TELCO CUSTOMER CHURN

Final Report

Data Science in Business and Entrepreneurship Premaster

Introduction to Machine Learning Course

Group 3:

Alina Baciu

Alexandra Benekou

Mikolaj Hilgert

Table of Contents

List of figures

List of Tables

1. Introduction

1.1 Business Understanding

1.1.1 Context

Telco is a telecommunications company, which offers miscellaneous services to its customers. They offer phone services, including multiple lines, also, internet services, such as online security and backup, tech support and device protection but also, streaming Tv and/or movies.
They also provide eco-friendly options to their customers, such as paperless billing.

1.1.2 Problem

Telco, like many other businesses, has been experiencing customer churn. This is very worrying for Telco as they rely on a subscription based revenue model. These types of businesses depend on customers to make recurring payments. This is opposed to transaction-based revenue models where customers make one-time purchases.

Consequently, that means if a customer of Telco churns, meaning they leave the business. That means that they have also canceled their subscription with them. This results in an immediate loss of revenue for Telco.

This however is not the only negative that can come from customer churn. A growing churn rate for a business can hurt its overall brand image amongst potential customers. As this may cause new potential customers to not want to do business with you, based on others leaving your company.

Furthermore, it is a known fact in business that it is more expensive to attain new customers than it does to retain existing customers. Which is supported by research by Harvard Business review who found that acquiring a new customer is anywhere from 5 to 25 times more expensive than retaining an existing one. (Amy Gallo, *The Value of Keeping the Right Customers ,* October 2014) As such, for the business to balance the lost revenue from churning customers, they will need to invest more time and money in acquiring new clients.

In summary, for subscription-based businesses like Telco, customer churn can have serious negative consequences for revenue, brand image, and overall growth potential within the already very competitive telecom market.

1.1.3 Business Goal

For Telco, the retention of customers is essential to their success in the very competitive telecom market.
Customer churn can have significant negative impacts on the business. Therefore, the ability to identify the customers at risk of churn is of great value.

Therefore, our formal business question is: *How to identify at-risk Telco customers for potential churn so that they can be targeted?*

This process will allow Telco to proactively address their at-risk customers concerns and attempt to retain their loyalty. This can be through multiple methods, such as providing them with targeted deals that aim to keep them with the business. This in return will help to reduce revenue loss, maintain market share, and lower the aforementioned costs associated with acquiring new customers.

1.1.4 Research Goal

Now, we have defined our business objective. We can look at what we are trying to achieve in an academic sense. There are many fields and topics that stand to gain a lot from machine learning. What we want to explore is whether Machine Learning can be of aid to businesses in the real world, and more specifically in regards to detection of customer churn. Since customers are what keeps the lights on for most businesses. We would like to explore the extent of the potential and viability of ML in this topic.

As such we have defined a research question that we would like to have answered at the end of this document. Namely; *To what extent can Machine learning be used to predict customer churn?*

2. Description of Machine Learning Techniques

For this project we applied Supervised learning techniques, that means that the model we created, trains on known input and output data, so that it will be able to make predictions about future inputs. There are many algorithms that can be used, but we shall focus on the three that we

used, those are Decision Trees, Support Vector Machines and Logistic Regression. We describe each in this section.

2.1 Decision Trees

Decision Trees is one of the simpler algorithms and it can be used for both Regression and Classification problems. As the name implies, this algorithm uses a tree-like model to make decisions.
At the start, the algorithm begins with the root node, which represents the entire data set. Then, the training set is split according to a certain parameter, into subsets (also called branches). This procedure is repeated on each subset until each branch ends with leaf nodes.

But how can one choose where to split the training set and which should be the root node? The root node or first test attribute is selected based on information gain. Information gain helps to measure the reduction of uncertainty of a certain feature and it is based on entropy and information content. The formula of information gain is:

$$Gain(D, A) = H(D) - \sum_{v \in Values(A)} \frac{|D_v|}{D} H(D_v)$$

Figure 1. Information Gain formula

Where D refers to the dataset we want to split on, Values(A) is the set of values for attribute A and Dv is the set of distances in D whose value for A is v.

