Preparation

Before starting work on given tasks, all needed libraries have been properly installed.
Some preliminary steps have been done before working with two given projects:

Question 1

1. Mall_Customers dataset has been imported into custData data frame;
2. Badly named column names in Mall_Customers have been renamed properly (‘AnnualIncome’ and ‘SpendingScore’);
3. custData has been checked for missing values;
4. User-defined function segmentCharacter() has been created. This function plots four graphs (histogram and boxplots) of each characteristic for each segment that has been assigned by clusterization. The argument of this function takes the vector of assigned clusters.

Question 2

1. Credit_Fraud dataset has been imported into creditFraud data frame;
2. creditFraud has been checked for missing values;
3. ‘Class’ variable has been converted into factor variables, and levels 0 and 1 have been renamed into ‘legitimate’ and ‘fraud’ respectively;
4. ‘Time’ and ‘Amount’ features have been scaled since KNN and SVM methods require a dataset to be scaled. Scaled dataset has been named as creditFraud.sc;
5. For further classification, the datasets creditFraud and creditFraud.sc have been randomly split into training (70%) and test (30%) sets;
6. User-defined functions oversampling(), undersampling(), smotesamping() have been created. These functions represent by themselves three resampling techniques: random oversampling, random undersampling and SMOTE respectively. The compulsory input parameter is an imbalanced train set, the output is a balanced train set;
7. User-defined functions knnMethod(), ldaMethod(), logisticregMethod(), dectreeMethod(), rfMethod(), SVMMethod() have been created. These functions perform KNN, LDA, Logistic Regression, Decision Tree, Random Forest and Support Vector Machine classification respectively. Compulsory input parameter for all methods is train set, for SVM additional compulsory parameter is kernel type. As an output these functions return AUC value;
8. User-defined functions samplingBoxPlot() and samplingBoxPlotSVM() have been created. These functions draw boxplots with AUC values for each resampling method and boxplot with AUC values for imbalanced train set. Since SVM has 8 possible configurations, for this method the separate function has been created. Compulsory input parameters are arrays of AUC values for each resampling technique and array with AUC values for imbalanced train set;
9. User-defined function accuracyBoxPlot() has been created. This function plots boxplots with AUC values of considered classification methods with their best resampling techniques, obtained from point 8. Compulsory input parameters are arrays of AUC values of best resampling technique.

One important thing that has been noticed is that even with set.seed(value) function the output from random processes can be different at each execution. The possible solution for this is to write RNGkind(sample.kind = “Rounding”) after importing all the needed libraries.

Project 1

Goals of shopping facilities include retaining their frequent customers, attracting new ones and increasing the profit. Before creating and applying any advertisement strategy it is vital to understand the customer preferences in order to scale up business. It is impossible to look at details of each customer and create a unique business strategy for him. What possible to do is to look at customer base data and cluster all of customers into segments based on their purchasing habits and then create a unique advertisement strategy for each segment and apply it for customers based on their belonging to the cluster.

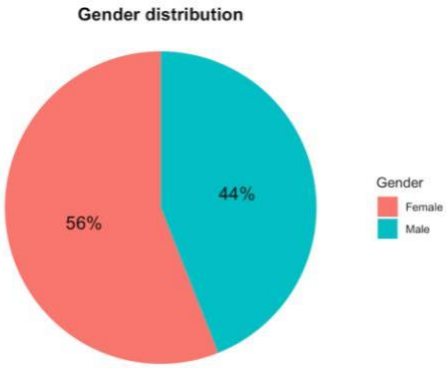
Therefore, the goals of this project are to:

1. Explore customer data and find any interesting insights;
2. Perform clustering and find similarities between customers in each group, which ultimately will help shopping facilities to find unique strategies for each customer segment.

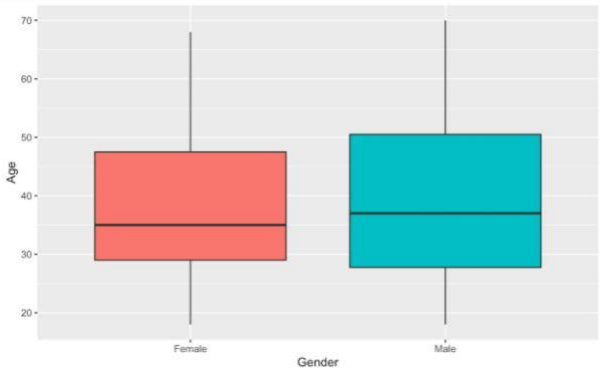
Data Exploration

Before conducting any analysis, it is worthwhile to gather the basic sense of the data set.

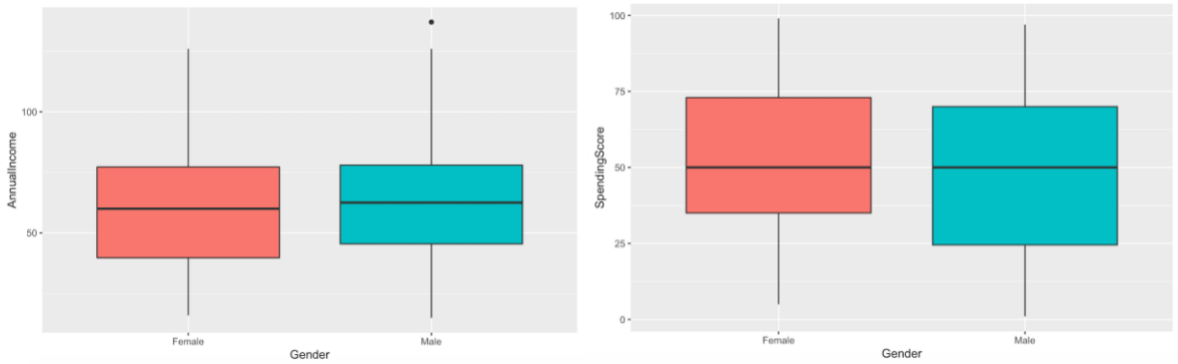
Firstly, the relationships between gender and other features have been presented by several charts below.



