Scuola Normale Superiore

Classe di Scienze

Telco Costumer Churn analysis

Intorduction to Machine Learning

Sunday 11th May, 2025

Author: Giulio Cordova

giulio.cordova@sns.it

Contents

1 Data Understanding & Preparation

The *Telco Customer Churn* dataset includes 7043 customer records and 21 variables, covering demographics, service subscriptions, contract details, and churn status (i.e., whether the customer left in the last month). The features can be grouped into four categories:

- **Demographics:** `gender`, `SeniorCitizen`, `Partner`, and `Dependents`. Most are categorical (`Yes`/`No`); `SeniorCitizen` is binary (0/1).
- **Contract Details:** `tenure` (subscription duration), `Contract`, `PaperlessBilling`, and `PaymentMethod`. `Contract` and `PaymentMethod` are multi-category variables, while `PaperlessBilling` is boolean.
- **Subscribed Services:** Indicators for phone, internet, and add-ons (e.g., `OnlineSecurity`, `StreamingTV`). Values like `No internet service` encode redundant information.
- **Target:** `Churn`, a binary categorical variable indicating customer loss.

After loading the CSV file, columns and data types were reviewed. `TotalCharges`, although numerical, was read as an object due to malformed entries. After converting this feature, 11 missing values were found and dropped given the small amount. Additionally, 22 duplicated rows were found and removed.

Many categorical features in the dataset are binary in nature and were therefore converted to boolean variables. A dedicated routine was implemented to automate this process and ensure consistency. In particular, the variables `PhoneService`, `MultipleLines`, `OnlineSecurity`, `OnlineBackup`, `TechSupport`, `StreamingTV`, `StreamingMovies`, and `DeviceProtection` were converted into boolean form. For these, the special values `No phone service` or `No internet service` were interpreted as `False`, since the presence of phone or internet service is already captured by other variables. The variable `gender` was transformed into a new binary feature named `IsMale`, equal to `True` if the customer is male, and `False` otherwise. Multi-category features like `Contract`, `InternetService`, and `PaymentMethod` were one-hot encoded.

Numerical variables were scaled to [0, 1] using MinMaxScaler. This normalization was important for correlation analysis, dimensionality reduction, and clustering.

Correlation analysis was conducted using Pearson correlation. The results confirmed intuitive relationships: `TotalCharges` is positively correlated with both `tenure` and `MonthlyCharges`; customers with additional services such as `StreamingTV`, `StreamingMovies`, or `Fiber optic` tend to pay higher monthly charges. Regarding the target variable `Churn`, it showed positive correlation with `Month-to-month` contracts, `Fiber optic` internet, and `Electronic check` payments, while it was negatively correlated with `tenure` and `Two year` contracts.

As a final step, Principal Component Analysis (PCA) was applied to the standardized numerical dataset to investigate its internal structure and to reduce its dimensionality. Importantly, the target variable `Churn` was excluded from this analysis, as this dataset will be used for clustering analysis and a goal of clustering is to uncover natural groupings in the data independent of the known labels. The PCA revealed that the first 21 components accounted for more than 99% of the total variance, indicating that the remaining components contributed negligibly to the overall information content. This strong compression is largely due to redundancy among features—`TotalCharges`, for example, is nearly a linear combination of `tenure` and `MonthlyCharges`. Based on this insight, the dataset was projected onto the first 21 principal components and this reduced representation was saved for use in subsequent clustering experiments.

2 Clustering

The dataset used for clustering is the PCA-transformed version described earlier, which retains 21 principal components to reduce dimensionality and eliminate redundancy among the original features.

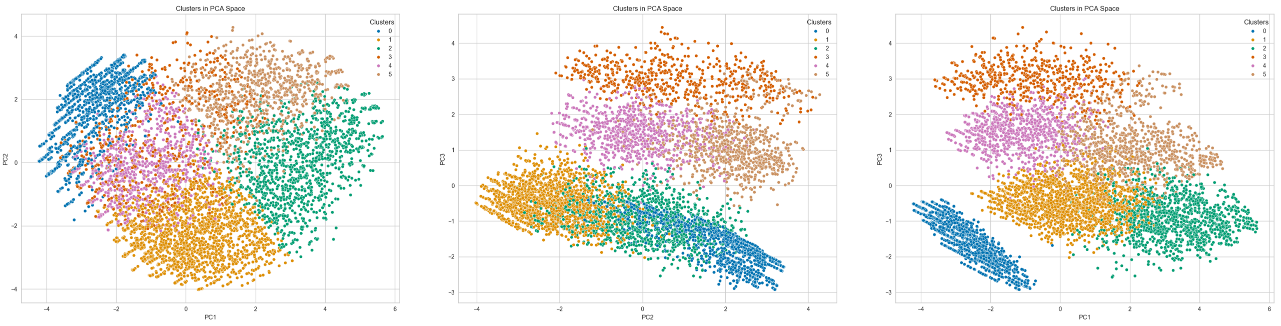
In this representation, all variables are continuous numerical values, and no longer boolean or categorical.