Entropy measures the purity (or impurity) of a dataset, or the uncertainty in the classification, the formula of it is : $\mathbf{S = k_B ln \Omega}$. Where **S** is the entropy, $\mathbf{k_B}$ is the Boltzmann constant, ***ln*** is the natural logarithm with the base of e (Euler's number) and **Ω** is the number of microscopic configurations. Using entropy, it is possible to know whether a subset is homogeneous or heterogeneous and it is also used to calculate the information gain. Information gain, measures the relative change in entropy with respect to the independent variables. It is also known that the higher the number of instances and of the impurity, the lower the information gain in using the tested attribute. (Catolino Gemma, *Decision Trees,* March 2022)

So, when choosing where to split, first it is needed to calculate the entropy per attribute included in the dataset. Then, select the attribute for which, after splitting, the entropy is minimized (thus,

information gain is maximized) and split the subset using that attribute. After, create a decision tree node containing that attribute. Repeat the process until each subset is pure (homogeneous).

During this process, it is possible for the created tree to "grow" too much and this will make it complex and could lead to overfitting. Overfitting, according to Tom Dietterich, is the phenomenon when we fit the noise of a dataset by memorizing various peculiarities of the training data rather than finding a general predictive rule (Dietterich, 1995). So, there are many ways to make sure that the tree does not reach this stage. It is possible to set a minimum set of training inputs to use on each leaf or a maximum depth of the model (Maximum depth is the length of the longest path from the root node till the leaf node).
Also, another option that can increase the efficiency of the tree is pruning. Pruning takes place when branches of less importance are removed from the tree, thus reducing overfitting and making the tree more accurate. There are many types of pruning that can be applied, such as reduced error pruning, cost complexity pruning and weakest link pruning, they all have different ways in determining which node will be removed from the tree. To see how a decision tree algorithm works, a pseudocode is given below, which shows the steps mentioned above.

Pseudocode:

1) Define function called create_decision_tree
2) Collect and pre-process the data
3) Split the dataset into training and testing sets
4) For each feature in the dataset:
   i. Calculate the information gain for that feature
5) Choose feature with highest information gain
6) Create decision node based on selected feature
7) For each possible value of selected feature:
   i. Create branch from decision node
   ii. Split dataset based on selected feature value
   iii. If subset of dataset is empty, assign majority class label to branch
   iv. Else, recursively apply create_decision_tree function to subset of dataset corresponding to branch
8) Return decision tree

To further explain the steps the decision tree algorithm uses in order to make a prediction, here is a random section of our data set, on which we shall apply a simplified version of the algorithm.

CustomerID	Tenure	Contract	MonthlyCharge	TotalCharges	Churn
7590-VHVEG	1	Month-to-month	29.85	29.85	No
5575-GNVDE	34	One year	56.95	1889.5	No
3668-QPYBK	2	Month-to-month	53.85	108.15	Yes
7795-CFOCW	45	One year	42.3	1840.75	No
9237-HQITU	2	Month-to-month	70.7	151.65	Yes
9305-CDSKC	8	Month-to-month	99.65	820.5	Yes
1452-KIOVK	22	Month-to-month	89.1	1949.4	No
6713-OKOMC	10	Month-to-month	29.75	301.9	No
7892-POOKP	28	Month-to-month	104.8	3046.05	Yes
6388-TABGU	62	One year	56.15	3487.95	No
9763-GRSKD	13	Month-to-month	49.95	587.45	No
7469-LKBCI	16	Two year	18.95	326.8	No
8091-TTVAX	58	One year	100.35	5681.1	No
0280-XJGEX	49	Month-to-month	103.7	5036.3	Yes

Figure 2. Example Decision Trees

A possible *simple* decision tree for the above data set could be:

```
If Contract = Month-to-month
        If MonthlyCharge > 53.5
                Churn = Yes
        Else
                Churn = No
Else
        Churn = No
```

This would be interpreted as follows, if a customer has a month to month contract, we check their monthly charge, if it is higher than 53.5, then we predict that the customer will churn. If the customer has a month to month contract and their monthly charge is lower or equal to 53.5, then we predict that the customer will not churn.

