Remaining Exercises

Alireza Afzal Aghaei

July 26, 2021

Abstract

In this report we answer some remaining questions in four past exercises. The report contains four different sections, each belonging to an exercise.

1 Exercise 1.1 (aka 2)

1.1 AirBnB

We first do some data cleaning preprocess such as filling NaN values, dropping duplicate rows, dropping some columns such as id and host name, etc. Now we draw some different plots from the data in the figure 1 and 2. See the figure captions for detailed information about the figures.

1,2) there are a lot of useful information can be derived form this data. we refer the reader to figures 1 and 2.

3) See figure 1 (d) for the busiest hosts. This maybe is the result of lower price.

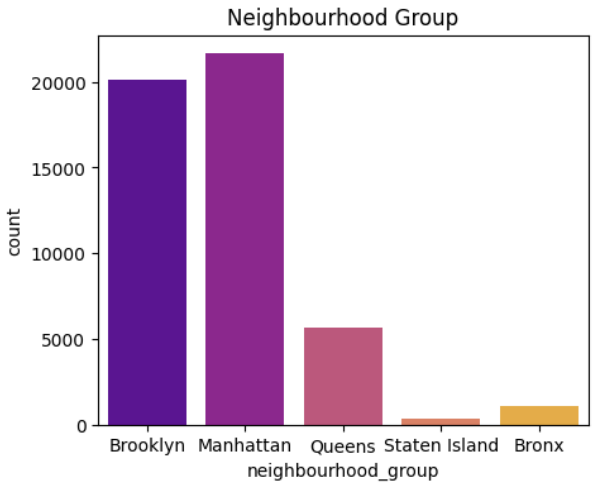
4) For this question we aggregate neighbourhood_group by counts. The results are reported on the table 1. For Brooklyn, because of lower price we see the higher traffic. For Manhattan we have a large variance and this maybe became the reason for high traffic.

1.2 Football dataset

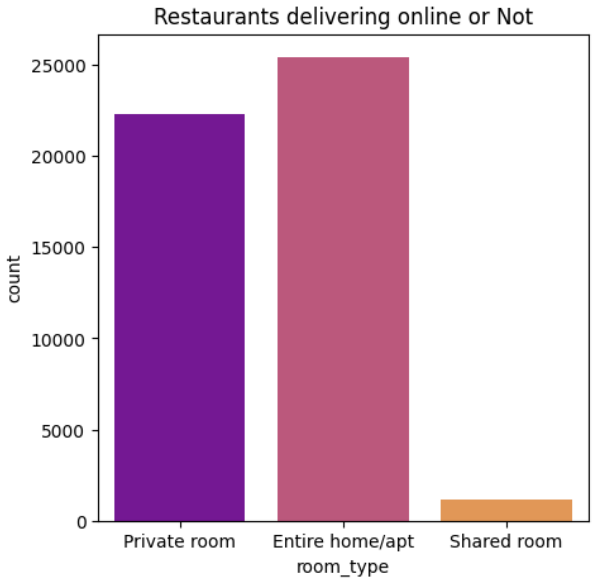
Fortunately this dataset does not contain any missing value and therefore there is no need to do complicated preprocessing steps. in the figure 4 we plot some of dataset features.

neighbourhood_group	count
Bronx	874
Brooklyn	16390
Manhattan	16471
Queens	4566
Staten Island	314

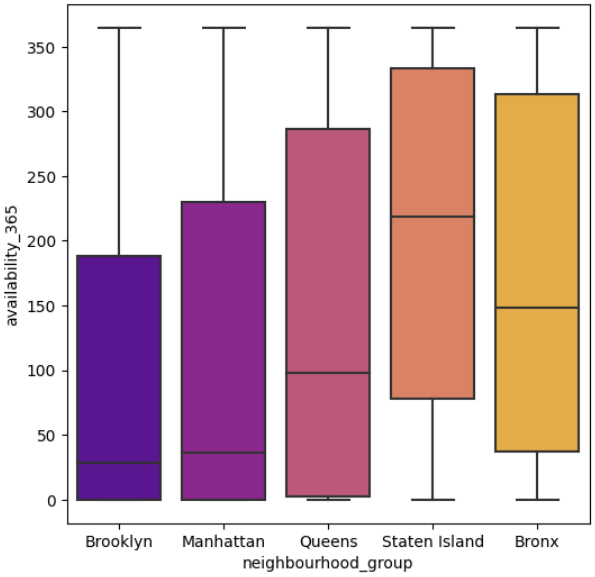
Table 1: AirBnB dataset neighbourhood counts.