Three different clustering algorithms were applied: K-Means, DBSCAN, and Hierarchical Clustering.

2.1 K-Means

To determine the optimal number of clusters K, the K-Means algorithm was run for values of K ranging from 2 to 50. The Elbow Method was then applied, revealing a clear inflection point at $K = 6$, which was chosen as the final number of clusters.

Clusters were visualized through pairwise projections of the first three principal components, with a 3D scatter plot of the first three PCs also available in the notebook.



(a) Projection in PC1-PC2 Plane (b) Projection in PC2-PC3 Plane (c) Projection in PC1-PC3 Plane

Figure 1: Color-coded clusters found with 6-Means clustering in the PCA space

For each cluster, I analysed the distribution of the original features and produced summary bar plots like the one shown in Figure 2a. In these plots, features are ordered by the variance of their mean values across all clusters, helping to highlight which variables are most informative in differentiating between groups. The goal of this analysis was to identify natural profiles of customers represented by each cluster. For example:

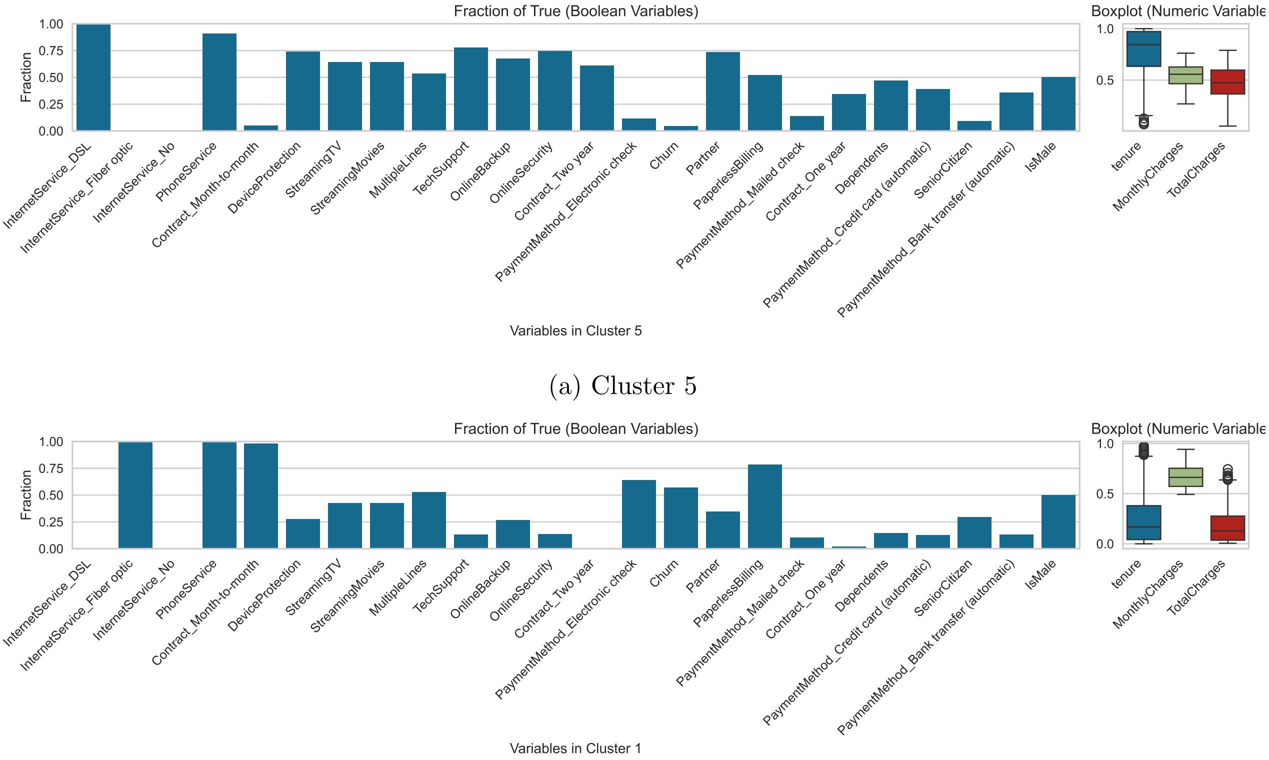
- **Cluster 5** contains users with long tenure, partners and dependents, and a wide range of internet services such as streaming or tech support. The churn rate in this group is low, suggesting that it may represent satisfied families with well-tailored contracts.
- **Cluster 1** includes users with high monthly charges and short tenure. These users often have fiber optic internet and additional services, but also a higher proportion of senior citizens. This may suggest customers who are paying for more than they need, leading to a higher churn rate.

Similar reasoning could be applied to other clusters, each potentially representing a different customer archetype.

2.2 DBSCAN

To determine an appropriate value for the ϵ parameter in DBSCAN, I performed a K-nearest-neighbor (KNN) analysis with $K = 5$. The resulting distance plot (in appendix) suggested the presence of multiple possible density thresholds, indicating that DBSCAN might struggle with this dataset due to the presence of clusters with varying densities.