If the customer doesn't have a month to month contract with the company, then we predict that they will not churn.

To summarize, in order to make a prediction we follow the decision tree based on the attributes, until we reach a leaf node.

2.2 Logistic Regression

The second algorithm we used in this project is logistic regression. This is a statistical method that is used for binary classification problems, such as customer customer churn. This algorithm uses a logistic function to mold the relationship of the input data and the class label.
There are three categories of logistic regression, binomial which can only have two types of dependant variables (e.g churn, no churn), multinomial which can have three or more unordered types (eg bananas, apples, mangos) and ordinal which also has three or more types, but they are ordered (eg. small, medium, large).
The logistic function, also called sigmoid function, is an S shaped curve that can have any value between the number 0 and 1. (Catolino Gemma & Pecorelli Fabiano, *Regression,* March 2023)

When wanting to apply the algorithm on a data set (Kumar, March 2023), these are the steps that are needed to follow:

1) Collect and preprocess the data. This step includes cleaning, transforming, and normalizing the data if needed.
2) Variable selection. Then we need to identify the independent variables that are related to the dependent variable and remove any that do not offer anything to the model.

3) Model fitting: Fit the logistic regression model to the training data, estimating the coefficients and intercepts that maximize the likelihood of the observed outcomes.

4) Model evaluation: Evaluate the performance of the model on the test data, using the confusion matrix (this includes metrics such as accuracy, precision, recall, and F1 score).

5) Model optimization: If we are not satisfied with the performance of the model then adjust its parameters or feature selection and repeat steps 3 and 4.

6) Model deployment: Once satisfied with the predictions of the model, it is ready to be deployed.

2.3 Support Vector Machines (SVM)

Support Vector Machines work by spotting the best possible decision boundary between two classes of data. The basic idea behind SVM is to find the hyperplane that separates the two classes in the best optimal way. In two dimensions, a hyperplane is simply a line that separates the two classes. In higher dimensions, the hyperplane is a hyperplane.

SVM tries to find the hyperplane that has the largest margin between the two classes. The margin is interpreted as the distance between the hyperplane and the closest data points from each class. The data points that are closest to the hyperplane are called support vectors, each class is given a sign (- or +).
The process of finding the optimal hyperplane involves solving a quadratic optimization problem. To make the problem more manageable , SVM uses a kernel function to transform the data into a higher dimensional space where the hyperplane can be more easily found. (Pecorelli Fabiano, *Support Vector Machines,* March 2023)

Hyperparameter tuning is the next step in the algorithm. This process includes selecting the hyperparameters that fit best, before the training of the algorithm begins. The hyperparameters can affect the performance of the model, so it is very important to select the correct parameters, they could include regularization strength, batch size, learning rate and number of layers.
Some of the possible methods that can be used to complete the hyperparameter tuning phase include, random search, grid search, bayesian optimisation and evolutionary algorithms.

Once the model is trained, it is able to make a prediction on new unlabeled data. At that time, the model predicts a class in which the new data point falls into, by checking in which side of the hyperplane it falls into. Lets see how the algorithm works, step by step.

1) Collect and preprocess data: Collect and preprocess the data, this includes cleaning it, removing missing values, and converting categorical variables to numerical ones.

2) Split the data: Split the data into training and testing sets.

3) Choose a kernel function: Choose a kernel function that will transform the data into a multidimensional space where the hyperplane can be more easily found. Some of the several kernel types that you can choose from include linear, polynomial, radial basis function (RBF) and sigmoid.