From the above pie chart which explains the distribution of Gender in the Mall, it can be seen that the females are in the dominant position with a share of 56%, whereas the Males have a share of 44%.

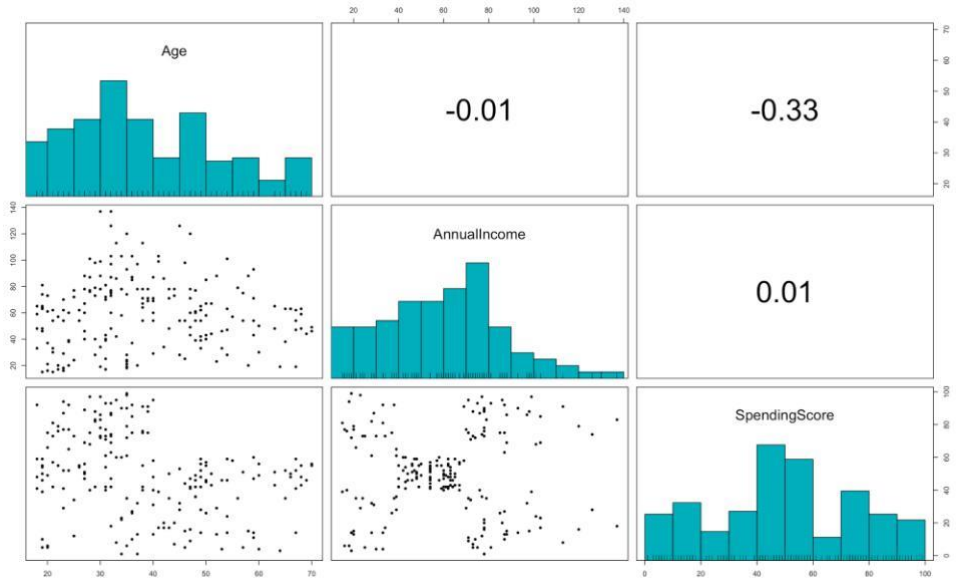


The box plot shows the distribution of age by gender. The majority of women in the mall are at the age of 29 to 48, with the average age of 35, and most men in the mall are 27 to 51, with a slightly older average age of 37.



From the above plots, it can be observed that the average annual income of women is slightly lower than that of men, and more women have incomes lower than 50k dollars, while more females have spending scores higher than 50. It can be inferred that in general, women have stronger purchasing power than men.

Secondly, the dependencies between age, annual income and spending score can be extracted from the paired graphs below.



On the diagonal there are three histograms. The first one shows the distribution of age groups in the mall. The most regular customers for the Mall aged around 30-35, whereas the senior age group is the least frequent visitor of the Mall. The second histogram presents the distribution of the annual income of the customers. Most of the people earn around 50-75k dollars a year, while few people have earnings of more than 100k dollars. From the last histogram, it can be observed that most of the customers have their Spending Score in the range of 40-60.

The scatter plot of age and annual income shows that customers with highest incomes are in the medium-age group from 30 to 40, and with the increasing age, the income shows a decreasing trend. The scatter plot at the left-bottom shows that customers in the younger age group usually have higher spending scores than customers aged from 40 to 70. Points in the scatter plot for annual income and spending score shows 5 clearly separated groups: low income low spending score, low income high spending score, medium income medium spending score, high income low spending score, and high income high spending score.

In addition, values in the upper part (-0.01, -0.33 and 0.01) represent the Pearson measure of the linear dependency between variables. These three values are all close to 0, indicating that there is no or little linear dependency between each two pairs.

Ma, Jinyi 190022007
Zubkov, Vadim 160013580
Jiacheng, Han 190004047
Stadnik, Kseniya 190045224
Sosnovskikh, Tatiana 160019978

Clusterization

The second goal of this project is to group customers into segments using their characteristics. This is an unsupervised machine learning task and more specifically this is a clusterization task.

Methodologies

The most popular algorithm of clustering is K-means clustering. One of the requirements for this method is that all features should be numerical. The used customer data is mixed data: with numerical features and categorical one, which is 'Gender'. Therefore, K-means cannot be directly applied to the given dataset. One of the ways to overcome this issue is to transform the categorical variable into a binary vector using one-hot vector encoding and then apply K-means method to the transformed data. The downside of this approach is that the sample space for categorical data is discrete, and doesn't have a natural origin, thus, a Euclidean distance function on a transformed space isn't really meaningful. Another downside is the curse of dimensionality, which is not the problem here.

The other way of dealing with categorical data is to apply advanced clustering algorithms that can directly work with mixed data. It has been decided to experiment with three advanced methods that haven't been learnt in ML course as well as in any other BA courses. Those methods are:

1. k-medoids or partitioning around medoids (PAM) (Kaufman and Rousseeuw, 1987);
2. simple and fast k-medoid (SFKM) (Park, Lee and Jun, 2006);
3. k-prototypes (Szepannek, 2018).