Nonetheless, I attempted clustering with several ϵ values. The best configuration identified approximately 10 clusters, although only 45% of the data points were classified as belonging to a cluster—the rest were labeled as outliers. Although I produced similar summary plots as for K-Means to analyze the resulting groups, I decided not to include those results in this report due to the limited interpretability and the high number of outliers.



(a) Cluster 5

(b) Cluster 1

Figure 2: Distribution of features in the clusters. For the boolean features only the percentage is reported in the form of a bar plot, while the distribution of the normalized numerical features is represented with boxplots

2.3 Hierarchical Clustering

Using the Euclidean distance metric, I evaluated several linkage methods: single, complete, average, and Ward's method. Among these, average and complete linkage performed best, as assessed by both the Silhouette Score and the Calinski-Harabasz Index.

2.4 Comparison

A comparison between the clustering algorithms was carried out using the Silhouette score and the Calinski-Harabasz (CH) score. The results are reported in Table 1. The highest Silhouette score is achieved by DBSCAN; however, as discussed earlier, DBSCAN tends to classify a large portion of the data points as outliers, which limits its effectiveness in this case. On the other hand, the CH score indicates that K-Means performs best, followed closely by hierarchical clustering using the `complete` and `average` linkage methods.

Metric	`K-Means`	`hier_complete`	`hier_average`	`hier_ward`	`DBSCAN`	`hier_single`
Silhouette	0.134388	0.140719	0.154223	0.127726	**0.180492**	-0.071615
Calinski-Harabasz	**838.415150**	802.136304	781.315251	733.134051	386.018306	1.240939

Table 1: Comparison of clustering algorithms using Silhouette and Calinski-Harabasz scores.

3 Classification

The goal of this task is to train and evaluate different classifiers to predict the target variable *Churn*. The dataset was split into training and test sets, with the test set comprising 30% of the original data. The methods explored include: Decision Tree, Random Forest, k-Nearest Neighbours (kNN), Linear Regression, and Logistic Regression.

3.1 Decision Tree (DT)

For the Decision Tree, I used the rescaled one-hot-encoded dataset, since the algorithm does not require numerical features per se. A naive implementation using the default parameters (Gini impurity, no limit on tree depth, etc.) leads to a highly complex tree, which overfits the training data and generalizes poorly to the test set.

To mitigate overfitting, I applied *Cost Complexity Pruning* (CCP), governed by the parameter α, which penalizes overly complex trees. This approach reduces the need to manually tune other hyperparameters such as *max depth* or *min samples leaf.*

The optimal value of α was determined by computing the accuracy on both training and test sets for different values of α. To increase statistical robustness, the evaluation was repeated using 10-fold cross-validation. The final pruned tree was trained using the α that maximized the test accuracy. The process was repeated for both impurity measures: Gini and Entropy.

3.2 Random Forest

Using the same dataset as for the Decision Tree, I performed a grid search to tune the hyperparameters of the Random Forest classifier. The results are discussed in the following comparison subsection.

3.3 k-Nearest Neighbours (kNN)

To determine the optimal value of k, the algorithm was evaluated over a range of k values, selecting the one that maximized the accuracy on the validation set. Results are included in the comparative analysis below.

3.4 Linear and Logistic Regression

Although Linear Regression is not ideally suited for binary classification, I experimented with it by encoding the target variable numerically (1 for churn, 0 otherwise). The model was implemented using PyTorch on a rescaled dataset (values between 0 and 1). The loss function used was the Mean Squared Error (MSE), equivalent to minimizing the χ^2. The model converged after a few epochs. Results are reported in the next subsection.

Logistic Regression was also tested using the Binary Cross Entropy loss. No hyperparameter tuning was performed. Performance metrics are discussed in the next section.

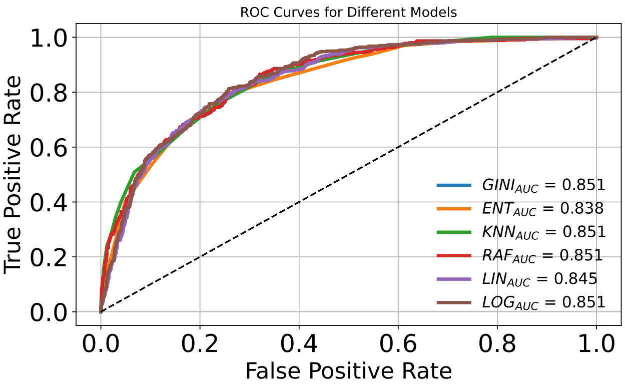


Figure 3: ROC curves of all tested classifiers.

