ITC 6003-Applied machine learning

- Final Project -

Petros Tamvakis (240760) - Vasileios Filipppidis (243989) -
Anastasios Katsaounis (244455)

March 29, 2020

PART B: Clustering

Cluster analysis groups data objects based only on information found in the data that describes the objects and their relationships. The goal is that the objects within a group be similar (or related) to one another and different from (or unrelated to) the objects in other groups. The greater the similarity (or homogeneity) within a group and the greater the difference between groups, the better or more distinct the clustering.

Cluster analysis is related to other techniques that are used to divide data objects into groups. For instance, clustering can be regarded as a form of classification in that it creates a labeling of objects with class (cluster) labels. However, it derives these labels only from the data. In contrast, classification is supervised classification; i.e., new, unlabeled objects are assigned a class label using a model developed from objects with known class labels. For this reason, cluster analysis is sometimes referred to as unsupervised classification.

In this part of the project the task is to perform clustering algorithms and evaluate their parameters. The assigned dataset was downloaded from `https://archive.ics.uci.edu/ml/datasets/Wholesale+customers` and refers to clients of a wholesale distributor. It includes 440 records describing the annual spending in monetary units (m.u.) on diverse product categories such as : Fresh, Milk, Grocery, Frozen, Detergents-Paper and Delicatessen.

1 Preparing the data

'Channel' and 'Region' categories were dropped as irrelevant and not contributing any information to the task at hand.

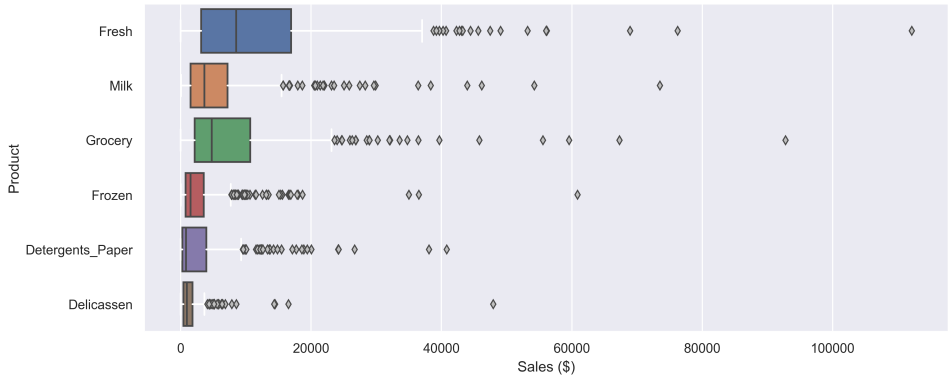

Figure 1: Annual spending distributions

Because of the annual spending's big distribution range, data were transformed to a logarithmic scale:

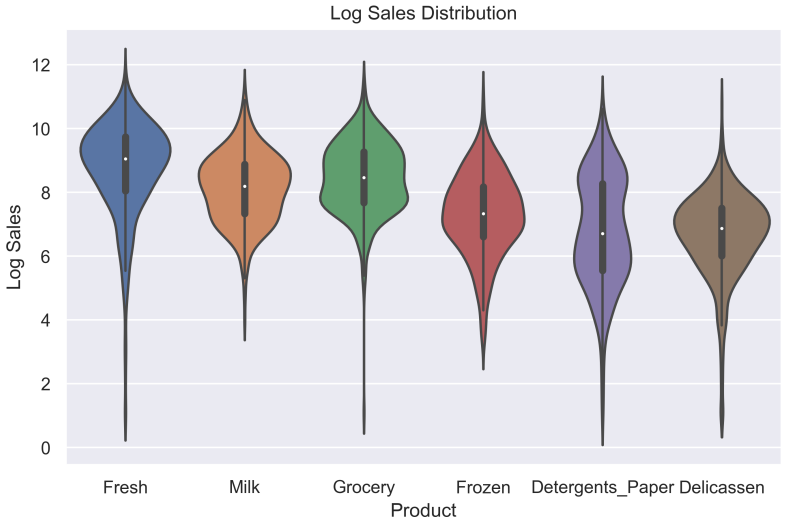

Figure 2: Scaled data

Both diagrams clearly show outliers in our data. To avoid infliction of a negative effect to the clustering techniques, outliers were removed using the Local Outlier Factor setting n-neighbors=20 and contamination factor as 0.05 resulting to a reduced dataset of 418 records.