The first two algorithms have been performed with the help of 'kmed' package (Budiaji, 2019) and the last one with the help of 'clustMixType' package (Szepannek and Aschenbruck, 2020). Technical high-level overview is represented below.

- PAM and SFKM

The difference between these two methods is only in the initial medoid selection where the PAM selects the initial medoid randomly.

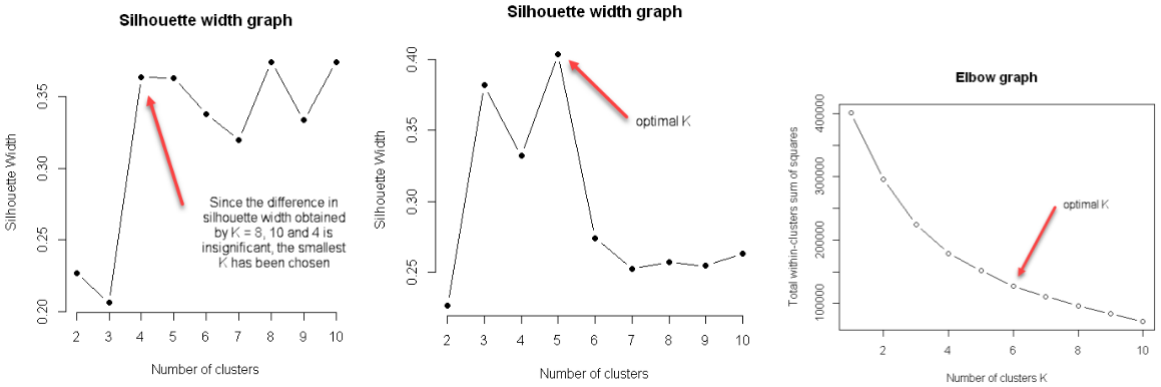
The compulsory inputs for SFKM are a distance matrix or distance object and a number of clusters, for PAM one more attribute is initial medoid. Since Euclidean distance is not applicable here, the Gower distance has been calculated for the mixed dataset (Martin, 2016). In order to choose the optimal number of clusters, the silhouette width has been used as an internal validation metric (Rousseeuw, 1987).

- k-prototypes

In simple words, the k-prototype method is just the combination of k-means for numeric and k-modes for categorical variables. The compulsory inputs for k-prototype are data and a number of clusters. In order to choose the optimal number of clusters, total within-cluster variation has been chosen as a validation metric.

Results

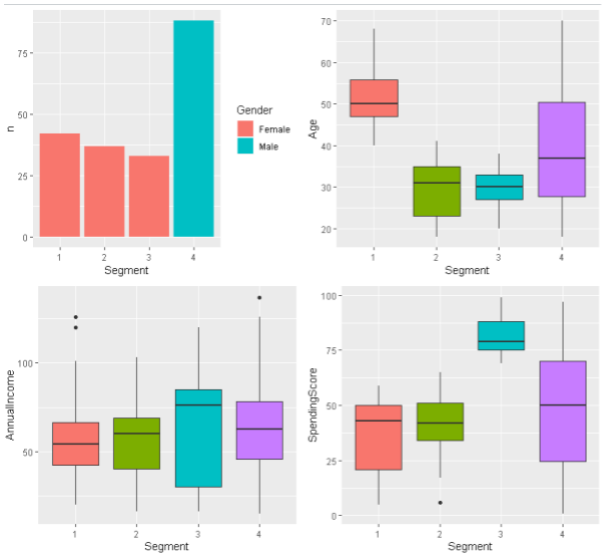
From the graphs below that show different number of clusters versus appropriate validation metrics, the optimal number of clusters can be obtained:



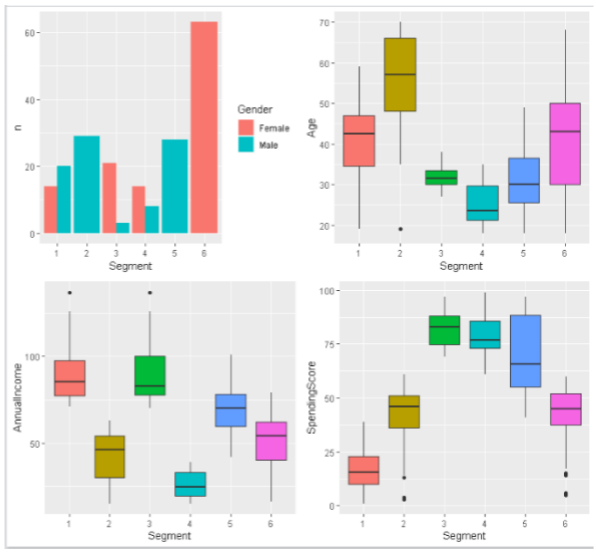
SFKM PAM k-prototypes

Thus, the optimal number of segments according to the SFKM method is 4, according to the PAM method is 5 and according to the k-prototypes method is 6.

