Analysis of Customer Churn Patterns to Improve Customer Retention

Gursimran Singh
IIIT-Delhi
gursimran14041
@iiitd.edu.in

Harish Fulara
IIIT-Delhi
harish14143
@iiitd.edu.in

Kushagra Mahajan
IIIT-Delhi
kushagra14055
@iiitd.edu.in

Abstract

Customer Churn or Customer Attrition occurs when customers or subscribers stop doing business with a company or service. For a subscription business, accurately predicting churn is critical to long-term success. Even slight variations in churn can drastically affect profits. Our aim was to apply multiple data-mining algorithms to derive certain insights from the data that might help these businesses to retain their already existent customer base.

We successfully applied various well known data-mining techniques and have been able to extrapolate some interesting insights about the customer churn patterns of customers.

1 Motivation

Customer churn occurs when customers or subscribers stop doing business with a company or service. Customer churn is a critical metric for such companies because it is much less expensive to retain existing customers than it is to acquire new customers. Acquiring new customers means working leads all the way through the sales funnel, utilizing your marketing and sales resources throughout the process. Customer retention, on the other hand, is generally more cost-effective as you have already earned the trust and loyalty of existing customers.

Customer churn impedes growth of a company, so companies should have a defined method for calculating customer churn in a given period of time. By being aware of and monitoring churn rate, organizations are equipped to determine their customer retention success rates and identify strategies for improvement.

There is a direct relationship between customer lifetime value and the ability to grow a business. As such, the higher the customer churn rate, the lower the chances of growing the business. Even if a company has some of the best marketing campaigns in its industry, they bottom line suffers if they are losing customers at a high rate, as the cost of acquiring new customers is so high. Much has been written on the subject of the cost of retaining customers versus acquiring customers, especially because study after study shows that customer acquisition costs far exceed customer retention costs. Generally, companies spend seven times more on customer acquisition than customer retention, and the average global value of a lost customer is $243. Obviously, customer churn is very costly for businesses.

2 Data Acquisition

The data we used is available as part of IBM Watson Analytics familiarity guide (here). . The data contains information about customers of a telecom service provider. It contains information about customers who left within the last month, services that each customer signed up for, customers' account information, and demographic info of the customers.

The data contains the following:

- ***customerID*** - Unique identifier for each customer. Not used as a feature
- ***Gender*** - Male or Female
- ***SeniorCitizen*** - Whether the customer is a senior citizen or not
- ***Partner*** - Whether the customer has a partner or not
- ***Dependents*** - Whether the customer has dependents or not
- ***Tenure*** - Number of months the customer has continued the service

- ***PhoneService*** - Whether the customer has a phone service or not
- ***MultipleLines*** - Whether the customer has multiple telephone line connections or not
- ***InternetService*** - Customers internet service provider
- ***OnlineSecurity*** - Whether the customer has online security or not
- ***OnlineBackup*** - Whether the customer has online backup or not
- ***DeviceProtection*** - Whether the customer has device protection or not
- ***TechSupport*** - Whether the customer has tech support or not
- ***StreamingTV*** - Whether the customer has streaming TV or not
- ***StreamingMovies*** - Whether the customer has streaming movies or not
- ***Contract*** - The contract term of the customer
- ***PaperlessBilling*** - Whether the customer has paperless billing or not
- ***PaymentMethod*** - The customers payment method
- ***MonthlyCharges*** - The amount charged to the customer monthly
- ***TotalCharges*** - The total amount charged to the customer
- ***Churn*** - Whether the customer churned or not. The class Label

3 Preprocessing

The raw data retrieved from the above mentioned source is not suitable to be used with the data-mining algorithms and as such must be transformed into a compatible format before analysis.

3.1 Handling Missing Values

Most real world datasets are prone to having some missing data. This might occur due to errors during the data acquisition process.

Since the proportion of missing data in the IBM dataset is very low, we simply chose to drop any instances that may be missing one or more feature values.