2 Principal Component Analysis

After outlier removal a correlation heatmap was constructed to further explore the relationship between the products and potentially reduce data dimensions through PCA. From the heatmap it is clear that two major

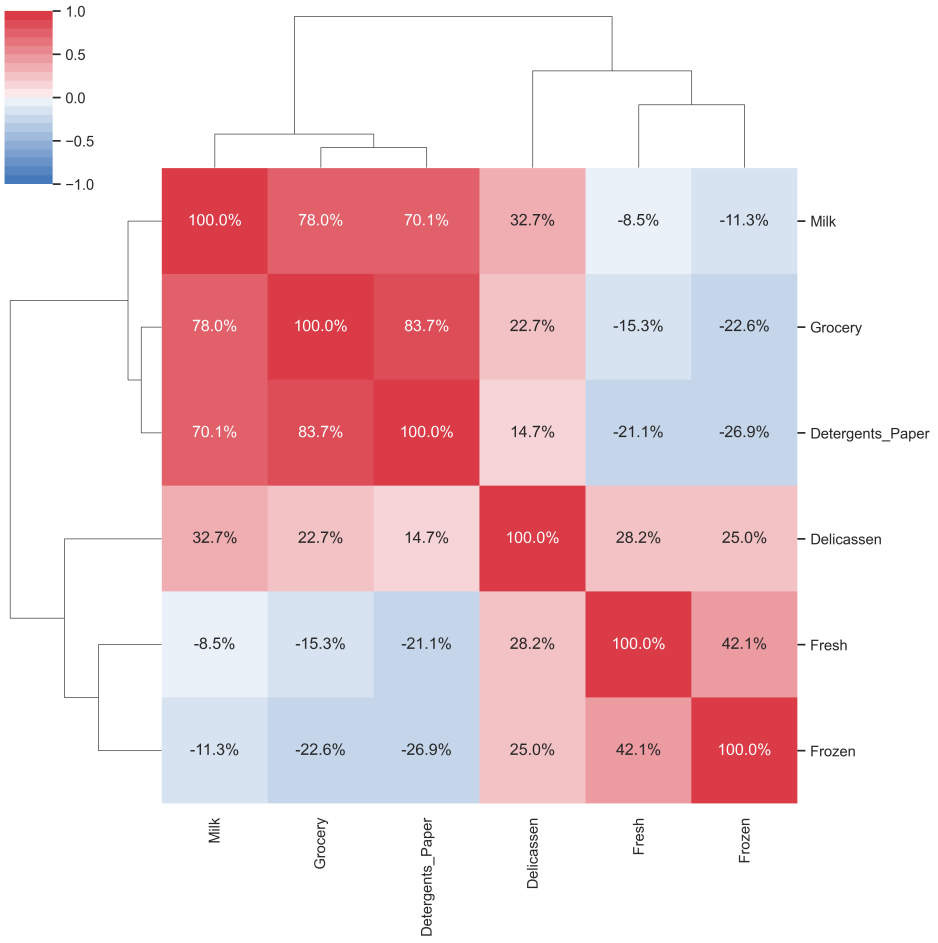

Figure 3: Correlation heatmap

groups (or clusters) exist in our data: the first group is: Milk, Grocery and Detergents-Paper, and the other : Frozen, Fresh and Delicatessen. Further analysis reveals that the two groups (components) capture 0.74% of total data variance directing to a dimensionality reduction, retaining the specific components.