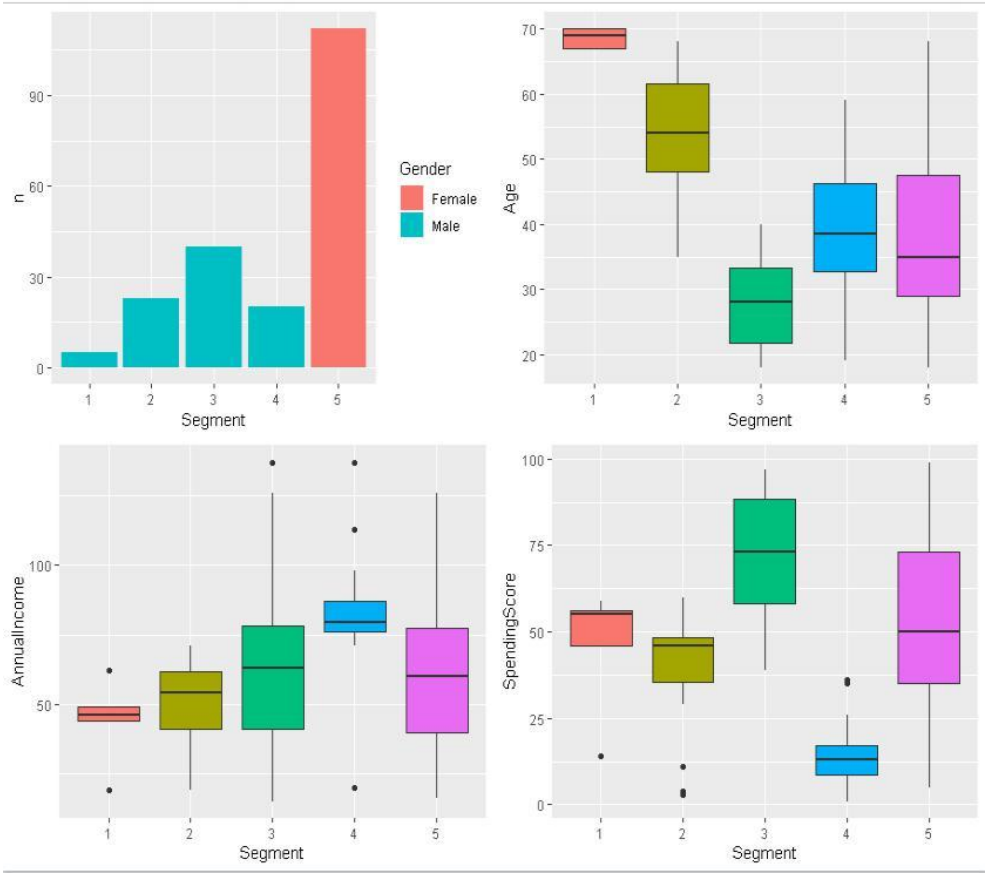
After performing all three methods, in order to compare how well they can separate customers into groups, box plots and histograms for each customer's feature for each segment have been created. The results are represented below:



SFKM



k-prototypes



PAM

From the obtained results, it can be seen that from the SFKM method the segmentation is not clear. PAM and k-prototypes methods have managed to group customers in a clearer way. However, comparing the last two methods, k-prototypes is more difficult for interpretation because of the large number of clusters. Additionally, in general, PAM is more preferable than k-prototypes (Budiaji and Leisch, 2019). Thus, as a base for interpretation, the PAM method has been chosen.

From the received segments it can be seen that:

1. Females belong to the segment 5, while males are distributed among the other 4 segments.
2. Ages are well separated for the first 4 segments: oldest group (about 67-70), relatively old group (about 48-62), youngest group (about 21-34) and matured group (about 34-46) respectively, whilst segment 5 contains all age groups for women customers.
3. As for annual income, there are some overlaps among segment 1 to 3, but the clear separated age and spending score groups help to distinguish these customers from others. Segment 4 contains customers with the highest annual incomes. Besides, segment 5 includes all women with all ranges of annual incomes.
4. The spending scores are also well separated for the first 4 segments. Customers have average spending scores about 55, 46, 73 and 13 respectively for segment 1 to 4. Segment 5 contains all spending score ranges for women as usual.

To sum up, it can be concluded that women will always be in segment 5 regardless of other characteristics, whereas men will be assigned to different segments depending mainly on age and spending score.

Project 2

Being able to identify fraudulent credit card transactions benefits both credit card companies and customers. Given historical records with clearly separated classes: fraudulent and non-fraudulent, inspirations could be derived from the characteristics of each class, therefore, contributions could be made to future fraud detection. Fraud detection can be made with the help of Supervised Machine Learning tasks and more specifically with different classification models.

Therefore, the goals of this project are to:

1. Explore the data set and gather important insights of it;
2. Taking the nature of data set into consideration, conduct different resampling processes, compare them and choose the best one;
3. Apply several classification methods, compare the results of each method, and obtain the one with the best performance.

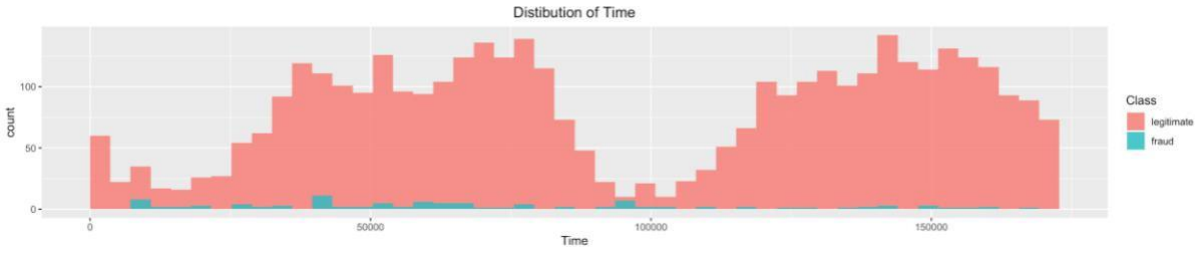
It should be mentioned that because of the large computation volume performed by created classification algorithms, this project takes quite a long time (approximately 30 minutes) to execute.