3.2 Categorical Data

Categorical variables represent types of data which may be divided into groups or may take only fixed number of values. Such variables must be converted to a Numeric format that can be used with a mathematical algorithm. We tried two different ways of encoding such data.

3.2.1 One Hot Encoding

One Hot Encoding is defined as follows. Each value is represented by N bits where N is the total number of categories possible. All the bits are cold (0) except the one representing the value's category which is turned hot (1), hence the name.

3.2.2 Label Encoding

Label Encoding is defined as follows. Each category is given an integer label ranging from 0 to $N-1$. Each value is represented by the corresponding category's integer label.

This encoding has an inherent flaw that it assumes an implicit order among different categories. This can be an issue with techniques which aggregate the feature values from all instances thus generating incorrect results.

The resulting feature set is shown in Table 1.

3.3 Numerical Data

Numerical values represent both Discrete and Continuous variables. Such variables can be converted to Encoded Categorical data by discretization into defined buckets of some size.

We discretized the feature ***Tenure*** into 5 categories - the buckets defined as $\{[0, 12], (12, 24], (24, 48], (48, 60], (60, \infty)\}$

4 Feature Selection

Feature selection, the process of finding and selecting the most useful features in a dataset, is a crucial step of the data mining pipeline. Unnecessary features decrease training speed, decrease model interpretability, and, most importantly, decrease generalization performance on the test set.

4.1 Correlation Based Feature Selection

We use the Pearson correlation metric to create a Correlation Matrix (in Figure 1). If two different features are highly correlated, it means that one of the features can be dropped since it won't provide the algorithm with any new information about the data.

Feature Name	One-Hot	Label
Gender	2 Bits	1 Int
SeniorCitizen	2 Bits	1 Int
Partner	2 Bits	1 Int
Dependents	2 Bits	1 Int
Tenure*	5 Bits	1 Int
PhoneService	2 Bits	1 Int
MultipleLines	3 Bits	1 Int
InternetService	3 Bits	1 Int
OnlineSecurity	3 Bits	1 Int
OnlineBackup	3 Bits	1 Int
DeviceProtection	3 Bits	1 Int
TechSupport	3 Bits	1 Int
StreamingTV	3 Bits	1 Int
StreamingMovies	3 Bits	1 Int
Contract	3 Bits	1 Int
PaperlessBilling	2 Bits	1 Int
PaymentMethod	4 Bits	1 Int
MonthlyCharges*	1 Float	1 Float
TotalCharges	1 Float	1 Float
Total	50	19

Table 1: Feature Representation with different encodings
∗ Removed at a further stage

We chose to **drop the following features** due to the high correlation with existing features as shown by the correlation matrix Figure 1.

- ***Tenure*** - Highly correlated with *Total Charges* and *Contract*
- ***Monthly Charges*** - Highly correlated with *Total Charges*, *StreamingTV* and *StreamingMovies*

5 Data Visualization

We first need to understand how the data looks from a human perspective. To achieve this we need to convert the high dimensional data down to 2 dimensions, so we can plot it on a 2D graph. We employ two different techniques for dimensionality reduction.

5.1 Principal Component Analysis

The main idea of principal component analysis (PCA) is to reduce the dimensionality of a data set consisting of many variables correlated with each other, either heavily or lightly, while retaining the variation present in the dataset, up to the maximum extent. The same is done by transforming the variables to a new set of variables, which are known as the principal components and are orthogonal, ordered such that the retention of variation present in the original variables decreases as we move down in the order.