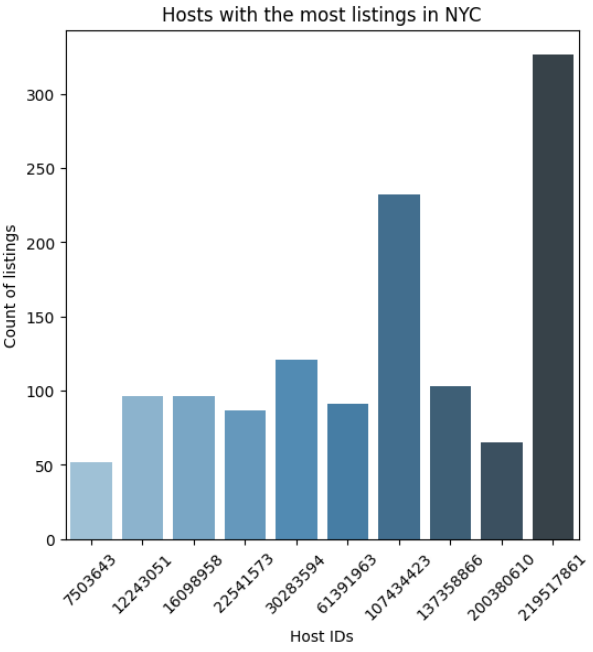
(a) Neighbourhood Group Count Plot



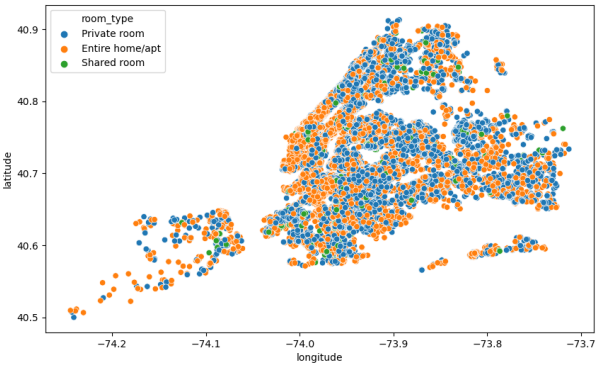
(b) Restaurants delivering online or Not



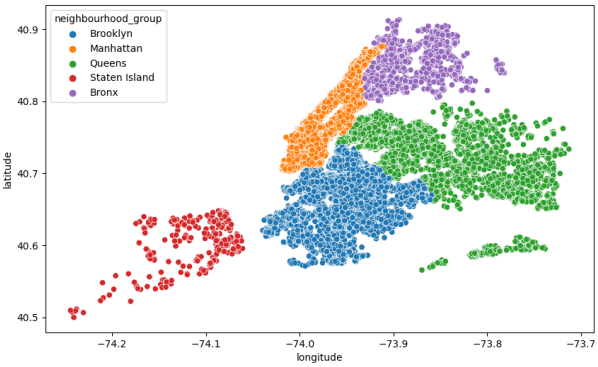
(c) Relation between neighbourgroup and Availability of Room