3.5 Comparison Between Classifiers

To compare the classifiers, I used the Receiver Operating Characteristic (ROC) curve and the

Classifier	**Accuracy**	**Precision**	**Recall (Sensitivity)**	**Specificity**	**F1 Score**
Decision Tree	0.772	**0.623**	0.595	**0.872**	0.609
Random Forest	0.812	0.514	0.676	0.850	0.584
kNN	0.817	0.402	**0.765**	0.828	0.526
Logistic Regression	**0.860**	0.559	0.685	0.851	**0.615**

Table 2: Classification metrics computed from confusion matrices.

Area Under the Curve (AUC). Figure 3 shows the ROC curves of all tested classifiers. The results indicate that all classifiers perform similarly, except for Linear Regression and the Decision Tree trained with Entropy, which performed worse.

The highest-performing classifiers reached an AUC of approximately 0.85, which is satisfactory but not optimal. From the confusion matrices, we can compute various performance metrics, which are summarized in Table 2.

From this comparison, Logistic Regression emerges as the best overall performer in terms of accuracy and F1 score, with a balanced trade-off between precision, sensitivity, and specificity. However, the Decision Tree achieves the highest precision and specificity, while kNN offers the highest recall, making it suitable when minimizing false negatives is a priority.

4 Pattern Mining

To uncover hidden relationships between customer features and churn behavior, I employed association rule mining using the Apriori algorithm. The input dataset, containing both numerical and categorical features, was first transformed into a boolean dataframe suitable for pattern mining. Numerical features were normalized between 0 and 1 and then discretized using histogram-based binning, resulting in four categories per feature.

I identified frequent itemsets using the Apriori algorithm. As a preliminary step, I explored how the number of frequent itemsets varied with different support thresholds and minimum itemset lengths. Based on this analysis, I chose to set a minimum support of 3% without imposing a restriction on the number of items per itemset. From the resulting itemsets, I generated association rules using lift as the primary metric, and filtered the results to retain only those with lift greater than 2.1, ensuring strong correlations.

I focused in particular on rules where the consequent is `Churn`, isolating them for interpretation. Among these, the most significant rule—i.e., the one with the highest lift *and* confidence—is:

- **Rule:** {`InternetService_Fiber optic`, `PaperlessBilling`, `tenure_(-0.001, 0.25]`, `TotalCharges_(-0.001, 0.25]`, `PaymentMethod_Electronic check`, `MultipleLines`, `Contract_Month-to-month`} → {`Churn`}

  Support: 3.20%, Confidence: 76.7%, Lift: 2.90

while the rule with the highest support is:

- **Rule:** {`InternetService_Fiber optic`, `PaperlessBilling`, `Contract_Month-to-month`} → {`Churn`}

  Support: 13.6%, Confidence: 56.8%, Lift: 2.14

These rules suggest that short-term contracts, paperless billing, and the use of fiber optic internet service are important factors associated with customer churn. Such insights can inform retention strategies by identifying high-risk customer profiles.

5 Appendix

Lists of plots and tables that could not make it into the report but provide useful information about all the things discussed in this report. Addional studies performed were not reported in this report.

