ITC 6003-Applied machine learning

- Final Project -

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

April 4, 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.

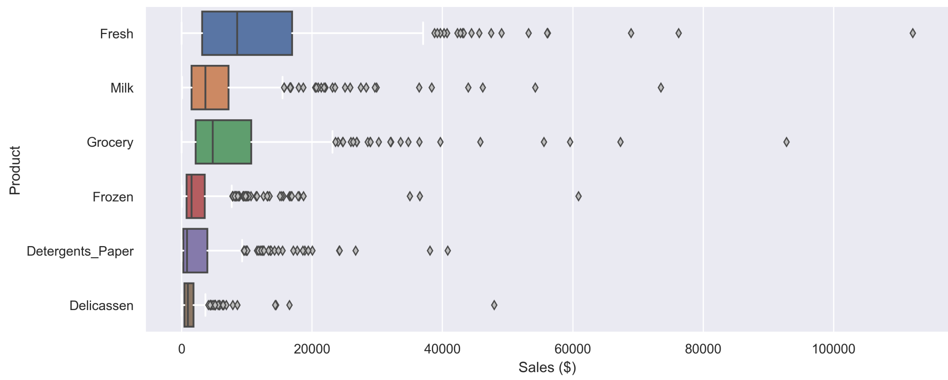


Figure 1: Annual spending distributions

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

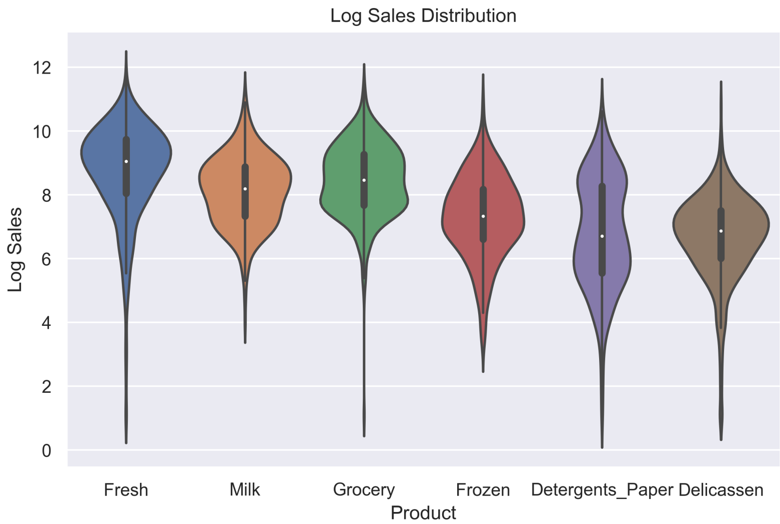