(d) Most listing hosts

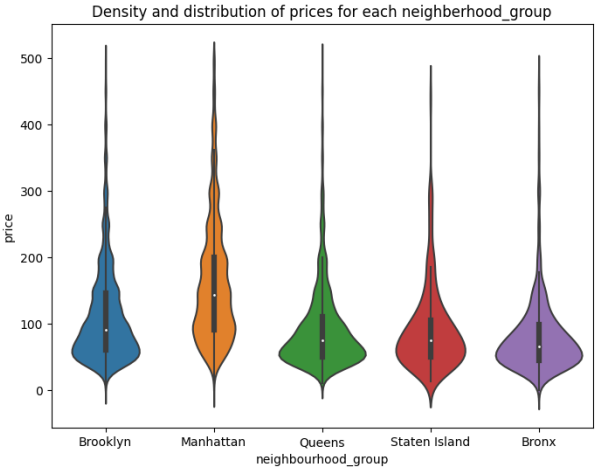


(e) Room type plot

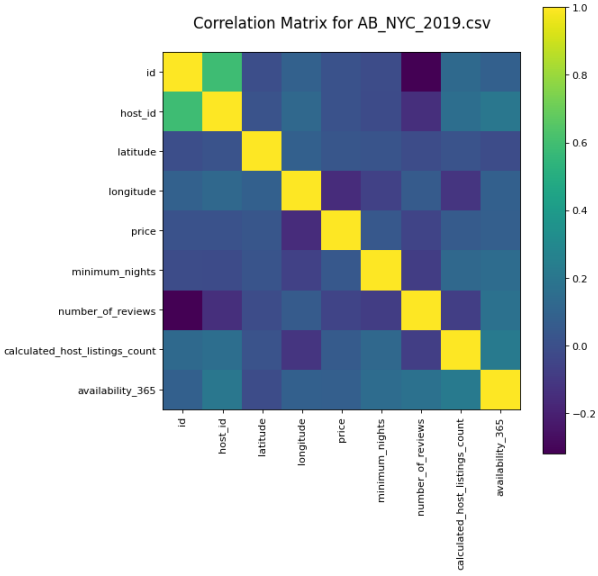


(f) Neighbourhood Group plot

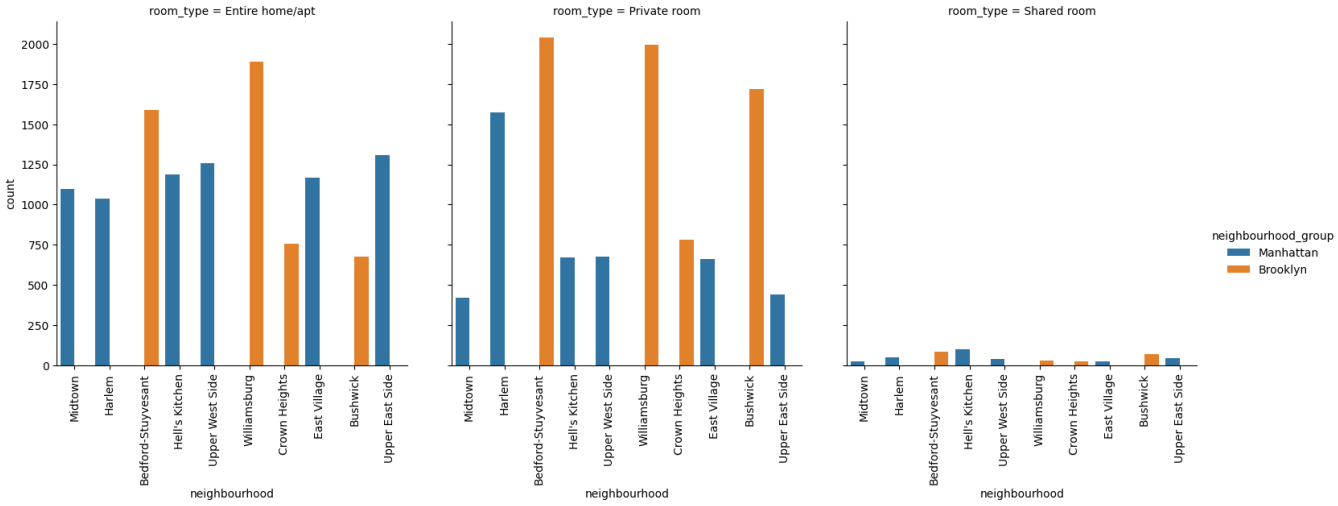
Figure 1: AirBnB plots



(a) Neighbourhood Price

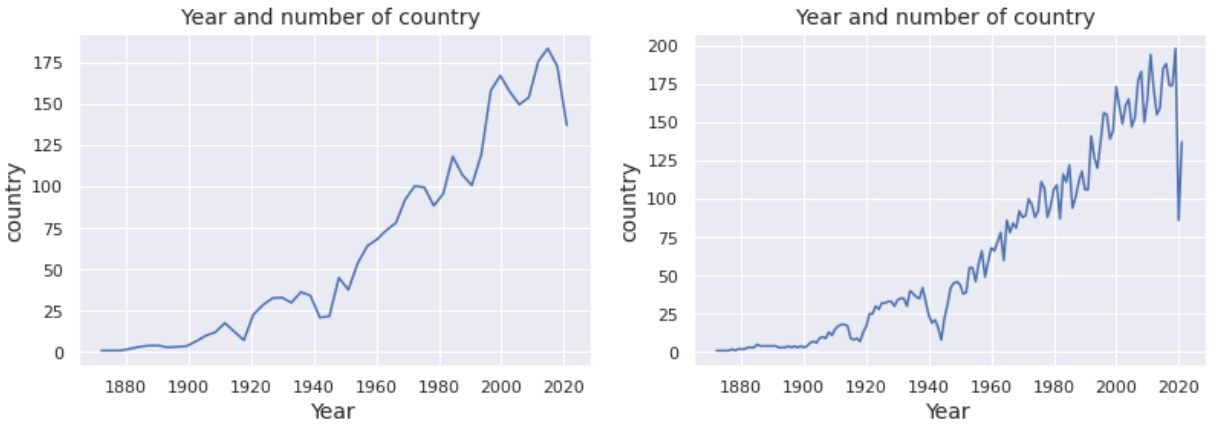


(b) Correlation matrix



(c) Neighbourhood/Room

Figure 2: AirBnB plots



(a) Smoothed match trend

(b) Real Match trend

Figure 3: Football dataset plots

who_win	home_score	away_score	total_score	who_win	result
Brazil	1386	841	2227	Brazil	638
England	1171	1041	2212	England	587
Germany	1244	891	2135	Germany	563
Sweden	1093	809	1902	Argentina	534
Argentina	1167	624	1791	Sweden	510
Hungary	1031	739	1770	South Korea	458
Netherlands	925	600	1525	Mexico	447
Mexico	955	486	1441	Hungary	443
France	908	525	1433	Italy	438
South Korea	889	518	1407	France	428

Table 2: Best and most winning teams

1) See table 2 for the best teams ever and the most winning teams.

2) See table 3.

3) see figure 4.

4) As you see in the figure 3, WWI, WWII and Coronavirus affected the number of matches in the era.

5) see table 3.

6,7) see figure 4 and 3.

