.00	2.67	1	1.48	11.55	False
1499	27.7	30.75	7.76	1.84	8.67	137.58	-2.02	80	4.04	304.56	3.90	20.40	10	6.93	10.41	False

1500 rows × 16 columns

Given data contains 1500 instances and 15 features and target variable Class.



First I counted how many True and False labels are in column Class. It is clear from Bar graph that True labels are less than False labels. Exactly True labels are 691 and False labels are 809.
From this observation we can conclude that there is need of using **Stratification** while splitting data into test and train sets. Stratification will ensure that the train and test sets have approximately the same proportion of samples of each target class.

DATA PREPROCESSING

Splitting the data:

It is not good practice to evaluate the model on the same data on which it was trained. It does not give good indication on how well the model will perform on unseen data. So data is divided into test and train data using stratification.

TRAIN DATA	80%
TEST DATA	20%

Missing Values:

Column F15 has some missing values. Exact number of missing values is 750. Missing data reduces performance of ML models. Not all ML models are sensitive to missing data such as KNN. KNN while calculating distance can ignore the missing values. There are many methods to deal with missing values such as discarding the feature, imputing mean or mode or median value, assigning zero value etc.

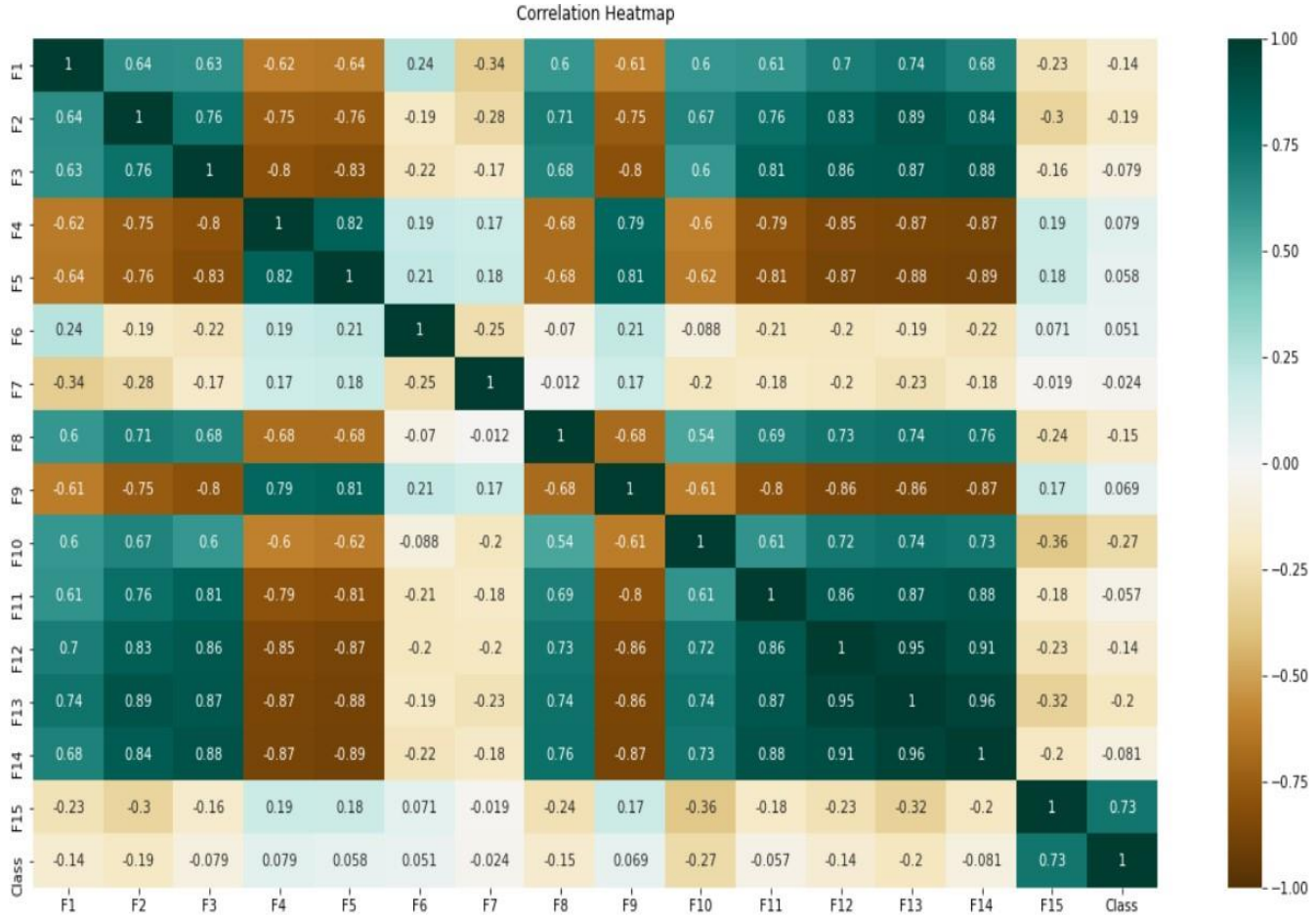
Here I imputed 0, mean, most_frequent and median value in place of missing values and then compared predicting accuracy of the model on test set. Below is the summary :-