Figure 2: Scaled data

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

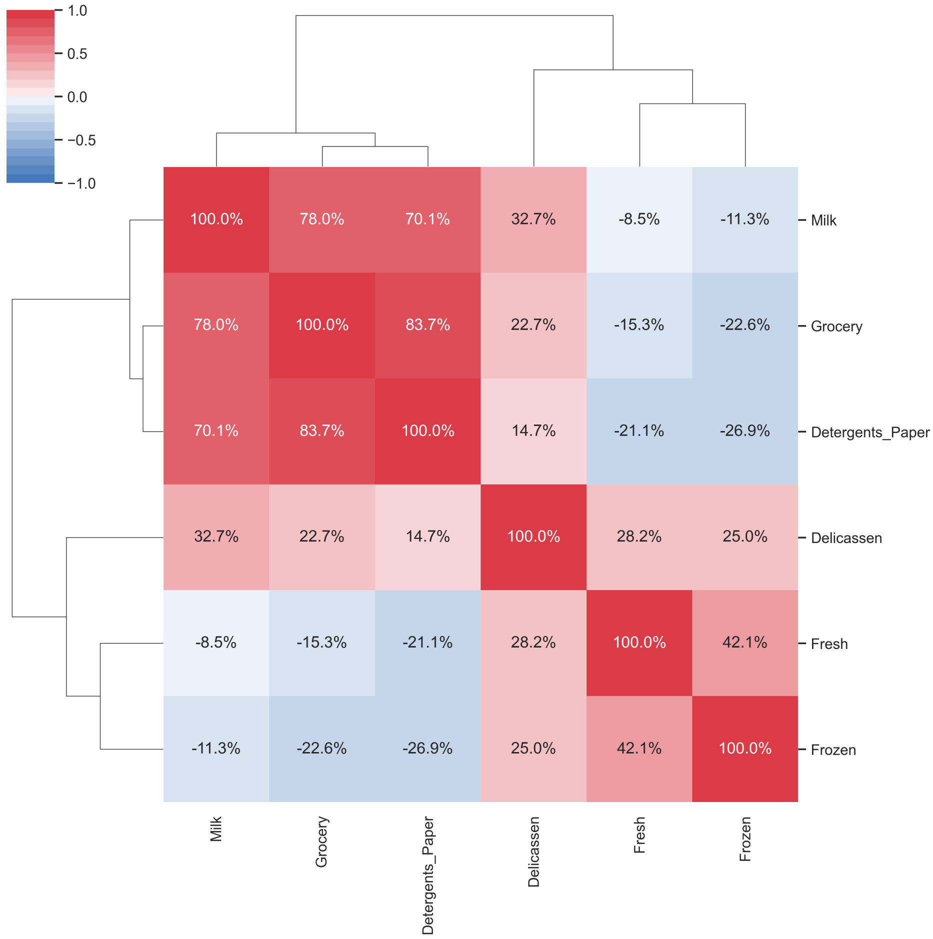


Figure 3: Correlation heatmap

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.

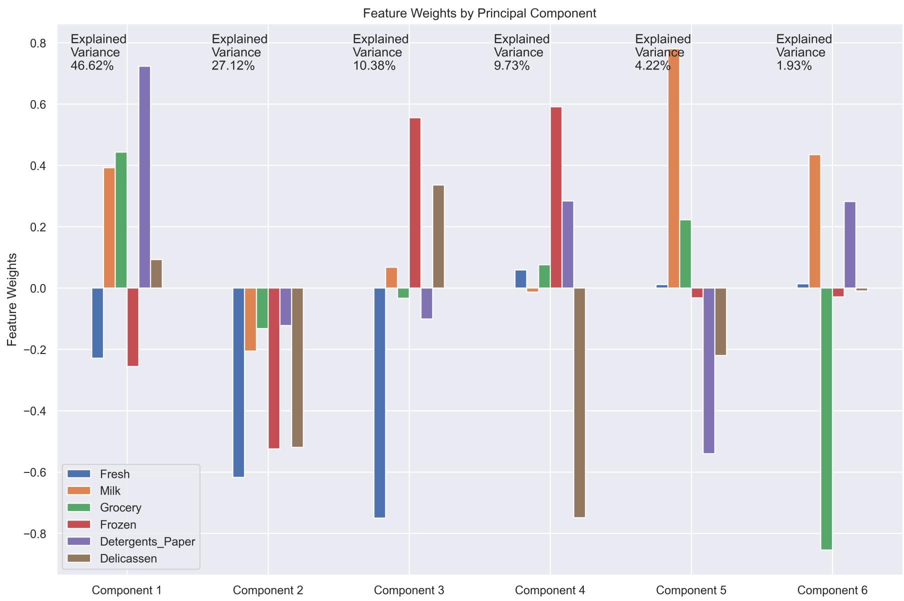


Figure 4: Feature weights

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.