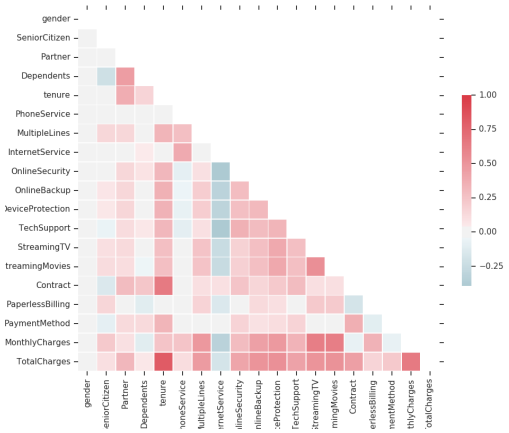


Figure 1: Pearson Correlation Matrix

2-Dimensional PCA plots of the data are provided in Figure 2

5.2 t-SNE

t-Distributed Stochastic Neighbor Embedding (t-SNE) is a non-linear technique for dimensionality reduction that is particularly well suited for the visualization of high-dimensional datasets.

2-Dimensional t-SNE plots of the data are provided in Figure 3

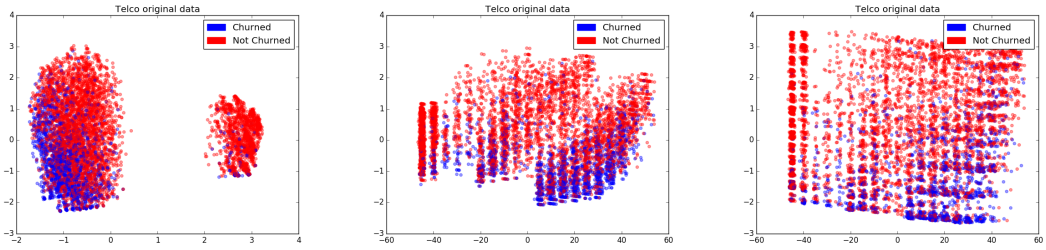
6 Outlier Analysis

We pass our data once through some outlier analysis tests to improve the quality of the said data before clustering and classification tasks. Later, we present our classification results with respect to both data - with outliers and without.

6.1 Grubb's Test or Z-Score Analysis

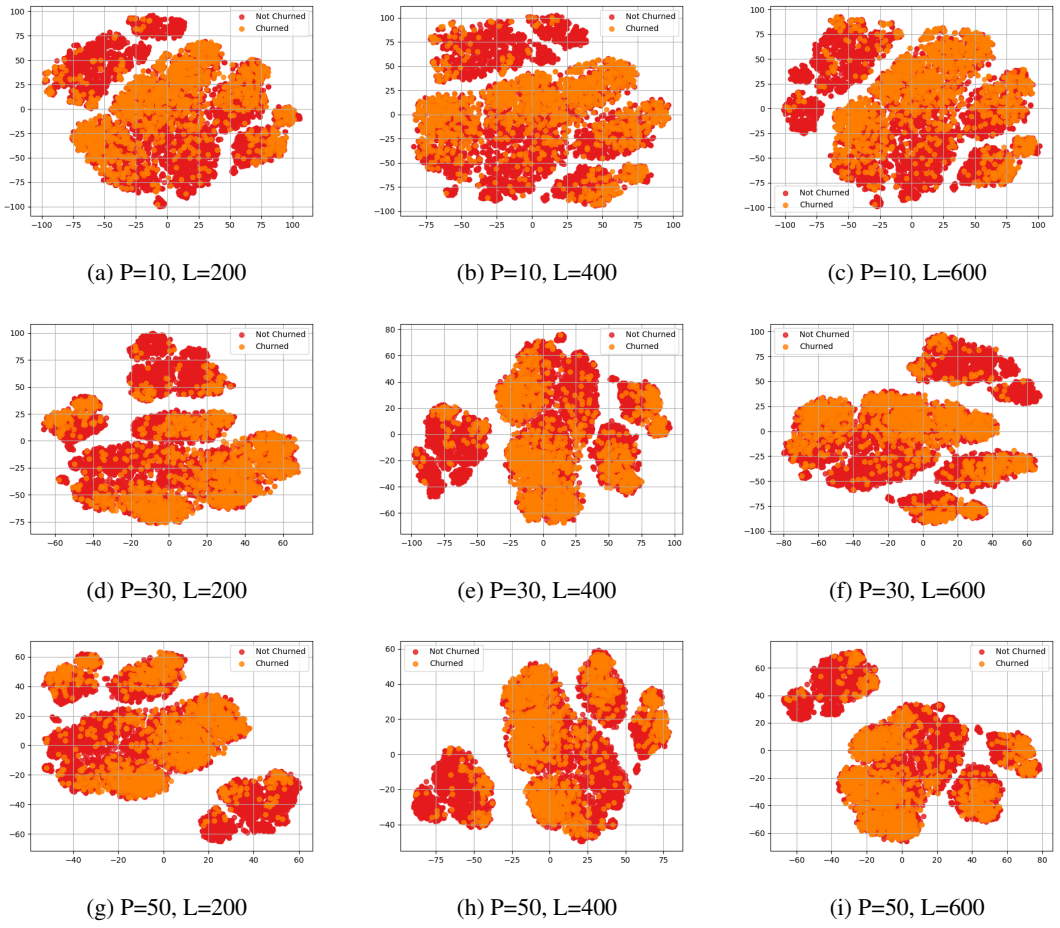
The z-score or standard score of an observation is a metric that indicates how many standard deviations a data point is from the samples mean, assuming a gaussian distribution. This makes z-score a parametric method. Very frequently data points are not to described by a gaussian distribution, this problem can be solved by applying transformations to data ie: scaling it.