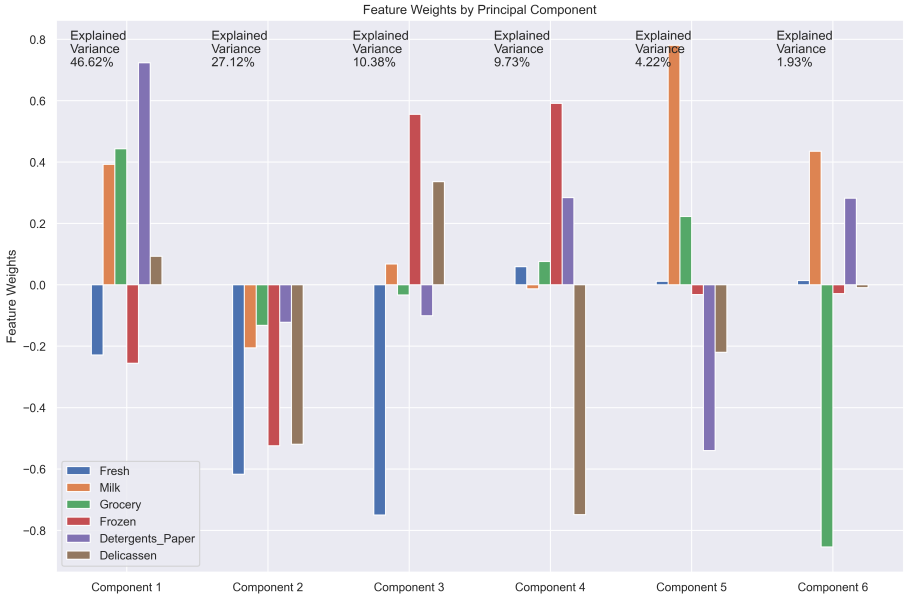

Figure 4: Feature weights

Below we present a joint plot of the data distribution (scatterplot and histograms) based on the new dimensions. On the diagram we can also see the initial component vector with respect to the new dimensions.

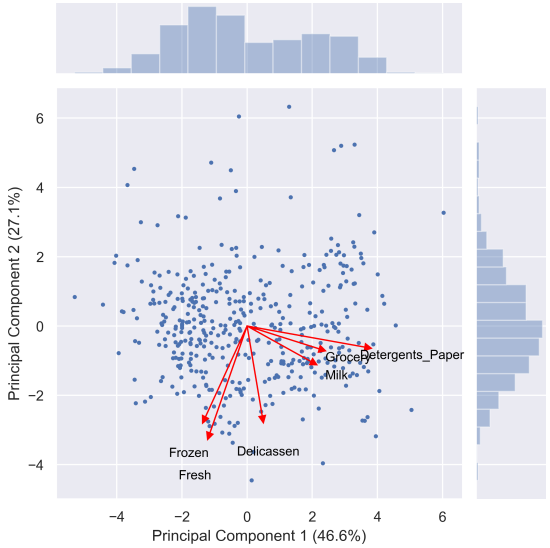

Figure 5: Feature weights

3 Clustering algorithms

3.1 K means

This is a prototype-based, partitional clustering technique that attempts to find a user-specified number of clusters (K), which are represented by their centroids. K-means defines a prototype in terms of a centroid, which is usually the mean of a group of points, and is typically applied to objects in a continuous n-dimensional space.

For research purposes K-means algorithm was tested on the reduced dataset for n=(2,11) clusters. Evaluation metrics (inertia,silhouette) were calculated and their respective diagrams along with cluster depictions and a result table for n=2,3,4,5 are presented below:

K means	**Clusters**			
	2	**3**	**4**	**5**
Inertia	1524.02	1137.13	869.31	710.44
Silhouette	0.436	0.407	0.344	0.365