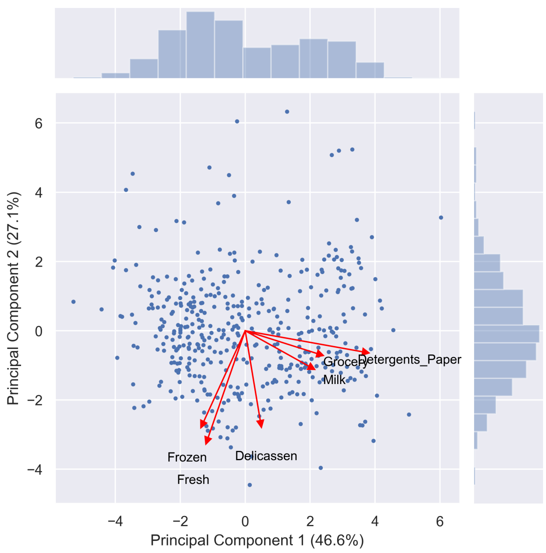


Figure 5: Feature weights

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=4,5,6,7 are presented below:

K means	**Clusters**			
	4	**5**	**6**	**7**
Inertia	869.25	710.43	593.83	503.99
Silhouette	0.344	0.365	0.378	0.370

Figure 6: K-means

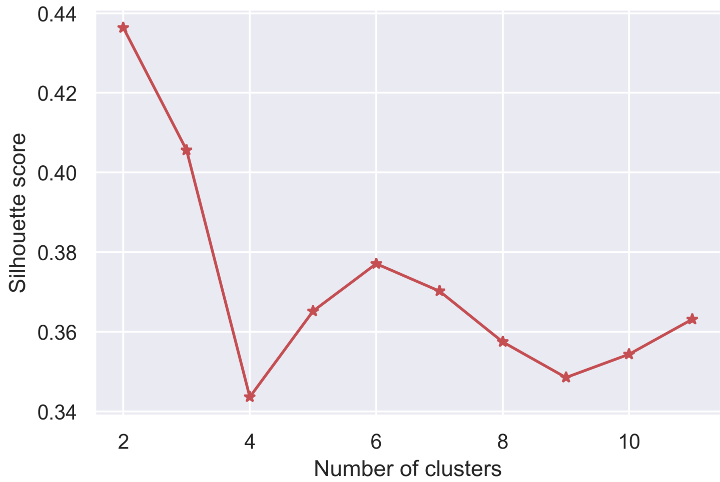


Figure 7: Silhouette

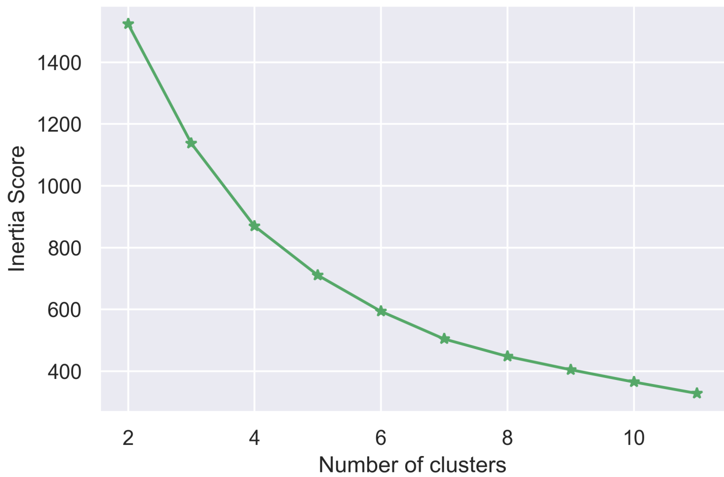


Figure 8: Inertia

From the inertia "elbow" diagram a remark to be made is that the optimum number of clusters for our analysis is 6. For 6 clusters the slope of the diagram starts to decrease. Also the value of silhouette is higher than 5 or 7 clusters and beyond. One has to keep in mind that the dataset has already been reduced by PCA to only two components.