This method eliminated 1500 instances as outliers from the IBM dataset.



(a) Label Encoding (b) One-Hot Enc. + Feature Selection (c) Label Enc. + Feature Selection

Figure 2: Principal Component Analysis plots of the data



(a) P=10, L=200 (b) P=10, L=400 (c) P=10, L=600

(d) P=30, L=200 (e) P=30, L=400 (f) P=30, L=600

(g) P=50, L=200 (h) P=50, L=400 (i) P=50, L=600

Figure 3: t-SNE plots of the data, P = Perplexity, L = Learning Rate

6.2 DBSCAN

Dbscan is a density based clustering algorithm, it is focused on finding neighbors by density (MinPts) on an n-dimensional sphere with radius ϵ.

This method eliminated 97 instances as outliers from the IBM dataset.

7 Clustering Analysis

We tried to cluster the data to check if this problem can be solved in an unsupervised fashion. We also hope to see some interesting patterns among the features if the data belonging to the two classes (churn or no-churn) were assigned to separable clusters by the algorithms.

7.1 K-Means

Used the standard K-Means algorithm. The results are shown in Table 2

7.2 K-Medians

Used the standard K-Medians algorithm. The results are shown in Table 3

8 Classification

Our final approach is to treat the problem as a classification task. We tried various classification techniques as discussed below.

The classification scores have been mentioned for 4 different iterations of the same data preprocessed in different ways:

- A: Label Encoding without Correlation based Feature Selection
- B: Label Encoding with Correlation based Feature Selection
- C: One Hot Encoding without Correlation based Feature Selection
- D: One Hot Encoding with Correlation based Feature Selection

All model parameters have been fine tuned to give the best possible F1 score using **Grid Search Cross Validation** for training. We kept the **train-test ratio** $70\% : 30\%$ and used **5 folds of cross validation** for calculating training scores. Also before training, we have kept a **held-out** set which is never seen during training and is used to calculate the testing scores.

8.1 Logistic Regression

Training the standard implementation of Logistic Regression Classifier on the data. The corresponding cross-validation (training) scores and testing scores on a completely held out set have been mentioned in Table 4.

The best tuned parameters are $C = 0.001$, Solver = liblinear and Penalty = L2.