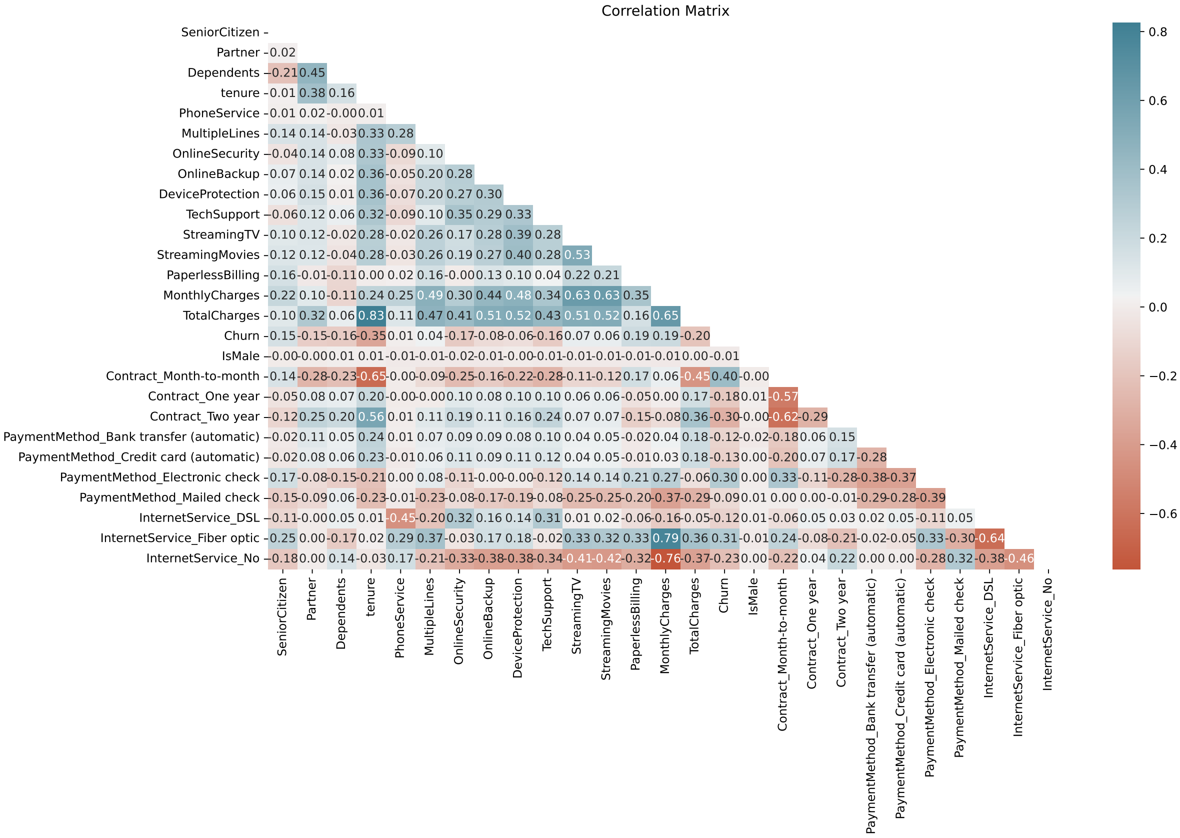


Figure 4: Correlation Heatmap

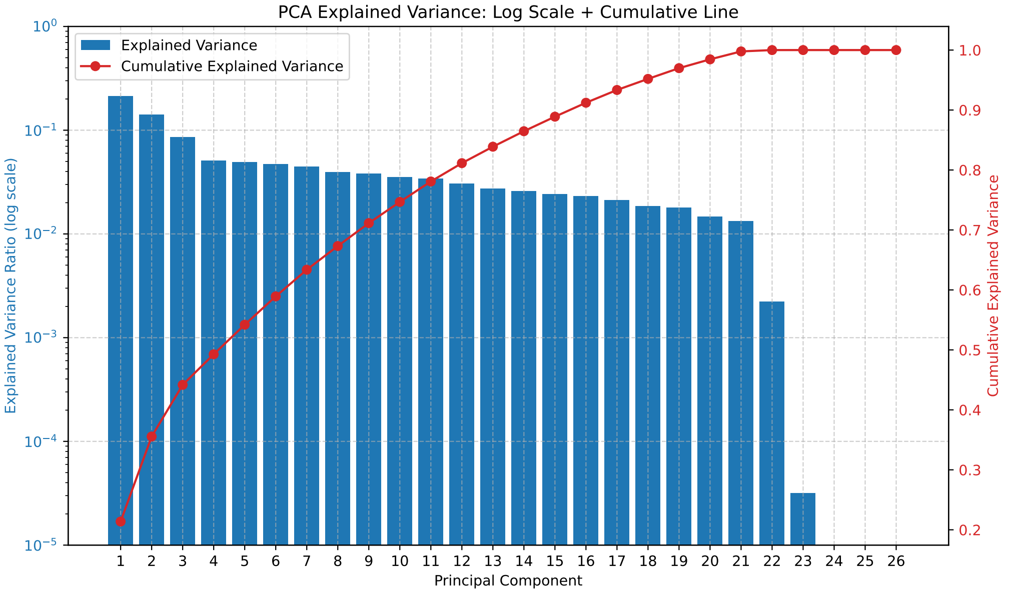
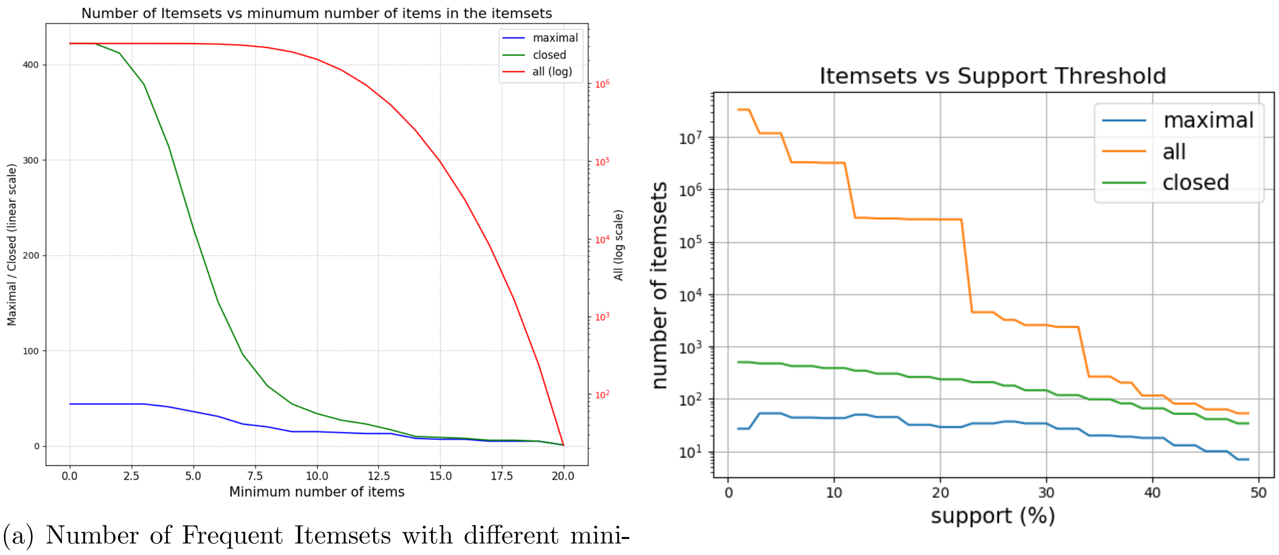