Data exploration

Before conducting any analysis, it is worthwhile to gather the basic sense of the data set.

The pie chart for class distribution shows that 97% of the transactions in this data set were legitimate while only 3% were fraudulent.

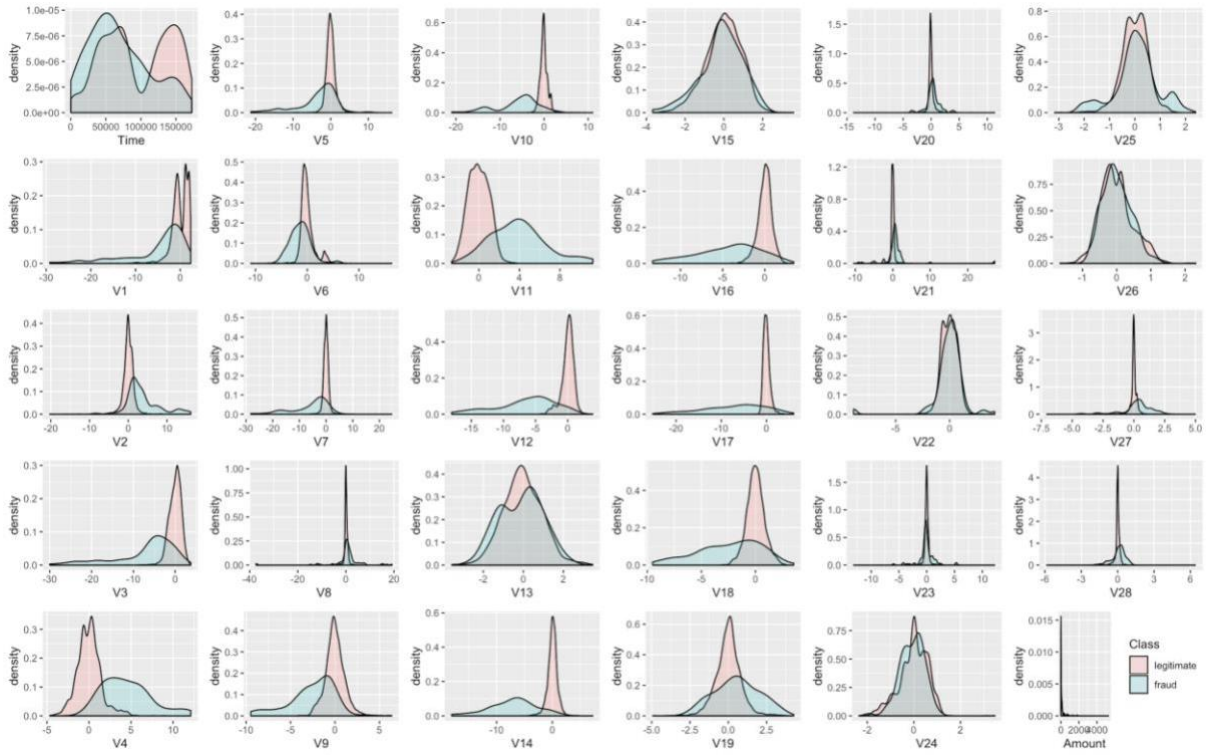
Since the majority of the predictor variables are anonymized, exploration of data will focus more on variables being clarified: time and amount.



The histogram of the distribution of time shows that a certain pattern can be observed with transaction time for legitimate transactions. Though it is unknown what time is the first transaction, according to the interval between the two lowest points (24 hours), it could be reasonable to assume that the drop in volume occurred

during the night. On the other hand, there is no obvious decrease for fraud transactions during the night time, sometimes even at a higher proportion than during the day time.

From the histogram of the distribution of amount above, it is clearly seen that the distribution of the amount of all transactions is heavily right-skewed. For both legitimate and fraudulent transactions, the vast majority of transactions are relatively small and only a tiny fraction of transactions is close to the maximum.



Features density plots were drawn here, in order to show the relationships between classes and each feature. For some of the features, it can be observed that a good selectivity in terms of distribution for the two values of Class: for example, V4 has quite clearly separated distributions for legitimate and fraud class, V12, V14, and V16 are partially separated, V1, V2, V3, V10 have a quite distinct profile, whilst V15, V22, V25 and V26 have similar profiles for the two values of Class.

In general, by exploratory data analysis, it can be concluded that:

1. creadFraud data set is highly imbalanced, therefore, re-sampling methods should be applied before classification for the purpose of creating a balanced training set;
2. In order to reduce the impact of variable variances on classification, data should be well scaled. As it can be seen that the variables V1 to V28 have already been scaled and had means that are very close to zero, while variable time and amount are not. Therefore, variables time and amount should be scaled before classification.

Classification

One of the goals of this question is to conduct different classification methods using a highly imbalanced dataset and compare their performances. This goal has been divided into two steps:

1. perform different resampling techniques for each classification method in order to receive balanced training set and compare their performance with each other as well as with the performance of algorithm without any resampling technique (using imbalanced training set);
2. perform different classification and compare their performances.