8.2 Naive Bayes

Training the standard implementation of Naive Bayes Classifier on the data. The corresponding cross-validation (training) scores and testing scores on a completely held out set have been mentioned in Table 5.

8.3 Support Vector Machine

Training the standard implementation of SVM Classifier on the data. The corresponding cross-validation (training) scores and testing scores on a completely held out set have been mentioned in Table 6.

The best tuned parameters are $C = 3$ and Kernel = RBF.

8.4 SVM after Outlier Removal using Grubb's Test

Training the standard implementation of SVM Classifier on the data which has been passed through the Grubb's Outlier elimination pipeline. The corresponding cross-validation (training) scores and testing scores on a completely held out set have been mentioned in Table 7.

The best tuned parameters are $C = 3$ and Kernel = RBF.

8.5 SVM after Outlier Removal using DBSCAN

Training the standard implementation of SVM Classifier on the data which has been passed through the DBSCAN Outlier elimination pipeline. The corresponding cross-validation (training) scores and testing scores on a completely held out set have been mentioned in Table 8.

The best tuned parameters are $C = 10$ and Kernel = RBF for the SVM Classifier. The best parameters for DBSCAN model are EPS $= 2.45$, Min-Samples $= 20$.

8.6 SVM after Dimentionality Reduction using PCA

Training the standard implementation of SVM Classifier on the data whose dimensions have been reduced using Principal Component Analysis. The corresponding cross-validation (training) scores and testing scores on a completely held out set have been mentioned in Table 9.

The best tuned parameters are $C = 3$ and Kernel = RBF for the SVM Classifier. The best parameters for PCA are Num-Components $= 20$.

8.7 Decision Tree

Training the standard implementation of Decision Tree Classifier on the data. The corresponding cross-validation (training) scores and testing scores on a completely held out set have been mentioned in Table 10.

Enc.	Score	2	3	4	5	6	7	8	9	10
One Hot	ARI	−0.007	0.027	0.016	0.031	0.043	0.028	0.02	0.025	0.017
	AMI	0.01	0.071	0.065	0.054	0.053	0.051	0.046	0.043	0.048
	Homogeneity	0.012	0.13	0.15	0.144	0.162	0.167	0.163	0.165	0.187
	Completeness	0.011	0.071	0.066	0.054	0.053	0.051	0.046	0.044	0.049
	Silhoutte	0.168	0.143	0.126	0.142	0.135	0.114	0.11	0.103	0.105
Label	ARI	−0.002	0.01	0.012	0.022	0.025	0.026	0.031	0.014	0.031
	AMI	0.025	0.016	0.019	0.016	0.016	0.016	0.018	0.018	0.016
	Homogeneity	0.03	0.03	0.045	0.045	0.049	0.054	0.064	0.067	0.063
	Completeness	0.025	0.016	0.019	0.016	0.016	0.017	0.018	0.018	0.016
	Silhoutte	0.065	0.058	0.06	0.057	0.053	0.051	0.049	0.044	0.044

Table 2: Results for K-Means Clustering, for One-Hot and Label Encoding, calculated for $K = 2$ to 10.

Enc.	Score	2	3	4	5	6	7	8	9	10
One Hot	ARI	0.014	0.00	0.035	−0.01	0.042	0.036	0.037	0.00	0.006
	AMI	0.042	0.036	0.048	0.045	0.045	0.048	0.043	0.047	0.037
	Homogeneity	0.05	0.066	0.115	0.115	0.137	0.156	0.15	0.154	0.143
	Completeness	0.042	0.036	0.049	0.045	0.045	0.048	0.043	0.047	0.038
	Silhoutte	0.055	0.115	0.102	0.058	0.09	0.092	0.081	0.039	0.033
Label	ARI	−0.013	0.003	0.017	0.022	0.001	0.023	0.032	0.009	0.013
	AMI	0.022	0.021	0.023	0.018	0.023	0.026	0.021	0.024	0.03
	Homogeneity	0.026	0.039	0.056	0.05	0.07	0.082	0.076	0.09	0.118
	Completeness	0.022	0.021	0.023	0.018	0.023	0.026	0.022	0.024	0.03
	Silhoutte	0.064	0.059	0.058	0.056	0.04	0.048	0.046	0.035	0.031

Table 3: Results for K-Medians Clustering, for One-Hot and Label Encoding, calculated for $K = 2$ to 10.