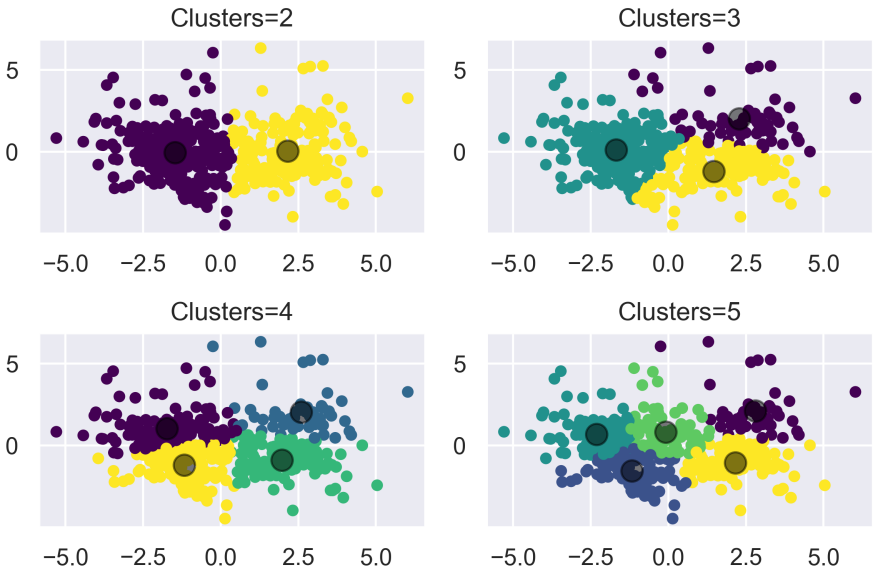

Figure 6: K-means

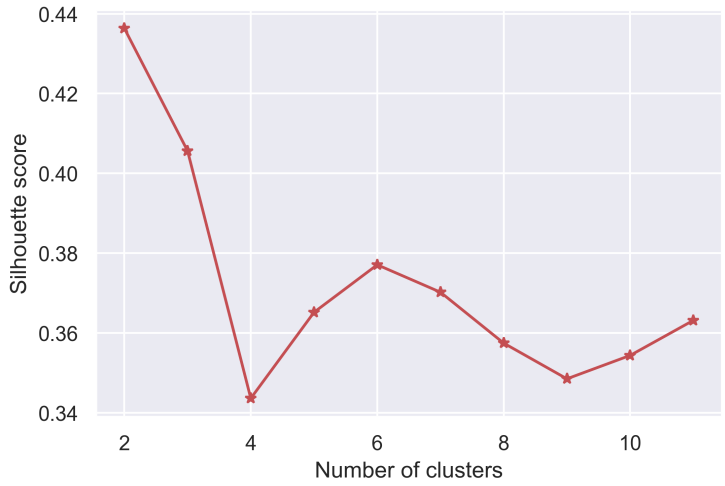

Figure 7: Silhouette

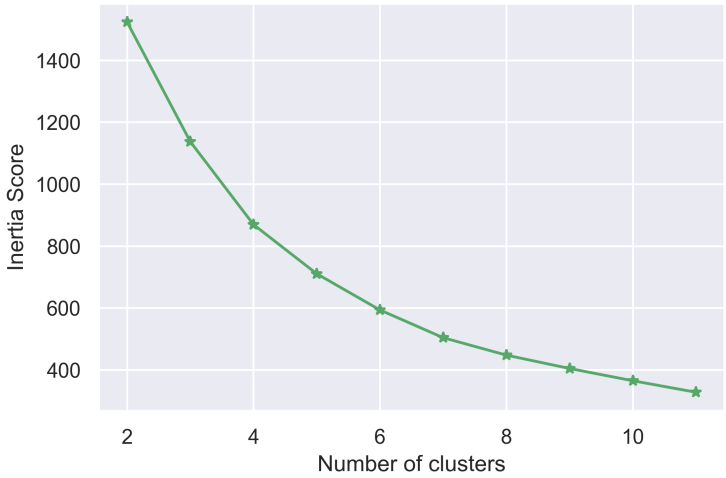

Figure 8: Inertia

A first remark is that the optimum number of clusters are 2 4. In that range silhouette is at an acceptable level while inertia is beginning to decrease. One has to keep in mund that the dataset has already been reduced by PCA (in basically two large clusters consisting of three products each).

3.2 Gaussian mixtures

A Gaussian mixture model (GMM) attempts to find a mixture of multi-dimensional Gaussian probability distributions that best model any input

dataset. In the simplest case, GMMs can be used for finding clusters in the same manner as k-means.

GMM algorithm was implemented to the reduced dataset for the same number of clusters (n=2,11) and the results for the log-likelihood metric are summarized in the table below. It can be deducted that an optimum number of clusters is 2 5.

GMM

Clusters	2	3	4	5	6
log-likelihood	**-3.946**	**-3.943**	**-3.898**	**-3.889**	**-3.866**
Clusters	7	8	9	10	11
log-likelihood	**-3.86**	**-3.823**	**-3.831**	**-3.823**	**-3.799**

3.3 DBSCAN

This is a density-based clustering algorithm that produces a partitional clustering, in which the number of clusters is automatically determined by the algorithm. Points in low-density regions are classified as noise and omitted; thus, DBSCAN does not produce a complete clustering.

In DBSCAN the parameters that have to be considered are the maximum radius of a cluster (eps) and the minimum number of points that must be included in the cluster (min-samples). Adjusting these parameters the conclusion was reached that based on the silhouette criterion the optimum number of clusters for DBSCAN algorithm is 1 or 2.