2 Exercise 3

2.1 Section 3

6,7,8) We implemented two different methods for feature selection. One using Ordinary Least Squares (OLS) and the other using Logistic Regression method. First one uses significance_level parameter to control the number of features which returns. The latter uses ROC AUC score. We reported the score of the selected features in tables 4 and 5. The results for PCA is reported on the table 6.

2.2 Section 4

6) Logistic regression is a classification algorithm traditionally limited to only two-class classification problems. If you have more than two classes then Linear Discriminant Analysis is the preferred linear classification technique. The general LDA approach is very similar to a Principal Component Analysis, but in addition to finding the component axes that maximize the variance of our data (PCA), we are additionally interested in the axes that maximize the separation between multiple classes (LDA). isted below are the 5 general steps for performing a linear discriminant analysis;

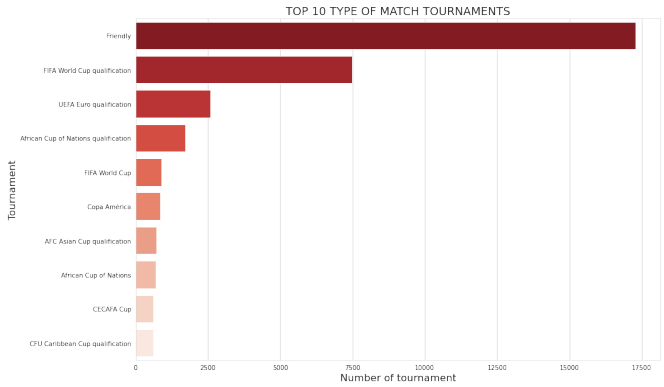
Year	who_win	count
1997	Brazil	20
2008	Trinidad and Tobago	17
2001	Saudi Arabia	17
1993	Mexico	17
1997	China PR	17
1999	Brazil	16
2004	Japan	16
2015	South Korea	16
1975	South Korea	16
1997	South Korea	15

Country	Count
United States	793
Malaysia	428
France	375
United Arab Emirates	292
South Africa	286
Qatar	255
England	253
Spain	224
Brazil	215
Thailand	208

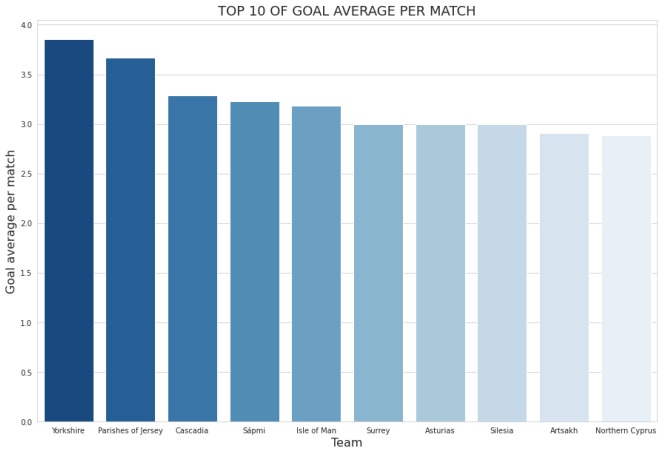
Table 3: Teams dominated different eras (Left), Which countries host the most matches where they themselves are not participating in (Right)

sig	features	Train Acc.	Test Acc.	precision	recall	f1 score
0.001	'ram-'battery_power'-'px_height'-'px_width'-'mobile_wt'	0.97915	0.97447	0.97478	0.97446	0.97445
0.01	'ram-'battery_power'-'px_height'-'px_width'-'mobile_wt'	0.97915	0.97447	0.97478	0.97446	0.97445
0.025	'ram-'battery_power'-'px_height'-'px_width'-'mobile_wt'	0.97915	0.97447	0.97478	0.97446	0.97445
0.05	'ram-'battery_power'-'px_height'-'px_width'-'mobile_wt'-'int_memory'-'battery'	0.98048	0.97548	0.97593	0.97547	0.97547
0.075	'ram-'battery_power'-'px_height'-'px_width'-'mobile_wt'-'int_memory'-'battery'	0.98048	0.97548	0.97593	0.97547	0.97547
0.1	'ram-'battery_power'-'px_height'-'px_width'-'mobile_wt'-'int_memory'-'battery'-'dual_sim'	0.98159	0.97698	0.97740	0.97697	0.97696
0.2	'ram-'battery_power'-'px_height'-'px_width'-'mobile_wt'-'int_memory'-'battery'-'dual_sim'-'wifi'-'network'	0.98426	0.97547	0.97602	0.97546	0.97547