Table	Metric	F-1	F-2	F-3	F-4	F-5	Train Avg.	Held Out
A	**Accuracy**	0.789	0.8	0.802	0.803	0.803	0.799	0.788
	Precision	0.602	0.653	0.656	0.657	0.657	0.645	0.602
	Recall	0.614	0.532	0.537	0.543	0.543	0.554	0.601
	F1	0.608	0.586	0.59	0.594	0.594	0.594	0.601
B	**Accuracy**	0.79	0.802	0.802	0.803	0.803	0.8	0.784
	Precision	0.603	0.655	0.656	0.657	0.657	0.646	0.595
	Recall	0.609	0.537	0.537	0.543	0.543	0.554	0.59
	F1	0.606	0.59	0.59	0.594	0.594	0.595	0.593
C	**Accuracy**	0.782	0.8	0.802	0.804	0.803	0.798	0.782
	Precision	0.573	0.642	0.66	0.663	0.661	0.64	0.575
	Recall	0.71	0.562	0.53	0.532	0.534	0.574	0.695
	F1	0.634	0.599	0.587	0.59	0.591	0.6	0.63
D	**Accuracy**	0.782	0.802	0.802	0.801	0.801	0.798	0.78
	Precision	0.578	0.649	0.662	0.662	0.662	0.643	0.576
	Recall	0.671	0.554	0.521	0.515	0.516	0.556	0.656
	F1	0.621	0.597	0.583	0.579	0.58	0.592	0.613

Table 4: Results for Logistic Regression Classifier: (A) Label Enc., (B) Label Enc. with feature selection, (C) One-Hot Enc., (D) One-Hot Enc. with feature selection

Table	Metric	F-1	F-2	F-3	F-4	F-5	Train Avg.	Held Out
A	**Accuracy**	0.763	0.744	0.748	0.769	0.774	0.76	0.762
	Precision	0.541	0.514	0.52	0.545	0.553	0.535	0.537
	Recall	0.733	0.698	0.695	0.782	0.774	0.736	0.742
	F1	0.622	0.592	0.595	0.643	0.645	0.619	0.623
B	**Accuracy**	0.769	0.749	0.746	0.779	0.78	0.765	0.757
	Precision	0.55	0.521	0.518	0.559	0.564	0.543	0.532
	Recall	0.714	0.698	0.653	0.793	0.759	0.723	0.724
	F1	0.621	0.597	0.578	0.656	0.647	0.62	0.613
C	**Accuracy**	0.713	0.638	0.69	0.691	0.72	0.69	0.678
	Precision	0.477	0.414	0.455	0.457	0.485	0.458	0.444
	Recall	0.824	0.874	0.832	0.874	0.862	0.853	0.836
	F1	0.604	0.562	0.588	0.6	0.621	0.595	0.58
D	**Accuracy**	0.763	0.743	0.735	0.762	0.763	0.753	0.749
	Precision	0.54	0.512	0.501	0.535	0.536	0.525	0.519
	Recall	0.756	0.763	0.718	0.801	0.793	0.766	0.766
	F1	0.63	0.613	0.59	0.641	0.64	0.623	0.619

Table 5: Results for Naive Bayes Classifier: (A) Label Enc., (B) Label Enc. with feature selection, (C) One-Hot Enc., (D) One-Hot Enc. with feature selection