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

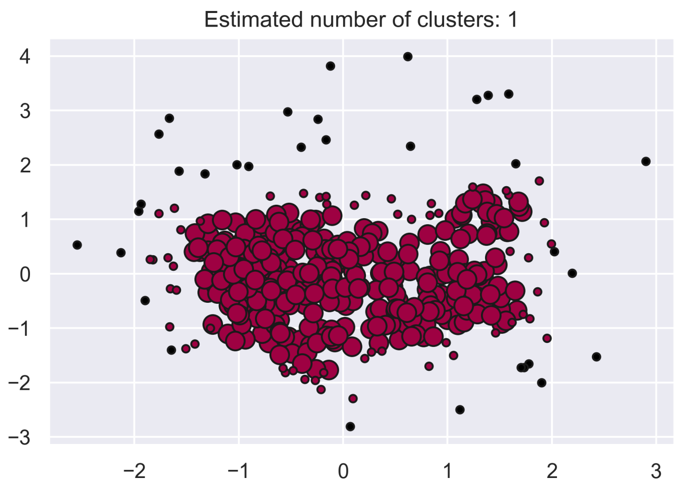


Figure 9: DBSCAN 1 cluster

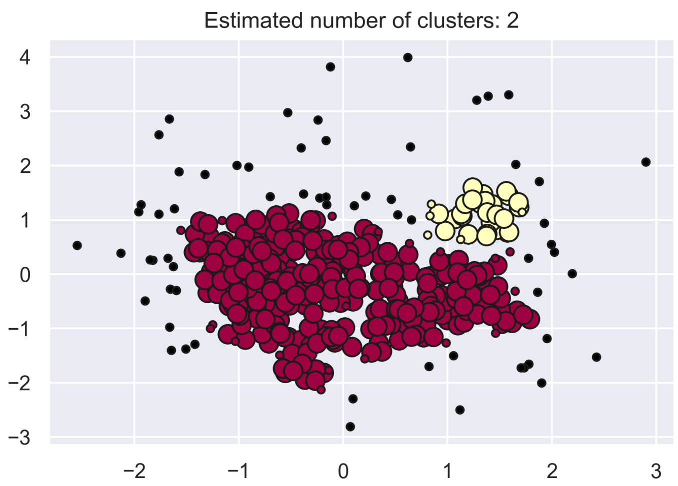


Figure 10: DBSCAN 2 clusters

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.

4 Performing clustering to the non-reduced dataset

A question that arises is: had we kept the dataset as a whole and its features hadn't been reduced would the clustering algorithms perform better or worst? All three algorithms were applied again this time to the whole dataset keeping all six original features.Results are summarized below.

4.1 K means

K means algorithm yielded almost identical results for the whole dataset resulting to 6 clusters as an optimum number for the analysis.

K means	**Clusters**			
	4	**5**	**6**	**7**
Inertia	1867.91	1703.90	1562.00	1463.08
Silhouette	0.211	0.209	0.218	0.215