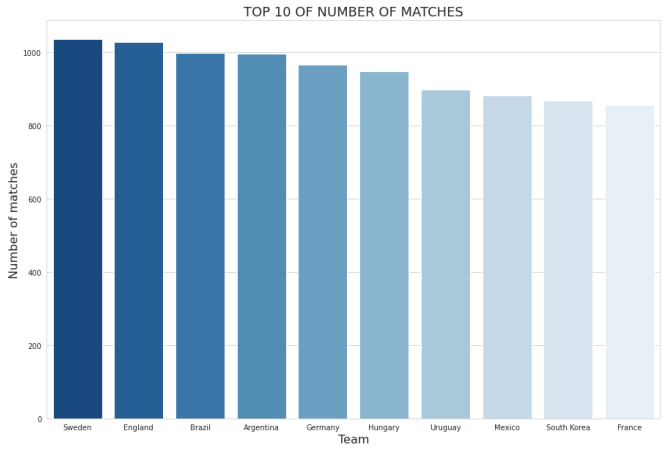
Table 4: Feature selection using OLS with different significance values



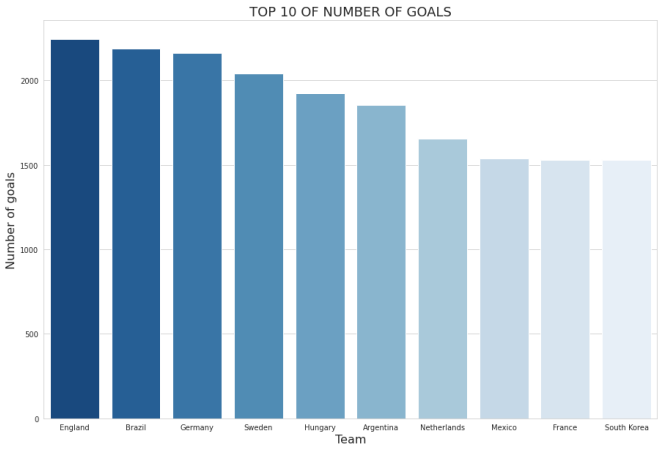
(a) Tournament types



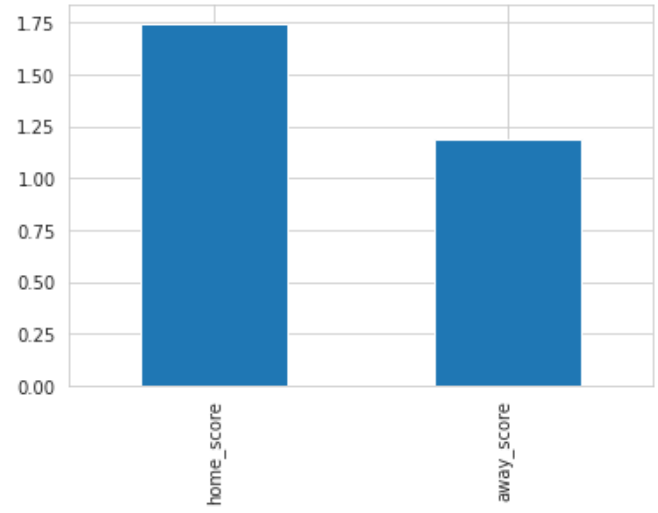
(b) Teams with most average goals



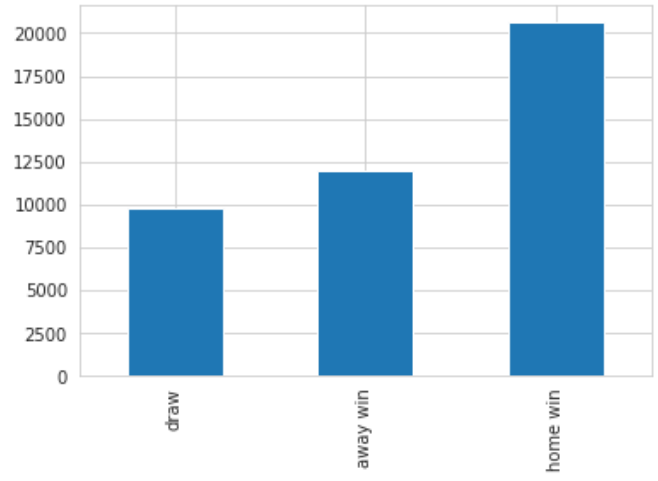
(c) Teams with most matches



(d) Teams with most goals



(e) Average score



(f) Home vs away wins

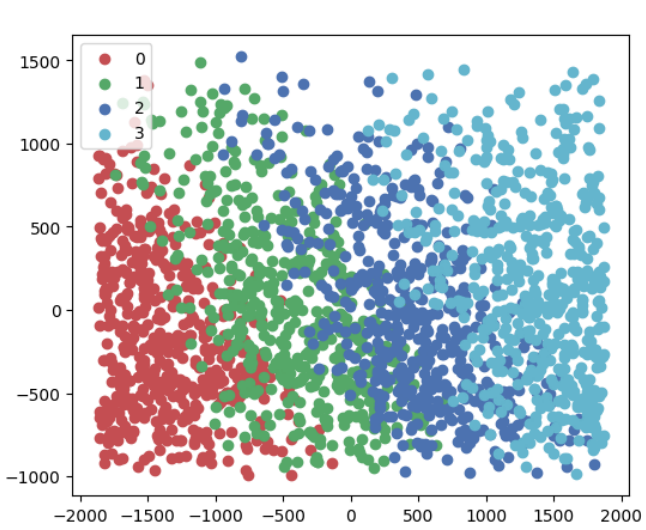
Figure 4: Football dataset plots

Feature to add	ROC AUC
ram	0.9758482385730213
battery_power	0.9895917279821628
px_height	0.9973826978818284
px_width	0.9995528874024526
mobile_wt	0.9997997101449275
dual_sim	0.9998397770345596
battery	0.9998732441471573
aspect_ratio	0.9998732441471573
m_dep	0.9998666220735786
talk_time	0.9998398885172797