Table	Metric	F-1	F-2	F-3	F-4	F-5	Train Avg.	Held Out
A	**Accuracy**	0.799	0.794	0.79	0.783	0.78	0.789	0.785
	Precision	0.665	0.64	0.625	0.603	0.595	0.625	0.614
	Recall	0.492	0.515	0.53	0.539	0.547	0.524	0.613
	F1	0.566	0.57	0.573	0.569	0.569	0.569	0.612
B	**Accuracy**	0.799	0.791	0.785	0.781	0.776	0.786	0.783
	Precision	0.664	0.631	0.613	0.598	0.584	0.618	0.615
	Recall	0.491	0.52	0.524	0.537	0.547	0.524	0.69
	F1	0.564	0.57	0.565	0.565	0.565	0.566	0.598
C	**Accuracy**	0.8	0.787	0.777	0.768	0.761	0.779	0.785
	Precision	0.663	0.617	0.59	0.568	0.555	0.599	0.667
	Recall	0.502	0.526	0.532	0.528	0.519	0.521	0.49
	F1	0.571	0.568	0.559	0.547	0.536	0.556	0.565
D	**Accuracy**	0.797	0.791	0.781	0.774	0.771	0.783	0.782
	Precision	0.662	0.632	0.599	0.581	0.574	0.609	0.611
	Recall	0.485	0.519	0.53	0.543	0.546	0.524	0.492
	F1	0.559	0.569	0.562	0.561	0.559	0.562	0.545

Table 6: Results for Support Vector Machine Classifier: (A) Label Enc., (B) Label Enc. with feature selection, (C) One-Hot Enc., (D) One-Hot Enc. with feature selection

The best tuned parameters are Max-Depth $=$ 10, Min-Splits $= 2$.

9 Results and Conclusions

This section summarizes all of our progress and mentions some of the key insights that we have derived and learned during the process.

9.1 Pre-Processing

We tried two different data encoding techniques (one-hot and label) and ran all the algorithms on both the formats. We do not see any significant difference in the accuracy scores that we have achieved for either technique to surely consider one better than the other.

Table	Metric	F-1	F-2	F-3	F-4	F-5	Train Avg.	Held Out
B	**Accuracy**	0.8	0.8	0.796	0.788	0.784	0.794	0.786
	Precision	0.624	0.608	0.593	0.57	0.558	0.591	0.608
	Recall	0.433	0.488	0.499	0.502	0.514	0.487	0.5
	F1	0.511	0.541	0.542	0.533	0.535	0.532	0.549
D	**Accuracy**	0.804	0.796	0.789	0.785	0.78	0.791	0.799
	Precision	0.636	0.596	0.573	0.559	0.545	0.582	0.656
	Recall	0.44	0.481	0.496	0.5	0.509	0.485	0.488
	F1	0.52	0.532	0.531	0.527	0.526	0.527	0.56

Table 7: Results for Grubb's Test + Support Vector Machine Classifier: (B) Label Enc. with feature selection, (D) One-Hot Enc. with feature selection

Table	Metric	F-1	F-2	F-3	F-4	F-5	Train Avg.	Held Out
B	**Accuracy**	0.791	0.789	0.791	0.789	0.787	0.789	0.8
	Precision	0.666	0.656	0.657	0.651	0.646	0.655	0.657
	Recall	0.461	0.469	0.476	0.473	0.469	0.47	0.474
	F1	0.544	0.547	0.552	0.548	0.544	0.547	0.551
D	**Accuracy**	0.79	0.791	0.791	0.793	0.791	0.791	0.794
	Precision	0.68	0.676	0.673	0.674	0.666	0.674	0.667
	Recall	0.468	0.483	0.485	0.497	0.501	0.487	0.471
	F1	0.554	0.563	0.563	0.572	0.572	0.565	0.552

Table 8: Results for DBSCAN + Support Vector Machine Classifier: (B) Label Enc. with feature selection, (D) One-Hot Enc. with feature selection