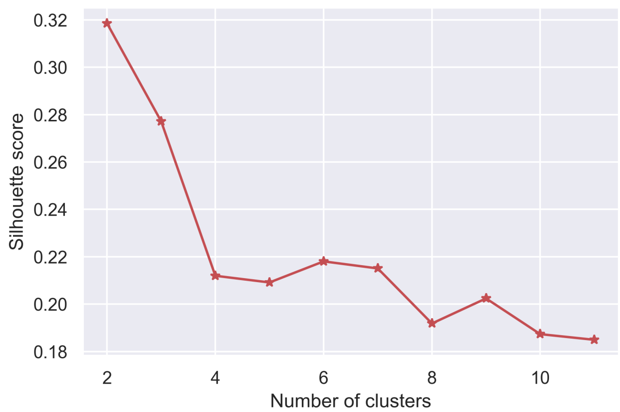


Figure 11: Silhouette

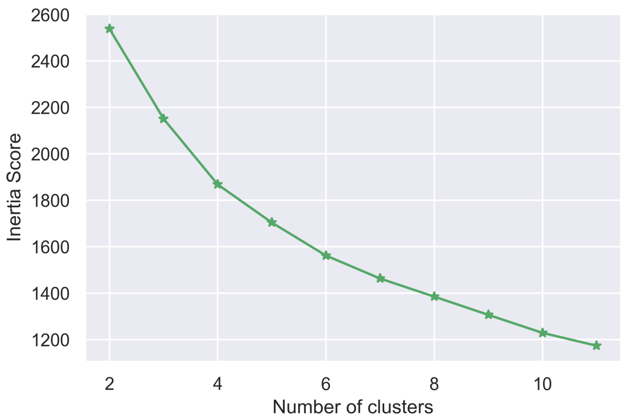


Figure 12: Inertia

4.2 Gaussian mixtures

GMM algorithm implementation:

GMM

Clusters	2	3	4	5	6
log-likelihood	**-7.99**	**-7.74**	**-7.625**	**-7.582**	**-7.422**
Clusters	7	8	9	10	11
log-likelihood	**-7.387**	**-7.328**	**-7.255**	**-7.212**	**-7.101**

4.3 DBSCAN

The majority of eps and min-samples combinations resulted to the same result: DBSCAN clustered the samples to one big cluster.

DBSCAN	*eps=1,min=10*	*eps=1.2,min=10*
	cluster=1	**clusters=2**
Silhouette	0.32	0.079

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:

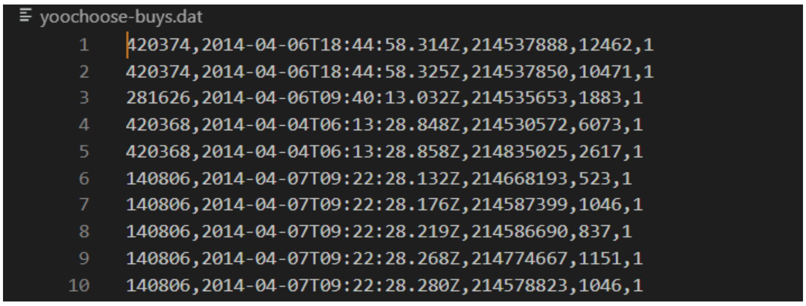
- Session ID – the id of the session. In one session there are one or many clicks.
- Timestamp – the time when the click occurred.
- Item ID – the unique identifier of the item.
- Category – the category of the item.

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



```
yoochoose-buys.dat
420374,2014-04-06T18:44:58.314Z,214537888,12462,1
420374,2014-04-06T18:44:58.325Z,214537850,10471,1
281626,2014-04-06T09:40:13.032Z,214535653,1883,1
420368,2014-04-04T06:13:28.848Z,214530572,6073,1
420368,2014-04-04T06:13:28.858Z,214835025,2617,1
140806,2014-04-07T09:22:28.132Z,214668193,523,1
140806,2014-04-07T09:22:28.176Z,214587399,1046,1
140806,2014-04-07T09:22:28.219Z,214586690,837,1
140806,2014-04-07T09:22:28.268Z,214774667,1151,1
140806,2014-04-07T09:22:28.280Z,214578823,1046,1
```

Figure 13: CSV format

1 Preprocess 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.

To preprocess the data and in order for the dataset to reach its final form the following steps were necessary:

1. **Dataset merging:** The two datasets were appended and indexed correctly. NaN values were replaced with zeros.

2. **Parse Timestamp column:** Timestamp feature was parsed and transformed to python datetime object.

3. **Unique items in session:** Dataset grouped-by sessions and number of unique items clicked per session was saved as a new column

4. **Aggregation:** Dataset grouped by sessions,column "Clicks" created (total number of clicks in session),column "Purchase" created. Columns "sessionStart" and "sessionEnd" created.

5. **Date-Time columns:** Weekday and hour the session commenced were extracted from the "sessionStart" using a custom function that was applied to the datasets. Furthermore the duration of session was calculated by subtracting session's "startTime" from session's

"endTime" and recalculated into seconds.