DBSCAN	*eps=1,min=20*	*eps=0.5,min=5*
	cluster=1	**clusters=2**
Silhouette	0.418	0.255

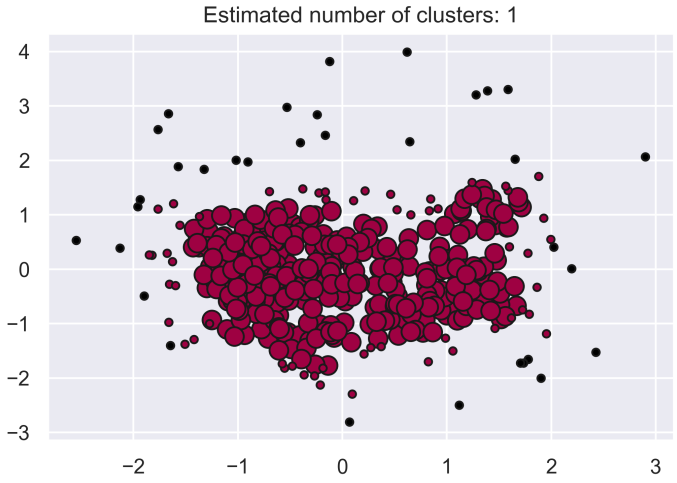

Figure 9: DBSCAN 1 cluster

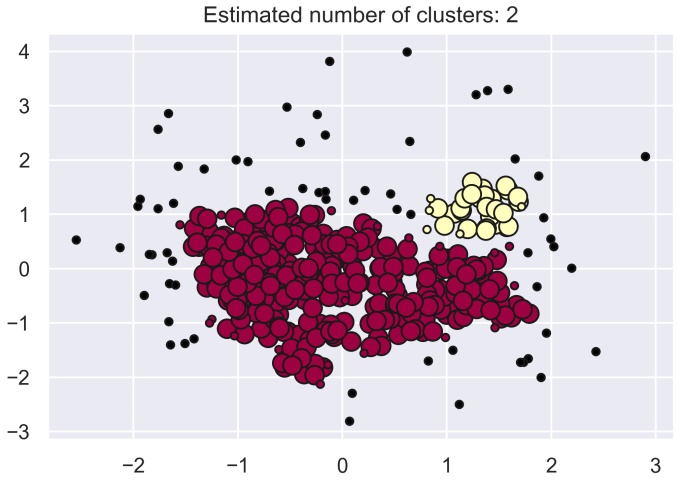

Figure 10: DBSCAN 2 clusters

Extensive analysis showed that for the majority of eps and min-sample combinations, DBSCAN clustered the samples in one big cluster. Only for very small cluster radius and minimum samples did the algorithm yield two clusters distinguishing DBSCAN from the other clustering techniques.

PART D: Predicting buys

In part D the goal is to predict whether a user will buy a product or not based on his/her online behavior, and in particular his/her clicks during a session. There are two datases which were downloaded from `https://2015.recsyschallenge.com/challenge.html`.

yoochoose-clicks.dat: Click events. Each record/line in the file has the following fields:

- Session ID – the id of the session. In one session there are one or many clicks.
- Timestamp – the time when the click occurred.
- Item ID – the unique identifier of the item.
- Category – the category of the item.

yoochoose-buys.dat: Buy events. Each record/line in the file has the following fields:

- Session ID – the id of the session. In one session there are one or many clicks.
- Timestamp – the time when the click occurred.
- Item ID – the unique identifier of the item.
- Price – item's price
- Quantity – how many items were bought.

Tasks to perform are:

- Build a data set that can be used in classifier to decide whether someone will buy or not.
- Preprocess the data and perform classification

1 Preparing the data

Perhaps the most challenging part was to prepare the two datasets. Both sets combined occupied memory size of 3GB and consisted of about 34 million rows. Decision was made to drop 'Category' and 'Price' columns based on the assumption that they play no major role to the number of clicks or the decision to buy or not.

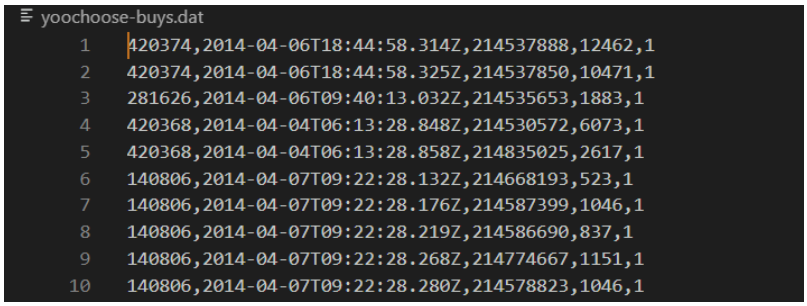

Figure 11: CSV format

The datasets initially needed merging because they literally are about the same sessions. Because every row in the datasets was basically a click, aggregation was performed to the merged set for each session in order to calculate each session's clicks which was essential for performing classification. Clicks were saved as a new column. In this way the two datasets were merged into a reduced one of about 10 million rows(unique sessions). Timestamp feature was parsed using python language datetime library and transformed into two new columns:

- Weekday - day of week the session commenced
- Hour - hour of day the session commenced

Furthermore a new column was created containing the number of unique items clicked per session in order to investigate the role of this feature in buying or not.