ML MODELS / IMPUTATION	SVM	KNN	D.TREE
MOST FREQUENT	87%	75%	77%
ZERO	88%	75%	77%
MEAN	88%	75%	79%
MEDIAN	88%	78%	82%

Accuracy is calculated on test set using tuned parameters of each model with each imputation method.
From above observations I decided to go with median.

Correlation between Features:

Here is heatmap of correlation between Features:



Here I have used Pearson's correlation coefficient. It basically shows the linear relationship between two variables. +1, -1, 0 value shows positive correlation, negative correlation and zero correlation respectively. Correlation is used to understand relationship between features. If two features are highly correlated, then one can be dropped thus reducing the number of features will increase the

computation speed of ML model. Then I tried removing highly correlated features by setting different threshold value and observed the accuracy of ML models.

Accuracy is calculated on test set using imputation method median and tuned parameters of each model.

ML MODELS / THRESHOLD VALUE	FEATURES ELIMINATED	SVM	KNN	D.TREE
0.9	F11,F9,F14,F4,F13,F12 F5	76 %	76 %	75%
0.8	F11,F9,F14,F4,F13,F12 F5,F3	74 %	76%	74 %
	NO FEATURES ELIMINATED	88%	78%	82%

From above table I concluded that I will go with all features. Removing highly correlated features doesn't have a significant effect on KNN model and D.tree. KNN and D.tree treats all features equally. So one will only drop features if there is speed or storage issues. SVM model accuracy significantly dropped after removing highly correlated features.

IMPLEMENTATION OF DECISION TREE CLASSIFIER

To implement Decision Tree Classifier I created a pipeline. Pipeline assembles several steps of ML model which can be cross-validated together while setting different parameters(scikit-learn.org/sklearn.pipeline.Pipeline). Pipeline consist of many steps such as standardising data, tuning of hyperparameters , cross-validation ,implementing classifier. ML pipeline run repeatedly to improve the accuracy of the model and achieve best algorithm.
There is no such need to scale the data before feeding to DT classifier but here I have scaled the data using StandardScaler.

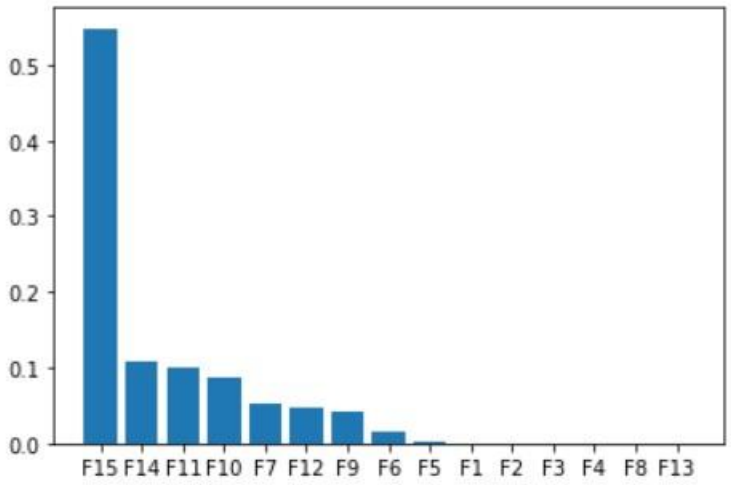
Tuning of parameters:

There are many parameters which can be tuned in a DT classifier but here I'm tuning two parameters that is **criterion** and **max_depth.**
In criterion Gini and Entropy are function to measure the quality of a split. The frequency of agreement/disagreement of the Gini Index and the Entropy is only 2% of all cases. (Laura Elena Raileanu and Kilian Stoffel, "Theoretical comparison between the Gini Index and Information Gain criteria"). Entropy is slow than Gini as in Entropy logarithm is calculated.
Max depth controls the depth of the tree. By increasing depth of the tree, tree will grasp more information about data and model complexity will increase and it will lead to overfitting. Very low depth of tree will cause underfitting.
Grid search cv is used to find optimal parameters which led to highest accuracy of the model. It uses cross-validation and uses all combination of passed values of parameters .

Final Decision Tree:

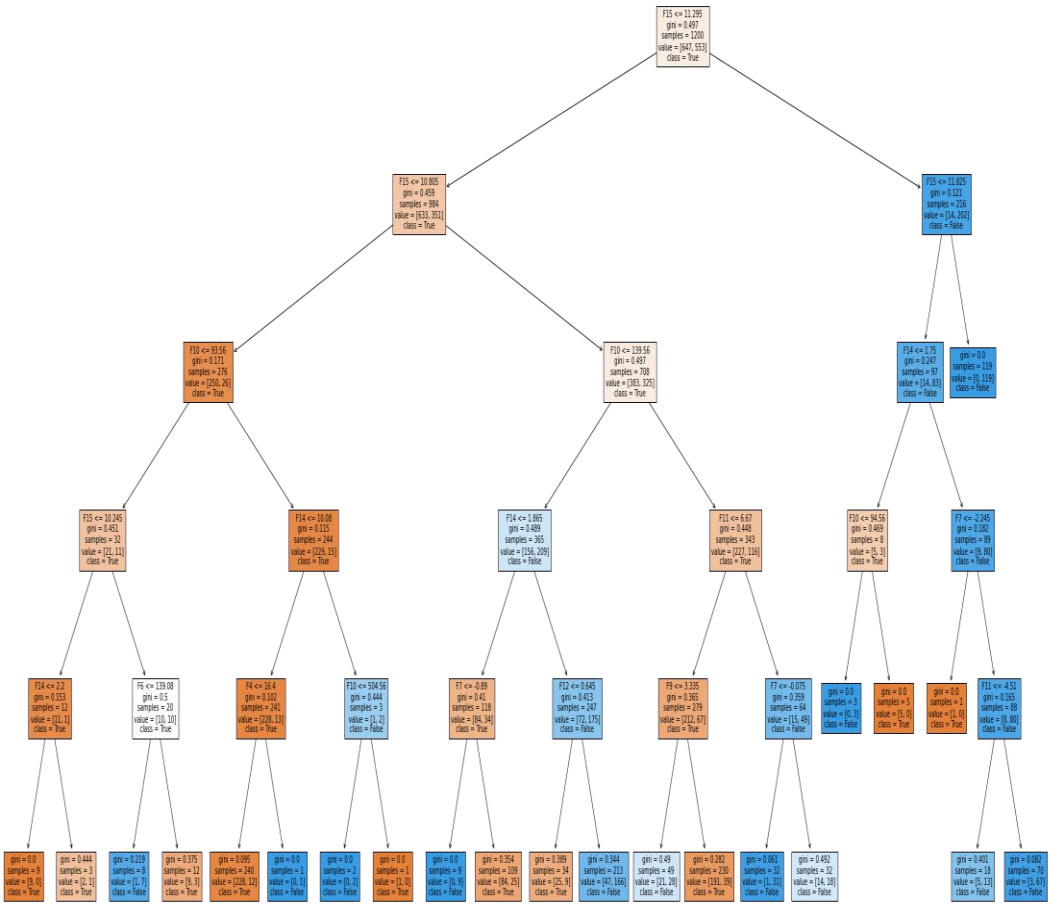
Tuned parameters- criterion(gini),max_depth(5).
Accuracy on test set – 82%

Decision tree feature importances-



From bar graph it can be interpreted that F15 feature has the most importance followed by F14.