6. **Class column:** "Purchase" column was transformed to a binary class feature: no-buy:0,buy:1.

7. **Feature drop:** unecessary/no-interest features dropped.

	SessionId	Clicks	clickedItems	Weekday	Hour	Duration	Purchase
0	1	4	4	0	10	351.029	0
1	2	6	5	0	13	359.275	0
2	3	3	3	2	13	745.378	0
3	4	2	2	0	12	1034.468	0
4	6	2	2	6	16	246.128	0

Figure 14: Final dataframe

Analyzing the newly formed dataset we were able to calculate the buying average per weekday, per number of clicks and per hour of day.

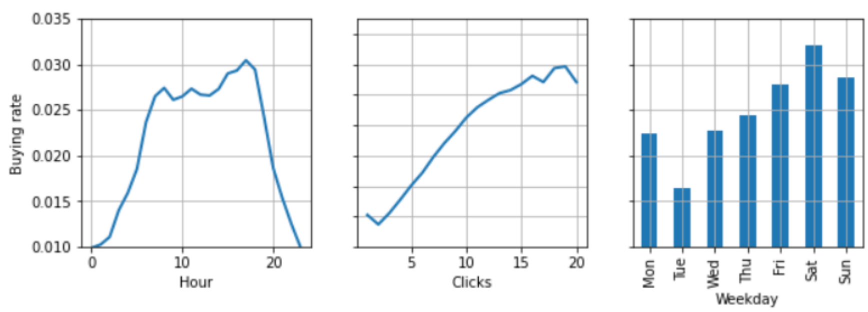


Figure 15: Buying rate

It is clear from the diagrams that people prefer to make purchases mostly on the weekends. Buying activity reaches its peak in noon and afternoon hours. Lastly more-clicks-sessions seem to end up in buys more than less-click sessions.

2 Prepare the data

After preprocessing the dataset was prepared for classification analysis. This was a highly imbalanced set with a predominant no-buy class (buy to no-buy ratio: 0.261%). Decision was made to downsample the no-buy class to produce a balanced set of equal sized classes. Normalization was also performed (mean=0, std=1) to all the features for classifier optimization.

3 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. Classifiers chosen were:

- **Naive Bayes**
- **Decision Tree classifier** (max depth=10)
- **Neural Network** (one hidden layer/5 nodes, max-iterations:20)

Evaluation metrics are presented below:

	Evaluation metrics		
	Macro Precision	**Recall**	**F1 score**
Decision Tree	0.899	0.894	0.894
Naive Bayes	0.714	0.660	0.638
ANN	0.897	0.893	0.893

In general all techniques performed adequately. Decision tree and ANN were on par with each other resulting in outstanding predictive performance.

ROC curves are plotted for comparison against each other on the same plot depicting predictive accuracy of each classifier:

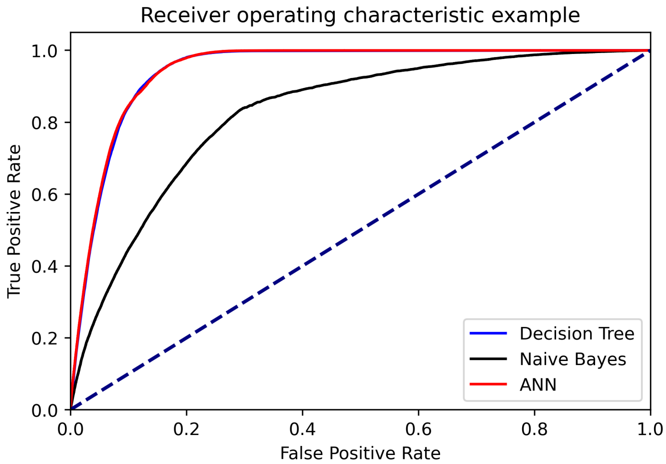


Figure 16: ROC curve

Future work includes building an SVM classifier for imbalanced classes or a k-fold validation technique to run on the whole dataset in order to take advantage of all data information.