4) Choose hyperparameters: Choose hyperparameters for the model, such as the regularization parameter, C, and the kernel function parameters.

5) Train the model: Train the SVM model on the training set using the kernel function and hyperparameters. We want to find the hyperplane that has the largest margin between the two classes, but does not include any data points.

6) Model evaluation: Evaluate the performance of the model on the test data, using the confusion matrix (this includes metrics such as accuracy, precision, recall, and F1 score).

7) Fine-tune the model: Fine-tune the model by adjusting the hyperparameters and kernel function.

8) Deploy the model: Deploy the trained model to make predictions on new data.

3. Experimental Evaluation

3.1 Dataset Description

The data used in this assignment was collected from Kaggle, a platform used by data scientists to complete various challenges in the field. The link to the dataset is found here.

The dataset, refers to a telecommunications company, named Telco. It consists of 21 columns and 7043 rows. Within the dataset there is information regarding, the demographics of the customers, various subscriptions that the company offers their customers, total and monthly charges of the customer and churn. The customerID attribute is the primary key of the table as it is used to uniquely identify each instance.

Variable	Statistical Data type
CustomerID	Categorical
Gender	Categorical
SeniorCitizen	Categorical
Partner	Categorical
Dependents	Categorical
Tenure	Numerical
PhoneService	Categorical
MultipleLines	Categorical
InternetService	Categorical
OnlineSecurity	Categorical
OnlineBackup	Categorical

DeviceProtection	Categorical
TechSupport	Categorical
StreamingTV	Categorical
StreamingMovies	Categorical
Contract	Categorical
PaperlessBilling	Categorical
PaymentMethod	Categorical
MonthlyCharges	Numerical
TotalCharges	Numerical
Churn	Categorical

Table 1. Data Type Description

3.2 Methodology

The CRISP-DM Methodology was used to build the deliverable of the assignment (see Figure 3). It organizes the data science project in six phases (Business Understanding, Data Understanding, Data preparation, Modeling, Evaluation and Deployment).

Figure 3. Crisp-DM Methodology

3.2.1 Data Understanding:

For the second phase of the Crisp-DM model, we needed to understand our data set further. We had to visualize it to help us identify possible patterns and relationships hidden in the data. We created bar charts and KDEs (Kernel Density Estimation), to further understand the preferences of the clients regarding the products which Telco offers. During this procedure we noticed that the dataset is unbalanced. Also, we created Cramer's V Correlation heatmap to evaluate the correlation between the variables.

To create these visualizations we used software tools that are provided by various Python libraries such as pandas, scikit-learn, scipy, plotly, matplotlib and seaborn.

3.2.2 Data Preparation:

Third step of the Crisp-DM model refers to Data Preparation.The effectiveness of a machine learning algorithm is very much dependent on the data that the algorithm will be using to make the predictions.
This means that if we want to create a very effective algorithm, we will have to prepare our data in an adequate way. There are many steps in the Data Preparation phase. The following pipeline presents the high level overview.

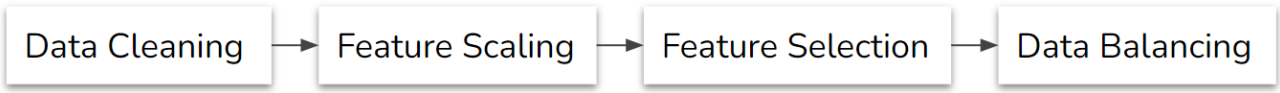

Figure 4. Data Preparation Pipeline

1. Concerning Data Cleaning, the following methods were used to detect potential outliers in our dataset: Z-score and IQR techniques. The intuition behind Z-score is to describe any data point by finding their relationship with the standard deviation and mean while IQR is used to measure variability by dividing a dataset into quartiles.

2. There are two common methods that can be applied for Feature Scaling, namely: Normalization and Standardization. The first technique implies scaling values to a standard range using min and max values while the second uses the mean and standard deviation.