Figure 5: Normalized eigenvalues (explained variance by each component) and total variance explained

	Predicted 1	Predicted 0	Predicted 1	Predicted 0	Predicted 1	Predicted 0	Predicted 1	Predicted 0
True 1	273	186	225	108	176	54	311	143
True 0	165	1129	213	1207	262	1261	246	1403
	Decision Tree		Random Forest		kNN		Logistic Regression	

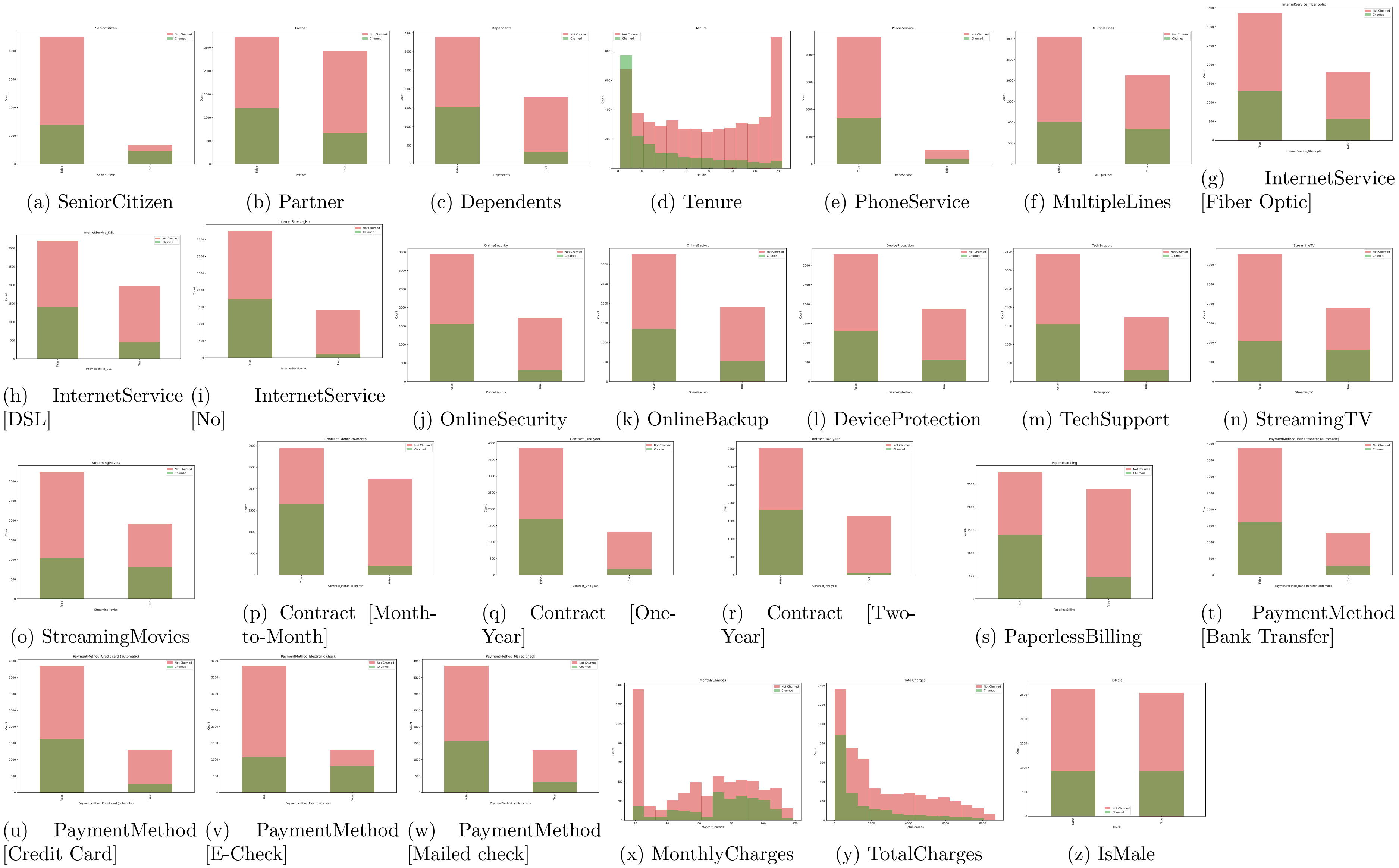
Table 3: Confusion matrices for classifying the target variable with different algorithms. Green indicates True Positives, yellow indicates True Negatives.