sc_w	0.9998332664437012
wifi	0.9998065328874025
lcd	0.9998198885172798
sc_h	0.9998198885172798
clock_speed	0.9997998439241919
network	0.9997598439241917
blue	0.9997530880713489
ppi	0.9997196878483836
pc	0.9996863099219622
fc	0.9996261538461539
int_memory	0.9995395986622073
n_cores	0.9995062430323299
touch_screen	0.9993993088071349

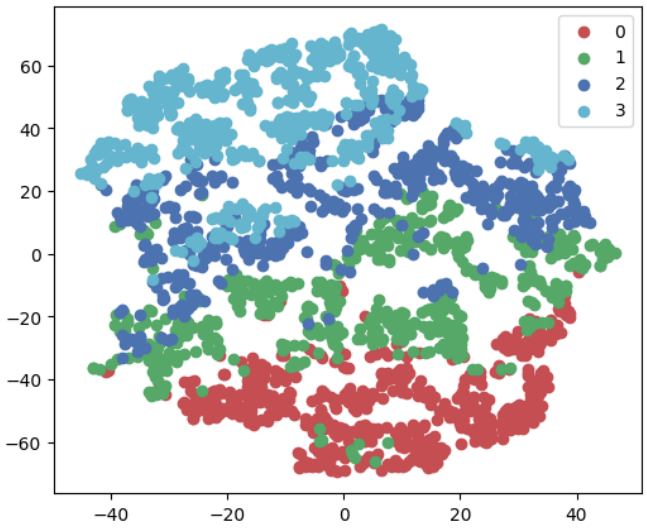
Table 5: ROC AUC score of incrementally forward selected features

N components	Train accuracy	Test accuracy	precision	recall	f1 score
5	0.97903	0.97447	0.97478	0.97446	0.97445
5	0.97903	0.97447	0.97478	0.97446	0.97445
5	0.97903	0.97447	0.97478	0.97446	0.97445
7	0.98109	0.97698	0.97737	0.97697	0.97697
7	0.98109	0.97698	0.97737	0.97697	0.97697
8	0.98170	0.97698	0.97740	0.97697	0.97696
10	0.98454	0.97497	0.97554	0.97496	0.97496