	SessionId	Clicks	clickedItems	Weekday	Hour	Purchase
0	1	4	4	0	10	0
1	2	6	5	0	13	0
2	3	3	3	2	13	0
3	4	2	2	0	12	0
4	6	2	2	6	16	0

Figure 12: Final dataframe

Analyzing the newly formed dataset we were able to calculate the buying average per weekday and per hour of day. It is clear from the

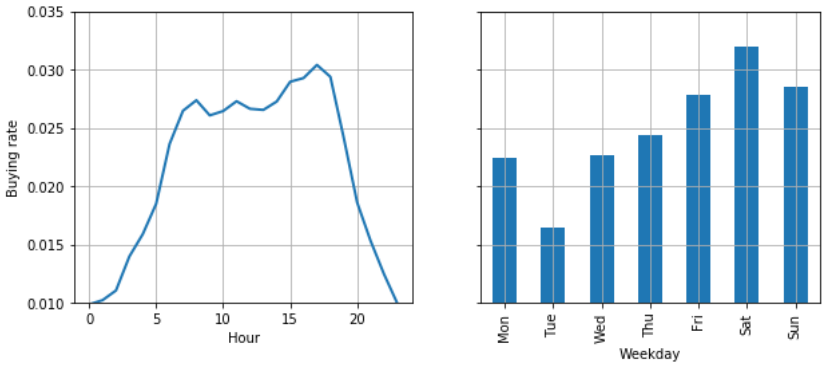

Figure 13: Buying rate

diagrams that people prefer to make purchases mostly on the weekends. Also buying activity reaches its peak in noon and afternoon hours.

2 Classification

After data preprocessing and preparation the dataset was split in train and test sets (50%) in order to run classification techniques and evaluate their performance on correctly predicting buys based on the amount of a session's clicks.

Classifiers chosen were a Naive Bayes and a Decision Tree classifier (max depth=10). Attempts with other classifiers (K nearest neighbors, Neural Networks) were made but memory and time consumption was

huge and were finally aborted.

Evaluation metrics are presented below:

	Evaluation metrics		
	Macro Precision	**Recall**	**F1 score**
Decision Tree	0.718	0.500	0.494
Naive Bayes	0.566	0.575	0.570

In general both techniques performed adequately and were on par with each other. Decision tree yielded a Macro precision while Naive Bayes came on top in Recall and F1 score.

Concerning AUC Decision tree was a clear winner with a result of 0.933 while Naive Bayes covered an AUC of 0.76.

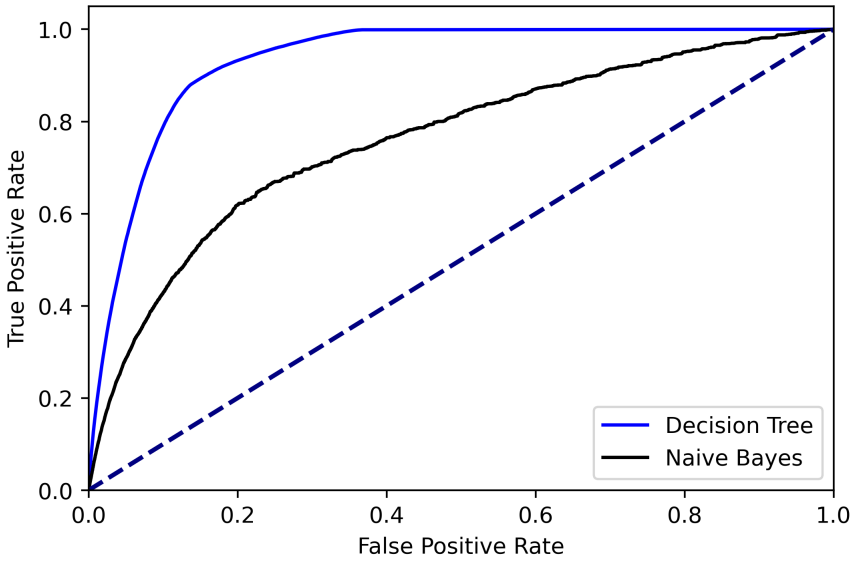

Figure 14: ROC curve