Plot of Decision Tree:



IMPLEMENTATION OF SVM

Before feeding data to SVM the data should be scaled. StandardScaler is used here which makes distribution mean = 0 and variance = 1. Here also I created a pipeline and tuned two parameters that is C and gamma.

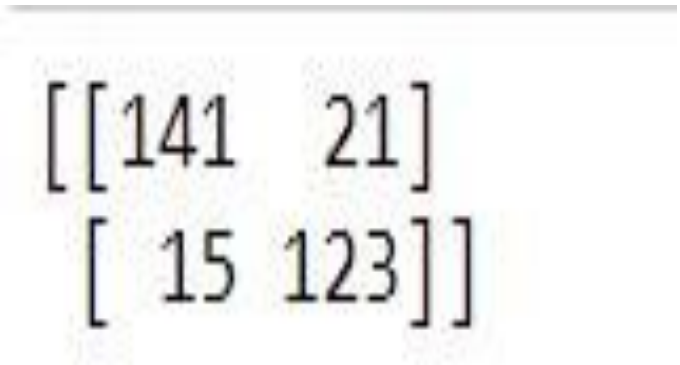
Tuning of Parameters:

C determines the trade off between margin maximization and error minimization(ce802 lineardiscrim_handouts). Default value of kernel is rbf in sklearn svm documentation. Rbf uses non-linear hyperplane. In non linear hyperplane gamma parameter role comes into play, the higher the value of gamma ,model tries to fit the data accurately. If gamma increases risk of overfitting also increases.
GridSearchCv is used to find the optimal parameters which led to highest accuracy.

Final SVM:

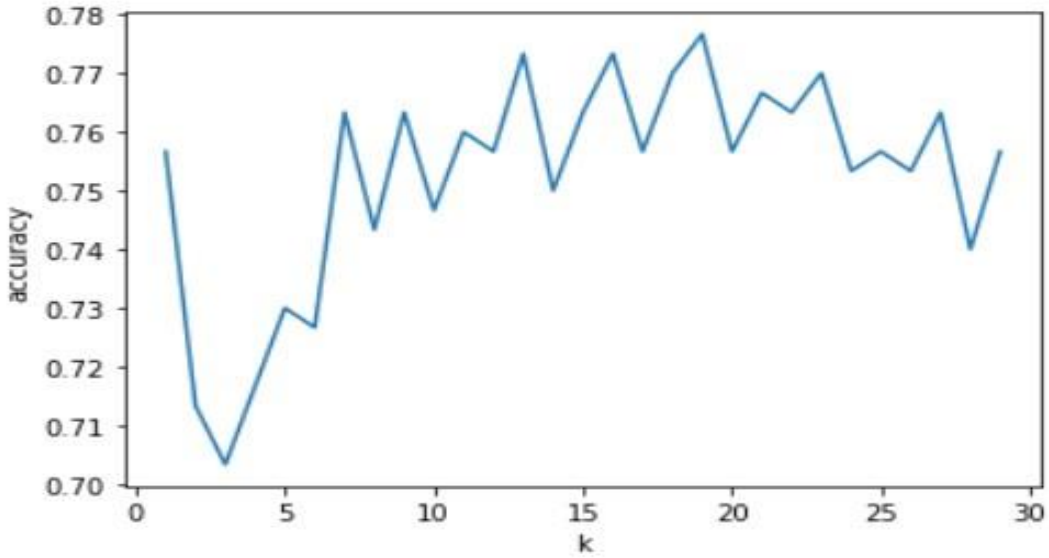
Tuned parameters: C = 100 , gamma = 0.01
Number of support vectors in our model = 481
Accuracy on test set = 88%

Confusion Matrix of Test data:



IMPLEMENTATION OF KNN

KNN model calculates distance between two instances so it is affected by non-scaled data. Here I have also used StandardScaler for scaling data before feeding to KNN model. I created a loop in which k value changes from 1 to 30 and calculated the accuracy on the test data. Got best accuracy when k is 17.