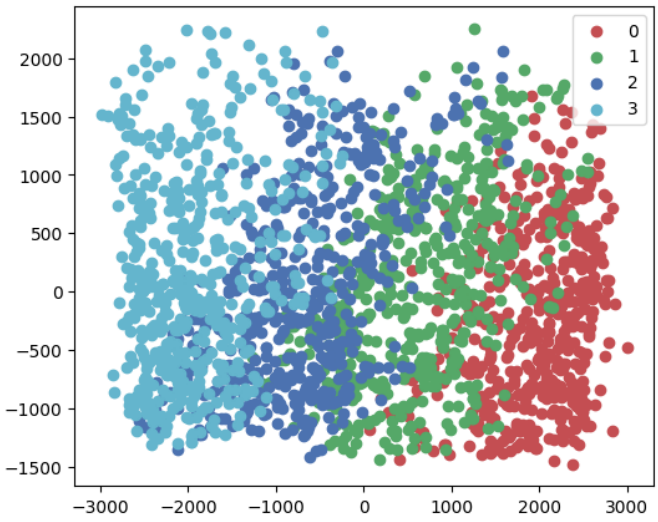
Table 6: Feature selection using PCA



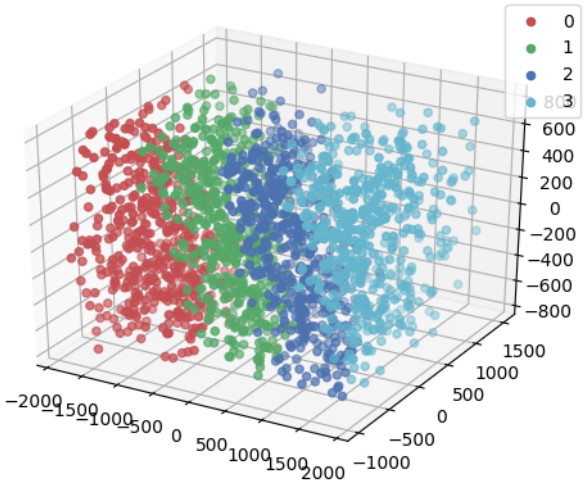
(a) PCA 2D



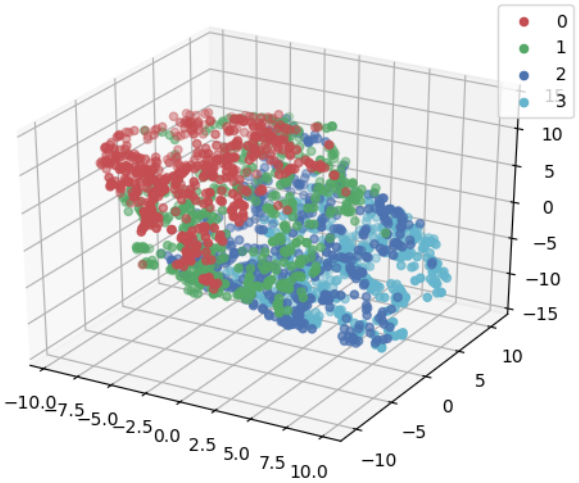
(b) TSNE 2D



(c) ISOMap 2D



(d) PCA 3D



(e) TSNE 3D

Figure 5: Data visualization using PCA, TSNE and ISOMap

- Compute the d-dimensional mean vectors for the different classes from the dataset.
- Compute the scatter matrices (in-between-class and within-class scatter matrix).
- Compute the eigenvectors (ee1,ee2,...,eed) and corresponding eigenvalues (λ_1,λ_2,...,λ_d) for the scatter matrices.
- Sort the eigenvectors by decreasing eigenvalues and choose k eigenvectors with the largest eigenvalues to form a d×k dimensional matrix W (where every column represents an eigenvector).
- Use this d×k eigenvector matrix to transform the samples onto the new subspace. This can be summarized by the matrix multiplication: $Y = X \times W$ (where X is a $n \times d$-dimensional matrix representing the n samples, and y are the transformed $n \times k$-dimensional samples in the new subspace).

All of these items are explained on this link. For the comparison between PCA and LDA we refer the reader to these papers: paper 1 and paper 2.

7) We can use different statistical significance tests to chose the best model for the final prediction. For example McNemar's test or 5×2 Cross-Validation (see this paper) or other refinements on 5×2 Cross-Validation (see this paper). This article explains these methods. See this article for more detailed information.

8) The Matthews correlation coefficient (MCC) or phi coefficient is used in machine learning as a measure of the quality of binary (two-class) classifications. the MCC is defined identically to Pearson's phi coefficient. The coefficient takes into account true and false positives and negatives and is generally regarded as a balanced measure which can be used even if the classes are of very different sizes. The MCC is in essence a correlation coefficient between the observed and predicted binary classifications; it returns a value between -1 and +1. A coefficient of +1 represents a perfect prediction, 0 no better than random prediction and -1 indicates total disagreement between prediction and observation. However, if MCC equals neither -1, 0, or +1, it is not a reliable indicator of how similar a predictor is to random guessing because MCC is dependent on the dataset.

$$|\text{MCC}| = \sqrt{\frac{\chi^2}{n}} \tag{1}$$

where n is the total number of observations.

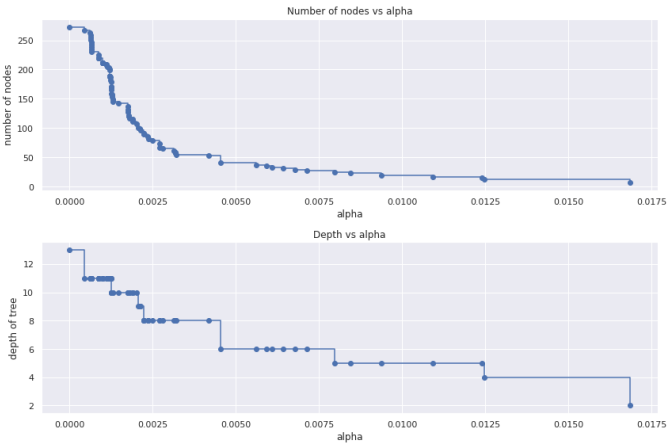
While there is no perfect way of describing the confusion matrix of true and false positives and negatives by a single number, the Matthews correlation coefficient is generally regarded as being one of the best such measures. The MCC can be calculated directly from the confusion matrix using the formula:

$$\text{MCC} = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \tag{2}$$

3 Exercise 4

11) In order to use tree pruning we apply two different approaches. One is to use min_samples_leaf and max_depth parameters of sklearn's decision tree class. Second is using Minimal Cost-Complexity Pruning

(a) Decision tree Impurity



(b) Number of nodes vs alpha and Depth vs alpha indices for different values of ccp_alpha

Figure 6: The effect of ccp_alpha on the tree structure

algorithm. The first method is used in the previous sent exercise. Using three different values for max_depth parameter, we saw that changing this parameter affects the model accuracy. The best accuracy has been gained by max_depth=10. For more detailed info see table 2, page 3, project 3 exercise.