Methodologies

Due to the high imbalance of the whole dataset (97% legitimate / 3% fraud), the classification methods cannot be directly applied to the original train dataset. The direct classification methods application to the imbalanced dataset can result in a high accuracy of the legitimate class while a very low accuracy of the fraud class.
However, the focus is on the accuracy of the fraud class: the objective is to correctly classify the minority class of fraudulent transactions.

In order to achieve this goal, three special resampling techniques have been applied to train dataset before feeding it up to the classification methods. These techniques are random oversampling, random undersampling and synthetic minority oversampling (Kotsiantis et al., 2005)

Classification methods that have been applied are:

1. K-nearest neighbours (KNN)
2. Linear discriminant analysis (LDA)
3. Logistic regression
4. Decision Tree (DT)
5. Random Forest (RF)
6. Support Vector Machine (SVM) with two different kernels (linear and radial)

Each resampling method has been applied to both training sets (original and scaled) and then each classification method has been trained on resampled training sets (those methods that require scaled data on scaled training set, others on non-scaled). Parameters that are needed for particular methods have been tuned by cross-validation. Due to imbalance of data, the accuracy of methods has been measured using AUC values.

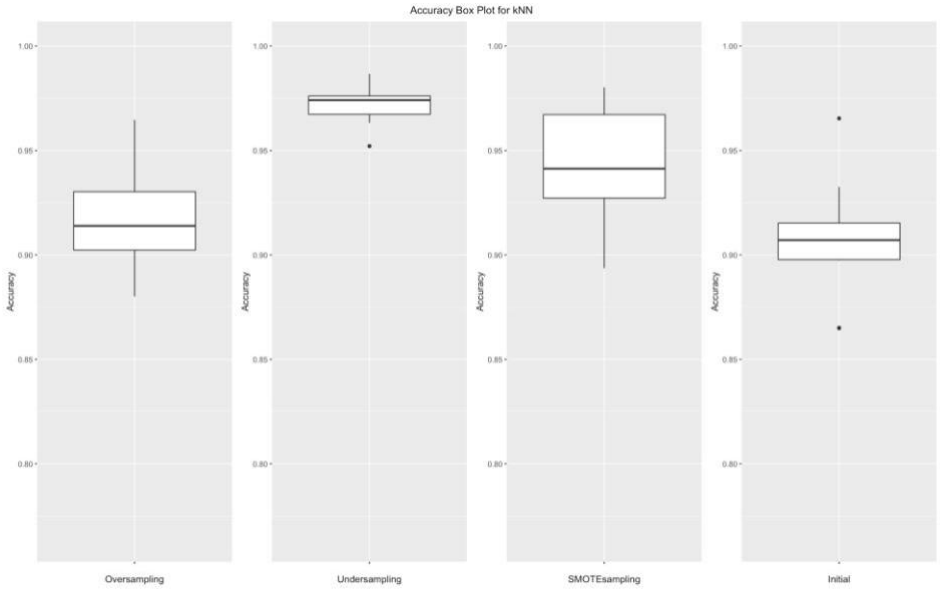
In order to choose the best resampling method for each classification method (the one that gives the highest AUC value), several train/test splits have been made. AUC values in each train/test split for each resampling method and each classification method have been recorded. Afterwards, for each classification method boxplots that demonstrate accuracies of each resampling method have been plotted. From these boxplots the best resampling method has been chosen then.

Final step was to compare the performance of classification methods between themselves. To do this, boxplots with AUC values for each method (with already chosen optimal resampling strategy) have been plotted. Then from these boxplots the best classification method has been chosen.

Results

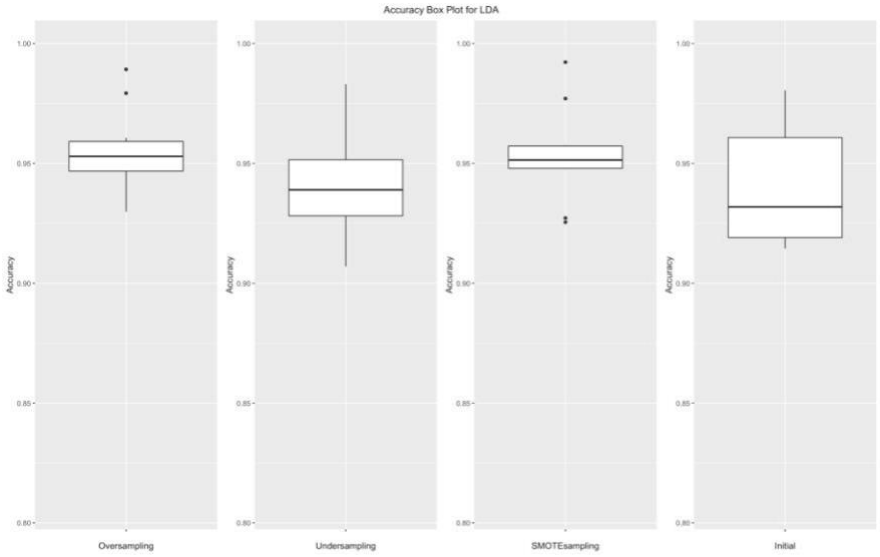
The results of resampling techniques accuracies for each method are represented below:

1. KNN



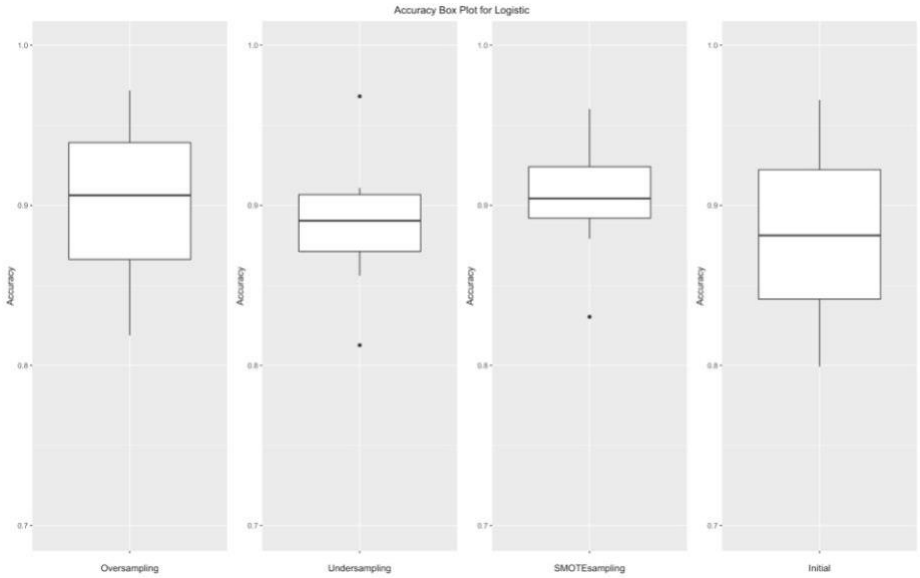
The best resampling technique is random undersampling (average AUC value is around 0.97).

2. LDA



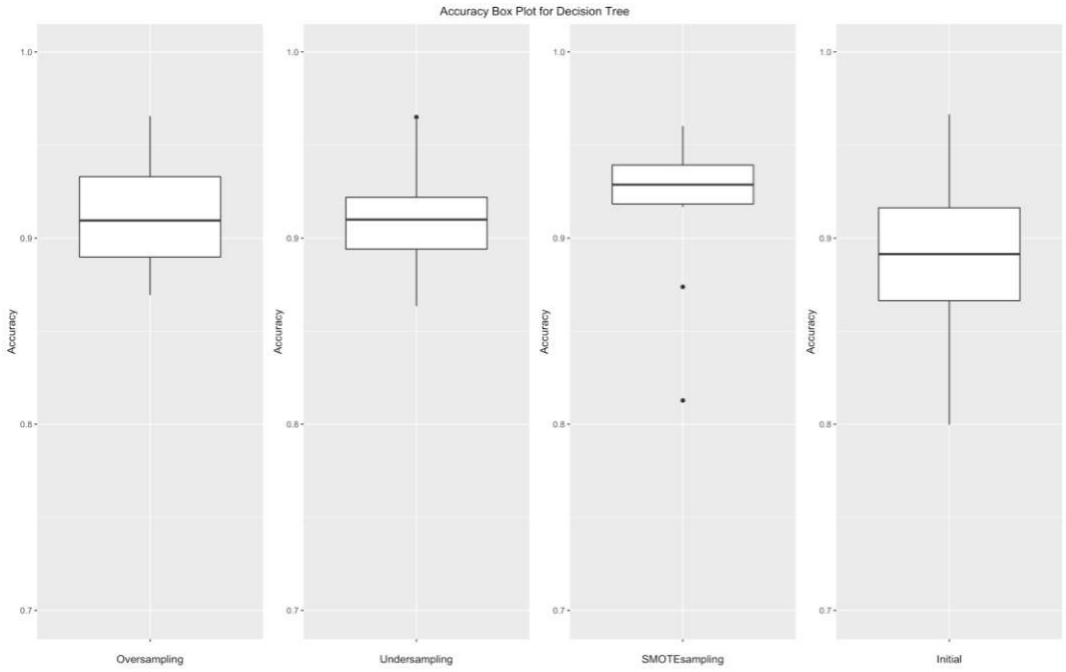
The best resampling technique is random oversampling (average AUC value is around 0.95).

3. Logistic regression



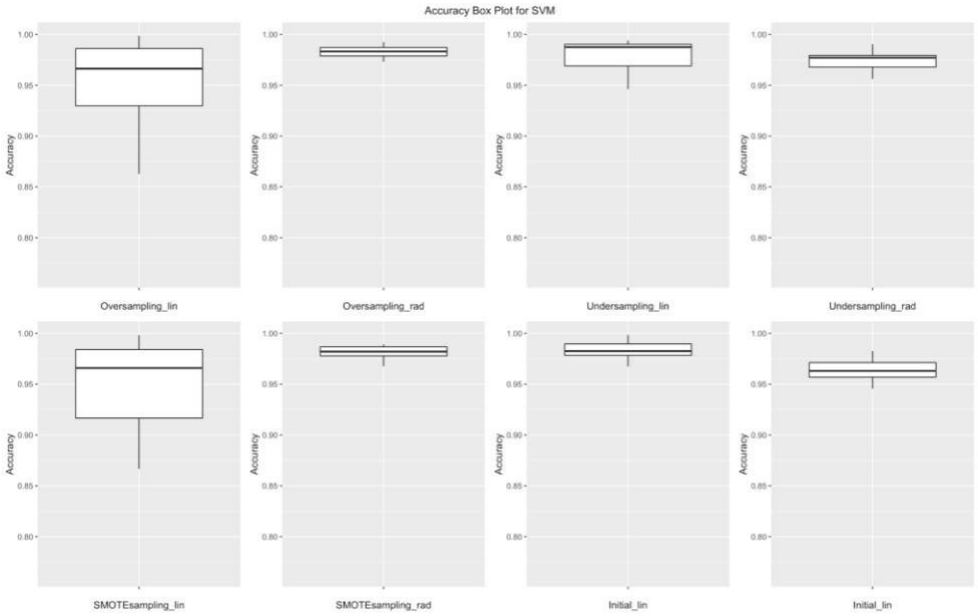
The best resampling technique is SMOTE (average AUC value is around 0.90).