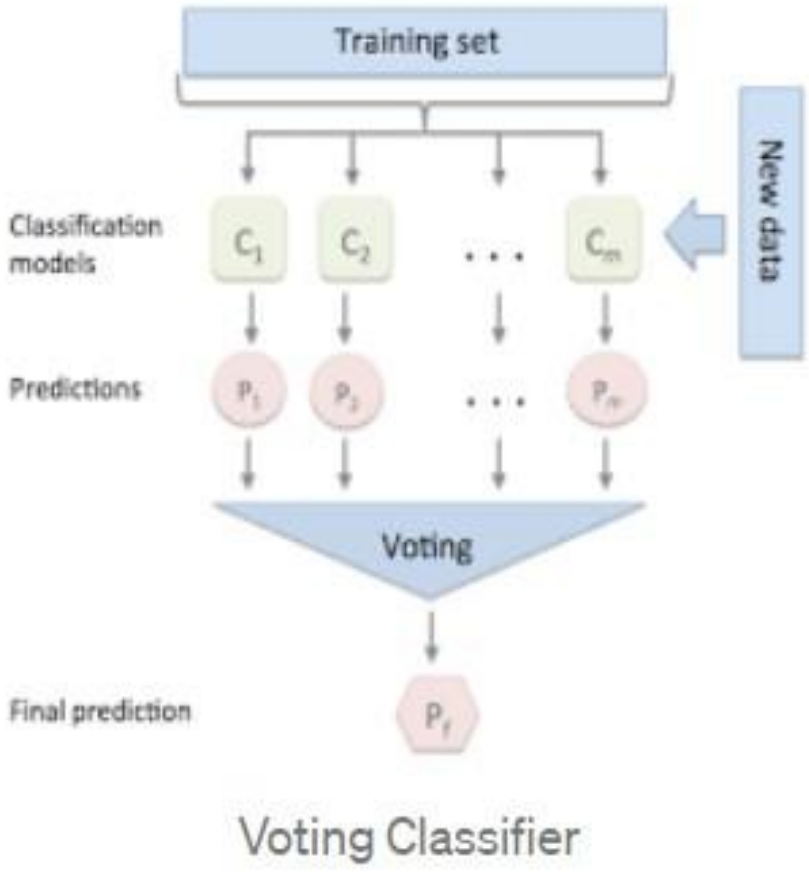


From above table at k=17 accuracy is 77.77%

Confusion Matrix of Test Data:

```
[[142  20]
 [ 53  85]]
```

IMPLEMENTATION OF VOTINGCLASSIFIER



Source of image(https://medium.com/@sanchitamangale12/voting-classifier-1be10db6d7a5)

Voting classifier is one of easiest ensemble function. It takes prediction from all models and return prediction which occur maximum times(hard voting).
It uses strengths of different models and give desired accuracy.
I used all my three models created above and got accuracy of 87.6% on test set.

ACCURACY TABLE OF DIFFERENT MODELS

ML MODEL	D.TREE	SVM	KNN	VOTING CLASSIFIER
ACCURACY	82%	88%	77.77%	87.60%

Test set provided in assignment is predicted by using SVM model.

INTERPRETATION

On the given data SVM out-performs the D.tree and KNN. SVM is a complex and more powerful algorithm which can learn complicated relationships in data. While I was training the models it was the one which had taken lot of time to train. D.tree is simple and is faster to train. D.tree is easy to interpret it means the knowledge which model learnt from data is easy to read whereas in KNN and SVM knowledge learnt by model cannot be expressed in comprehensible way(Comparative study between decision tree and knn of data mining classification technique M Mohanapriya and J Lekha Mrs 2018 J. Phys.: Conf. Ser. 1142 012011). D.tree is fastest to classify new instances as it is just executing number of Boolean comparisons compared to KNN and SVM but as tree grows speed decreases.