(a) Number of Frequent Itemsets with different minimum number of items required

(b) Support

Figure 6: Number of Frequent Itemsets with different minimum requirements (support or number of items)



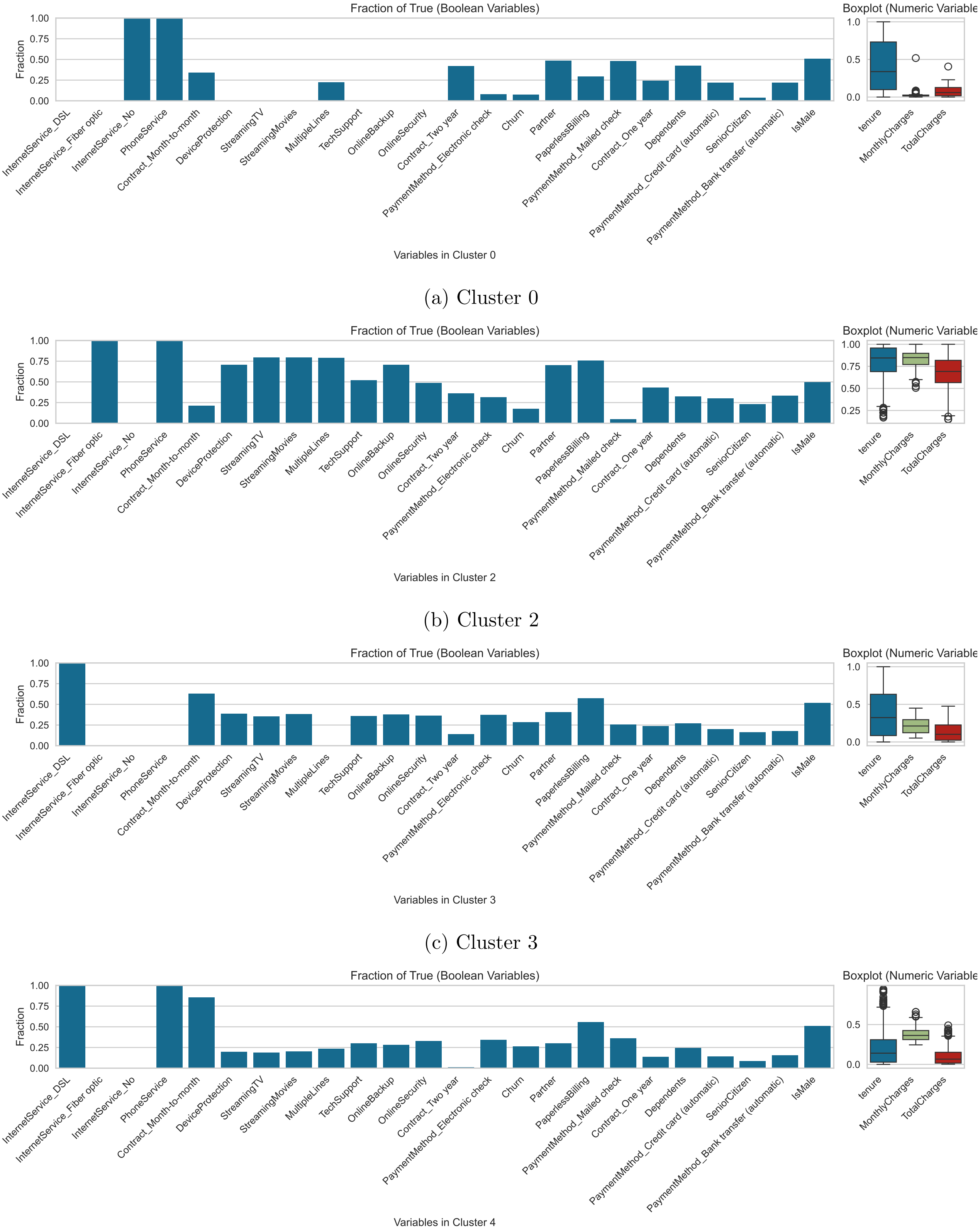
(a) SeniorCitizen (b) Partner (c) Dependents (d) Tenure (e) PhoneService (f) MultipleLines (g) InternetService [Fiber Optic]

(h) InternetService [DSL] (i) InternetService [No] (j) OnlineSecurity (k) OnlineBackup (l) DeviceProtection (m) TechSupport (n) StreamingTV

(o) StreamingMovies (p) Contract [Month-to-Month] (q) Contract [One-Year] (r) Contract [Two-Year] (s) PaperlessBilling (t) PaymentMethod [Bank Transfer]

(u) PaymentMethod [Credit Card] (v) PaymentMethod [E-Check] (w) PaymentMethod [Mailed check] (x) MonthlyCharges (y) TotalCharges (z) IsMale

Figure 7: Distributions of the dataset features after preprocessing.