4. DT



Best resampling technique is SMOTE (average AUC value is around 0.97).

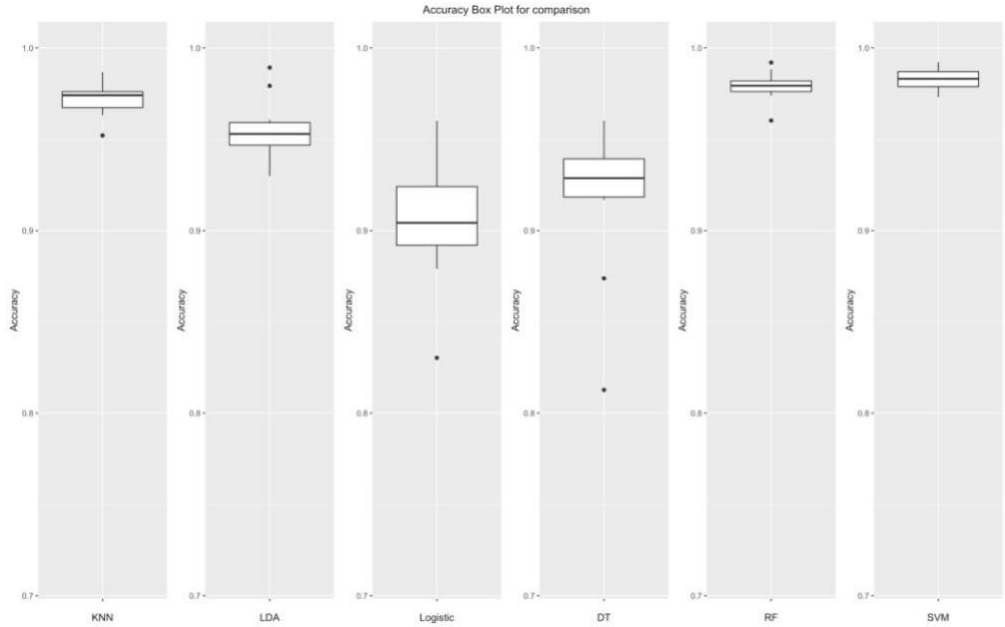
5. SVM with different kernels



The best resampling technique is random oversampling and the best kernel is radial (average AUC value is around 0.98).

Overall, it can be seen that applying particular oversampling techniques provides better performance than applying classification algorithms directly to the imbalanced data set.

The final plot that compares performances of classification methods with their optimal resampling strategy is represented below:



From this plot it can be seen that SVM with radial kernel and random oversampling technique for the originally imbalanced train data outperforms other methods. Its AUC values are very high with an average around 0.98. It also can be seen that RF and KNN with their sampling techniques (can be seen from box plots above) performs very well. The logistic regression with SMOTE technique for the originally imbalanced train data performs worst of all although its average accuracy is still high, around 0.9.

To sum up, overall all considered methods perform well with a given dataset. Depending on a purpose different methods can be chosen as the best ones: if the purpose is to achieve the highest accuracy, then it's SVM, if there is a business purpose for which the simplicity of interpretation is important, then it could be DT.

Bibliography

Budiaji, W. (2019). *'Kmed': Distance-Based K-Medoids*. R package

Budiaji, W., Leisch, F. (2019) Simple K-Medoids Partitioning Algorithm for Mixed Variable Data. Algorithms. [Online] 12 (9), 177. [online]. Available at: http://dx.doi.org/10.3390/a12090177.

Kaufman, L., Rousseeuw, P.J. (1987). 'Clustering by means of medoids', in: Dodge Y. (ed.) *Statistical data analysis based on the L1 norm and related methods*. North-Holland; Amsterdam pp. 405–416.

Kotsiantis, S., Kanellopoulos, D., Pintelas, P. (2005). Handling imbalanced datasets: A review. *GESTS International Transactions on Computer Science and Engineering*. 30. 25-36.

Martin, D. (2016). 'Clustering Mixed Data Types in R'. [Blog] *Wicked Good Data*, Available at: https://dpmartin42.github.io/posts/r/cluster-mixed-types#calculating-distance [Accessed 10 April 2020].

Park, H., Lee, J. and Jun, C. (2006). 'A K-means-like Algorithm for K-medoids Clustering and Its Performance'. *Department of Industrial and Management Engineering*, POSTECH.

Rousseeuw, P.J. (1987). 'Silhouettes: a Graphical Aid to the Interpretation and Validation of Cluster Analysis'. *Computational and Applied Mathematics*. 20, pp.53–65. Available at: 10.1016/0377-0427(87)90125-7.

Szepannek, G. (2018). 'clustMixType: User-Friendly Clustering of Mixed-Type Data in R', *The R Journal*, [online] 10(2), pp.200-208. Available at: https://doi.org/10.32614/RJ-2018-048 [Accessed 10 April 2020].

Szepannek, G., Aschenbruck, R. (2020). 'clustMixType': k-Prototypes Clustering for Mixed Variable-Type Data. R package version 0.2-2, CRAN. version 0.3.0, CRAN.