3A ADDITIONAL COMPARATIVE STUDY

DATA ANALYSIS

	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F12	F13	F14	F15	F16	Target
0	-39.32	5.72	-13.83	High	UK	0.03	-200.46	122.09	-2.96	6	2.67	8	607.14	19.19	10523.40	-635.80	1051.99
1	-29.38	8.20	-11.07	Medium	UK	56.97	-427.78	74.25	-22.94	4	3.84	10	867.03	12.65	10037.04	-469.73	816.64
2	2.46	20.32	-7.59	Low	Rest	45.00	-329.02	96.98	-10.90	4	13.14	8	-153.66	13.01	15100.28	-662.31	3241.77
3	16.33	2.76	-8.40	High	Rest	0.12	-196.88	42.45	-12.16	12	1.29	6	1461.87	9.19	22518.15	-1100.35	0.00
4	-14.93	9.98	-5.28	Medium	USA	557.61	-249.50	76.25	-20.54	8	2.58	2	-433.89	18.44	20111.46	-752.48	0.00
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
1495	-58.26	7.90	-26.52	High	Europe	1.95	-220.48	43.35	-16.14	8	42.69	8	175.98	14.41	16245.14	-721.31	603.30
1496	-52.83	3.06	-11.25	High	USA	3.12	-191.54	81.12	-13.12	2	11.91	6	-810.09	17.36	15345.11	-900.86	0.00
1497	-21.87	5.56	0.15	Very low	Rest	0.06	-167.14	127.47	-9.64	10	16.17	10	-800.91	14.25	12910.41	-332.72	1070.63
1498	5.24	2.04	1.83	Very high	USA	8.85	-236.68	114.56	-24.24	8	4.50	10	-169.98	12.25	16347.86	-1210.51	0.00
1499	-39.20	21.86	-16.32	High	UK	0.12	-357.78	126.43	-2.12	14	16.80	8	196.86	19.71	13478.48	-807.20	1988.69

1500 rows × 17 columns

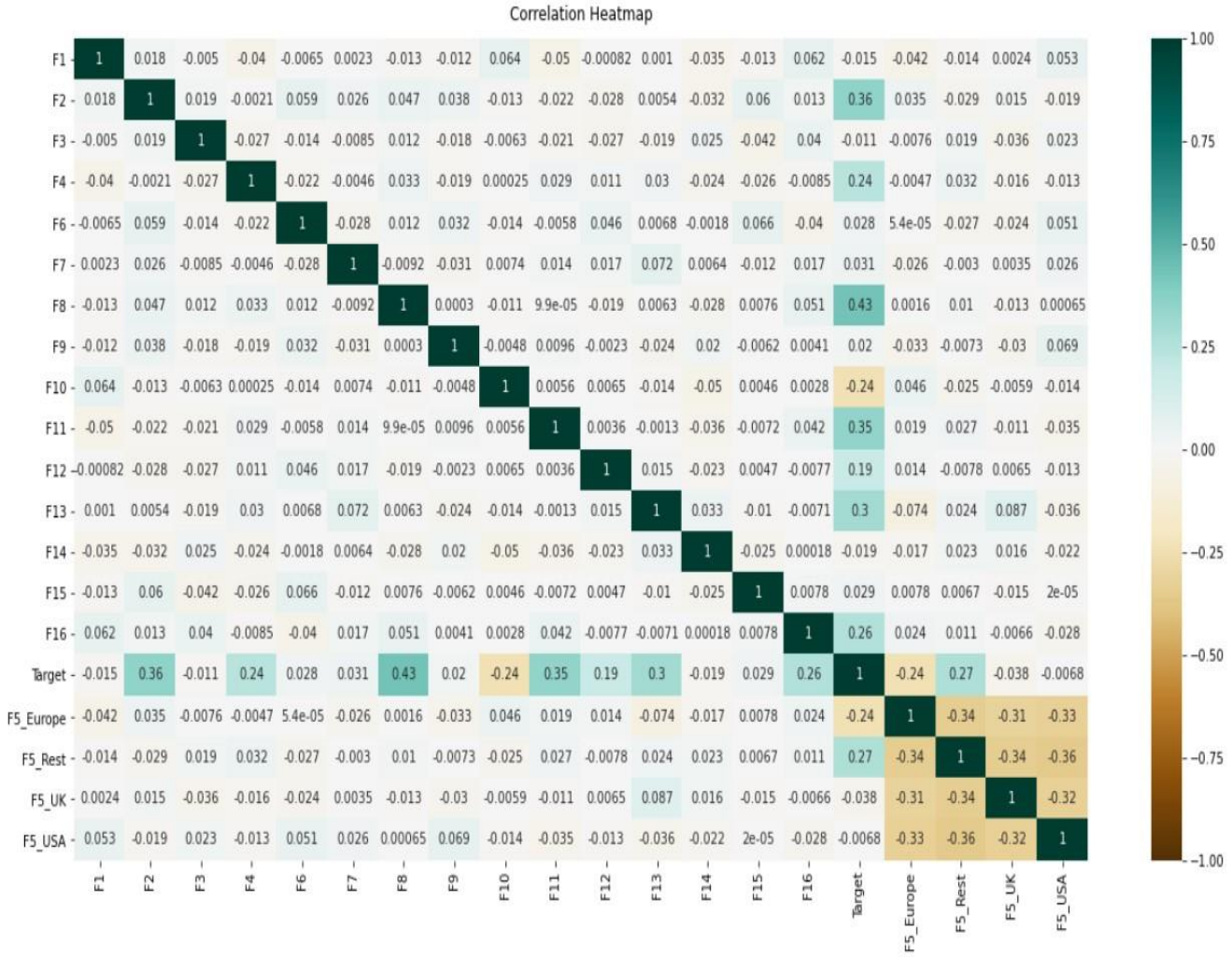
Observing data it is clear that feature F4 and F5 are categorical features. There are no NaN values.