(a) Cluster 0

(b) Cluster 2

(c) Cluster 3

(d) Cluster 4

Figure 8: Distribution of features in the clusters (not shown in main). For the boolean features only the percentage is reported in the form of a bar plot, while the distribution of the normalized numerical features is represented with boxplots

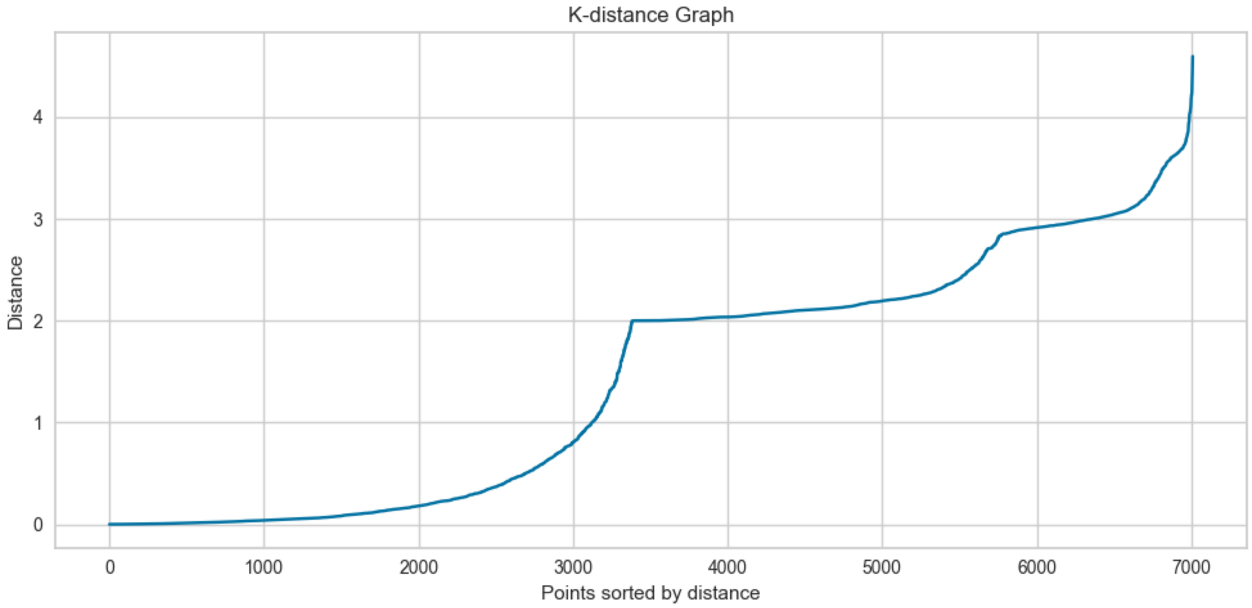
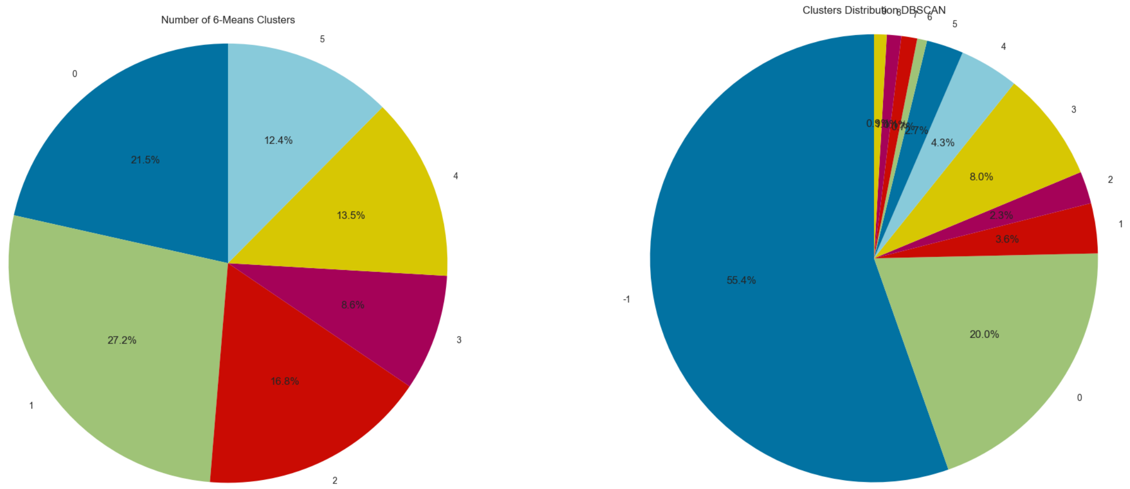


Figure 9: kNN in PCA space with k=5. Analysis performed for choice of parameter ϵ in DBSCAN.



(a) 6-Means

(b) DBSCAN

Figure 10: Number of points in each cluster for both k-Means (with k=6) and DBSCAN. The majority points found by DBSCAN are classified as outliers (`cluster_id`=-1)