The next approach is using Minimal Cost-Complexity Pruning algorithm. In sklearn's DecisionTreeClassifier, this pruning technique is parameterized by the cost complexity parameter, ccp_alpha. Greater values of ccp_alpha increase the number of nodes pruned. We see this behaviour in the Fig 6.

14) Classification rules represent knowledge in the form of logical if-else statements that assign a class to unlabeled examples. The earlier rule learning algorithms (Separate and conquer, and The 1R algorithm) have some problems like slow performance for an increasing number of datasets, and prone to being inaccurate on noisy data. Johannes Furnkranz and Gerhard Widmer in 1994 proposed a solution towards solving these problems. Their incremental reduced error pruning algorithm (IREP) uses a combination of pre-pruning and post-pruning methods that grow very complex rules and prune them before separating the instance from the complete dataset. The RIPPER (repeated incremental pruning to produce error reduction)algorithm is introduced by W. Cohen in 1995, which improved upon IREP to generate rules that match or exceed the performance of decision trees. Having evolved from several iterations of the rule learning algorithm, the RIPPER algorithm can be understood in a three-step process: Grow, Prune, Optimize. The first step uses a 'separate and conquer' method to add conditions to a rule until it perfectly classifies as a subset of data. Just like decision trees, the information gain criterion is used to identify the next splitting attribute. When increasing a rule's specificity no longer reduces entropy, the rule is immediately pruned. Until reaching stopping criterion step one and two are repeated at which point the whole set of rules is optimized using a variety of heuristics.

25) We created different indicators, such as simple moving average with two different time windows, simple moving average ratio, Relative Strength Index (RSI), Moving Average Convergence Divergence

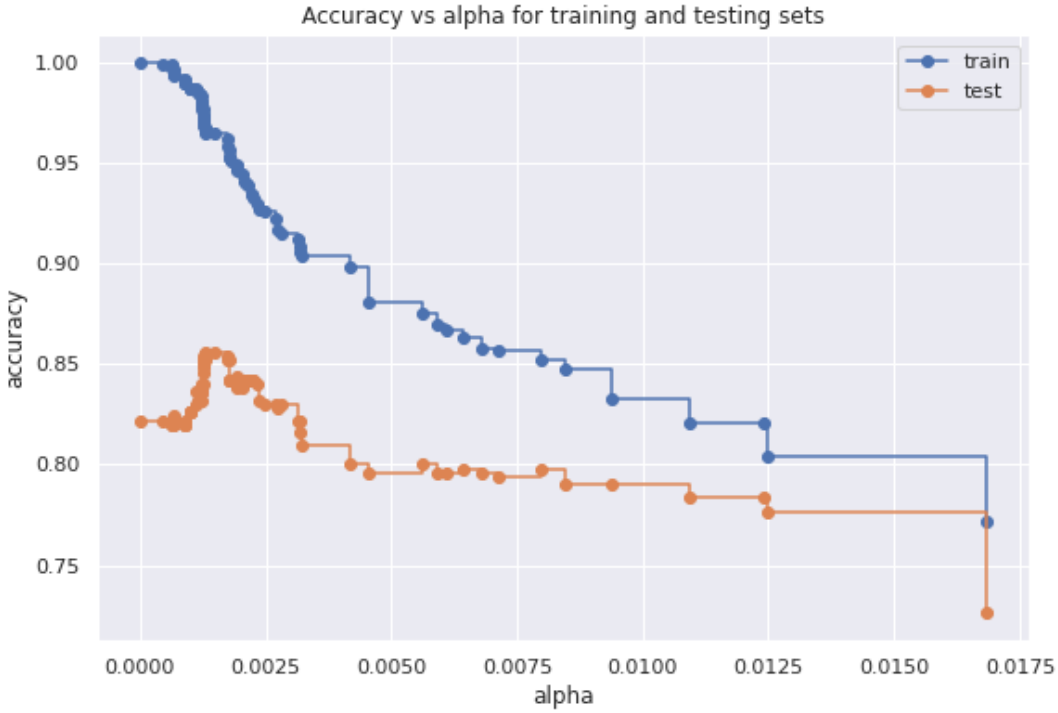


Figure 7: The impact of ccp_alpha on the train and test data

Method	Algorithms	MAE
Bagging	LinearRegression	.1581
Bagging	DecisionTreeRegressor	.1784
Bagging	MLPRegressor	.1537
Forest	RandomForestRegressor	.1785
Boosting	AdaBoostRegressor	.1543
Voting	GradientBoosting, RandomForest, LinearRegression	.1573

Table 7: Ensemble methods for tournament dataset

(MACD), etc. The new data has been tested on a Multi-layer perceptron model. The model is same as exercise file 3, section 3.5 page 8. The new result was worse than that. RMSE increased from 12.36 to 1432! The accuracy is also reduced from 100% to 68%.

27) The results are reported on the table 7.