DATA PREPARATION

On F4 feature I am using label encoding as values have ordinal relationship and on F5 feature I am using one hot encoding as values don't have ordinal relationship.

Data is scaled before feeding to ML models using StandardScaler.

Correlation Heatmap:



No two features are highly correlated.

Splitting the data:

It is not good practice to evaluate the model on the same data on which it was trained. It doesnot give good indication on how well the model will perform on unseen data. So data is divided into test and train data.

TRAIN DATA	80%
TEST DATA	20%

IMPLEMENTATION OF LINEAR REGRESSION

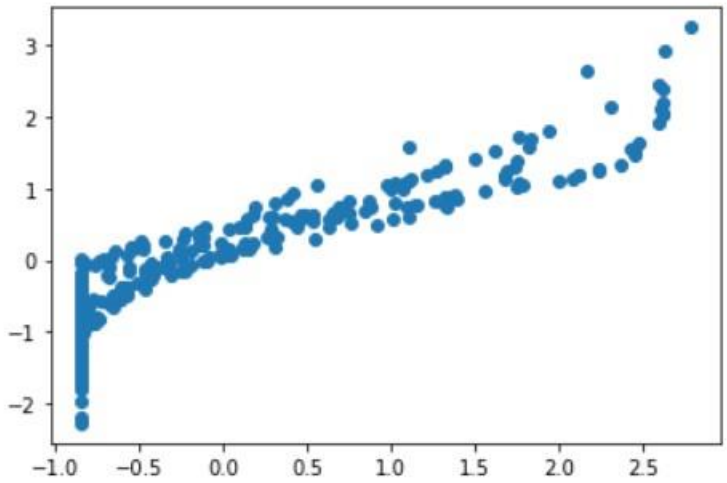
After fitting linear regression model here are the coefficients:

	cofficient
0	-8.786018e-03
1	3.550830e-01
2	-2.131738e-02
3	2.010488e-01
4	5.150152e-03
5	-1.567127e-03
6	3.959933e-01
7	8.715872e-03
8	-2.082835e-01
9	3.362909e-01
10	2.126295e-01
11	2.850550e-01
12	5.619676e-03
13	6.076651e-03
14	2.415269e-01
16	-8.857319e+12
17	-9.221592e+12
18	-8.797670e+12
19	-9.019852e+12

Coefficients provide relationship between dependent and independent variable. From Negative coefficient it can be interpreted that as independent variable increases dependent variable decreases and vice versa.

Scatter plot:

Plot between predicted values of test set from the Linear Regression model and original values of test set.