Table	Metric	F-1	F-2	F-3	F-4	F-5	Train Avg.	Held Out
B	**Accuracy**	0.8	0.798	0.793	0.792	0.787	0.794	0.792
	Precision	0.671	0.647	0.629	0.626	0.61	0.637	0.634
	Recall	0.488	0.532	0.539	0.546	0.547	0.53	0.513
	F1	0.564	0.584	0.58	0.583	0.577	0.578	0.567
D	**Accuracy**	0.788	0.785	0.778	0.773	0.766	0.778	0.772
	Precision	0.636	0.615	0.594	0.581	0.565	0.599	0.584
	Recall	0.473	0.508	0.517	0.524	0.519	0.508	0.488
	F1	0.542	0.556	0.553	0.551	0.541	0.549	0.532

Table 9: Results for PCA + SVM Classifier: (B) Label Enc. with feature selection, (D) One-Hot Enc. with feature selection

9.2 Feature Importance

The Pearson correlation matrix showed that only 2 of the features were highly correlated. Removing these features from the data did not improve the accuracy significantly.

This also means that all the features in the dataset can potentially be important in the classification task (since each one has the potential to provide unrelated information)

9.3 Data Visualization

The t-SNE and PCA plots show extreme overlap among the instances belonging to the two classes. This indicates that no single (or two for that matter) principal component(s) of the data that is/are able to determine the customer churn behaviour. A more comprehensive dataset might be better for the task at hand.

Table	Metric	F-1	F-2	F-3	F-4	F-5	Train Avg.	Held Out
A	**Accuracy**	0.758	0.758	0.73	0.736	0.734	0.743	0.739
	Precision	0.546	0.546	0.493	0.504	0.499	0.517	0.508
	Recall	0.53	0.531	0.508	0.497	0.519	0.517	0.56
	F1	0.537	0.538	0.5	0.5	0.509	0.517	0.533
B	**Accuracy**	0.759	0.761	0.727	0.733	0.727	0.741	0.755
	Precision	0.548	0.553	0.486	0.499	0.486	0.515	0.542
	Recall	0.534	0.525	0.505	0.476	0.489	0.506	0.512
	F1	0.541	0.538	0.495	0.487	0.488	0.51	0.526
C	**Accuracy**	0.723	0.728	0.711	0.714	0.705	0.716	0.72
	Precision	0.482	0.489	0.459	0.462	0.45	0.468	0.476
	Recall	0.529	0.518	0.495	0.474	0.484	0.5	0.512
	F1	0.504	0.503	0.477	0.467	0.466	0.483	0.493
D	**Accuracy**	0.759	0.76	0.725	0.73	0.731	0.741	0.753
	Precision	0.547	0.548	0.484	0.492	0.494	0.513	0.535
	Recall	0.554	0.55	0.505	0.476	0.506	0.518	0.545
	F1	0.55	0.549	0.493	0.483	0.499	0.515	0.54

Table 10: Results for Decision Tree Classifier: (A) Label Enc., (B) Label Enc. with feature selection, (C) One-Hot Enc., (D) One-Hot Enc. with feature selection

9.4 Cluster Analysis

It is quite evident from the lower values of ARI score (measures the similarity between ground truth labels and assigned labels), AMI score (measures the agreement between ground truth labels and assigned labels), homogeneity (each cluster contains only members of a single class), and completeness (all members of a given class are assigned to the same cluster) scores that the data does not have clustering tendency. Silhouette score is also close to zero which implies that the clusters overlap with each other. The tendency of the dataset to not form good clusters makes it harder to analyze the churn patterns in the dataset.

9.5 Classification

Classification techniques are much more successful in recognizing the customer churn behavior. All of our classification models - Gaussian Naive Bayes, Decision Tree, Logistic Regression, and Support Vector Machines - give an accuracy of more than 75% on the validation as well as the held-out sets. SVM gives the best accuracy of 80% for label encoding with feature selection after removing noise using DBSCAN.