IMPLEMENTATION OF SVR

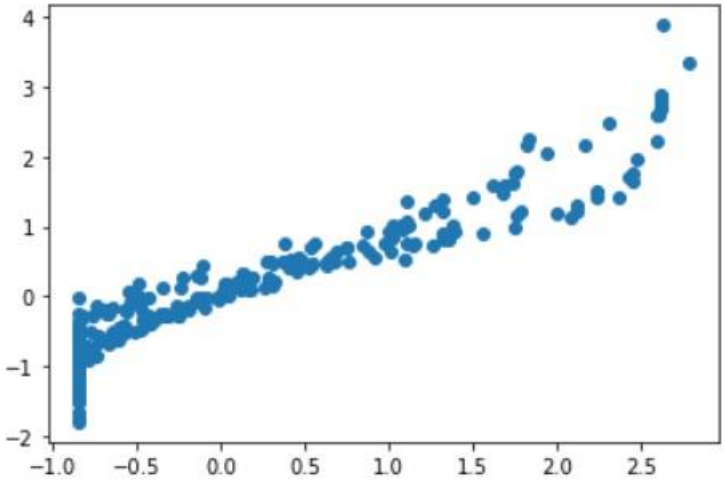
Tuned two parameters one c and another gamma using GridSearchCV.

Final SVR:

Tuned parameters – C = 100 , gamma = 0.001

Scatter plot:

Plot between predicted values of test set from the SVR model and original values of test set.



IMPLEMENTATION OF RANDOM FOREST REGRESSOR

Random forest regressor is a meta estimator that fits a number of classifying decision trees on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting(Scikit learn Documentation RandomForestRegressor)

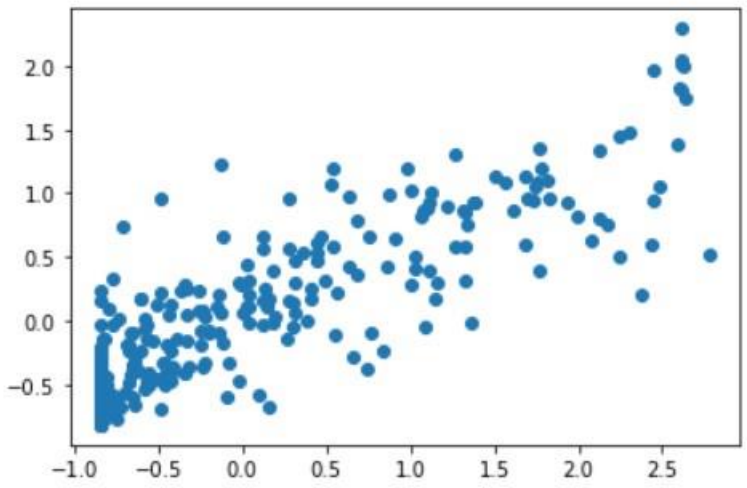
Created a pipeline and here I tuned three parameters first is bootstrap. If bootstrap is False then whole dataset is used in building different trees and if True then subsamples are used. Size of subsamples can be controlled by parameter max_samples.
Second is max_depth and third is n_estimators controls the number of trees.

Final Model:

Tuned parameters: bootstrap-True , max_depth – 9 ,n_estimators – 100

Scatter plot:

Plot between predicted values of test set from the RFR model and original values of test set.



ML MODEL / COMPARISON METRICS	LINEAR REGRESION	SVR	RANDOM FOREST REGRESSOR
MSE	0.1918	0.1050	0.3063
R2 SCORE	0.815	0.8988	0.7050

Test set provided in assignment is predicted by using LinearRegression model.

INTERPRETATION

SVR outperforms the LR and RandomForestRegressor on the given data. Here also SVR took more time to train than other two models. Random forest regressor uses lot of computational power and also take considerable time to train as it creates lot of trees. It is based on concept of ensemble(bagging). Linear regression was simplest model compared to others but is rarely used in real applications because of strong assumption of linearity between dependent and independent variable.

REFERENCES

1. Scikit learn Documentation https://scikit-learn.org/stable/modules/generated/sklearn.pipeline.Pipeline.html
2. Laura Elena Raileanu and Kilian Stoffel, “Theoretical comparison between the Gini Index and Information Gain criteria” https://link.springer.com/article/10.1023/B:AMAI.0000018580.96245.c6
3. Comparative study between decision tree and knn of data mining classification technique M Mohanapriya and J Lekha Mrs 2018 J. Phys.: Conf. Ser. 1142 012011 https://www.researchgate.net/publication/329329175_Comparative_study_between_decision_tree_and_knn_of_data_mining_classification_technique
4. Scikit learn Documentation RandomForestRegressor https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html#:~:text=A%20random%20forest%20regressor.,accuracy%20and%20control%20over%2Dfitting.