3. There are many techniques and approaches for feature selection, such as: removing features with low variance, univariate feature selection (eliminate high correlation between variables) and exhaustive feature selection (pick best performing feature subset of a given size).

4. Regarding Data Imbalancing, there are various techniques that can be used to address this issue, such as: Random Oversampling, Random Undersampling and SMOTE. The first two focus on randomly adding/removing instances to/of the minority/majority class while the last one implies adding synthetic data points based on nearest neighbors.

3.2.3 Modeling:

Cross validation is an advanced form of data splitting, since it is also used to evaluate the performance of a machine learning model. The model divides the dataset into multiple subsets or "folds" and uses all folds but one to train and the remaining one as a test set.

This process offers a more accurate and robust estimate of the efficiency of the model, than a simple train-test set. Also, due to the multiple testing there are less chances of overfitting.
Since we have a small dataset, we want to utilize as much as possible. With Cross Validation this is possible because it uses all available data for training and testing.

In our project, we chose the best model by performing cross validation, after, we retrained the whole training and test set on the test set.

3.2.3.1 Hyperparameter Tuning

The values of the models' hyperparameters were tuned using the Grid Search technique based on the Recall score. Even if Grid Search implies a more costly execution than Random Search, we considered it to outperform the latter.

3.2.3.2 Algorithms

There are many algorithms from which we could choose from, but due to the nature of the customer churn problem, we chose to follow the ones mentioned below.

Decision Trees

The Decision trees algorithm is frequently used for classification problems where the goal is to predict a categorical or discrete outcome variable, such as whether a customer will churn or not. Also, they are easy to understand and visualize.

Support Vector Machines

Support vector machines are a popular algorithm for classification tasks, including predicting customer churn. They can also handle both categorical and numerical data. Additionally, SVMs are less prone to overfitting than other algorithms, making them a better pick for datasets with lesser training samples.

Logistic Regression

Logistic regression is a simple and interpretable algorithm. It is a regularly applied machine learning algorithm used for classification tasks, where the goal is to predict a binary or categorical outcome variable. In addition, logistic regression can work well with small datasets, as it is less prone to overfitting.

3.2.3.3 Evaluation

In order to measure the performance of our machine learning model we used the evaluation metrics, mentioned below.

Accuracy : Measures the proportion of correctly classified instances out of all instances in the dataset. (ACC= (TP + TN)/ (TP + TN + FP + FN))

Precision: Measures the proportion of true positive predictions out of all positive predictions. (PVV = TP / TP + FP)

Recall : measures the proportion of true positive predictions out of all actual positive instances in the dataset. (TPR = TP / TP + FN)

Specificity: measures the proportion of true negatives that are correctly identified by the model among all actual negatives. Thus, specificity tells us how well the model can identify negative cases. (SPC = TN / TN + FP) (Kumar, April 2023)

Matthews Correlation Coefficient (MCC): measures the quality of the classification, considering true and false positives and negatives. The values range from [-1, 1], with -1 meaning a wrong prediction, 0 being random and +1 having perfect predictions.

$$MCC = (TP*TN - FP*FN) / \sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}$$

For our model, we decided that Recall was more important to us than precision. We made that decision, based on the theory that the overall costs of keeping the current customers satisfied enough to continue their subscription, are less than attracting new customers. Also, even though recall is important to us, we still wanted an overall better score, than an overall lower score and better recall. In addition, we chose not to take accuracy into consideration when choosing the best tuned candidate because the calculated values of this metric can often be deceiving.

3.3 Process & Results

3.3.1 Data Understanding

3.3.1.1 Data Requirements

To have a better understanding of the available data content, we decided to design a Star Schema. This type of model can be used to represent a data mart in a warehouse environment, depicting the visual relationship among different entities (see *Figure 5*).

It has one fact table (FactTransaction), which contains transaction measurements and three-dimension tables (DimAccount, DimService, DimCustomer), that include descriptive attributes.

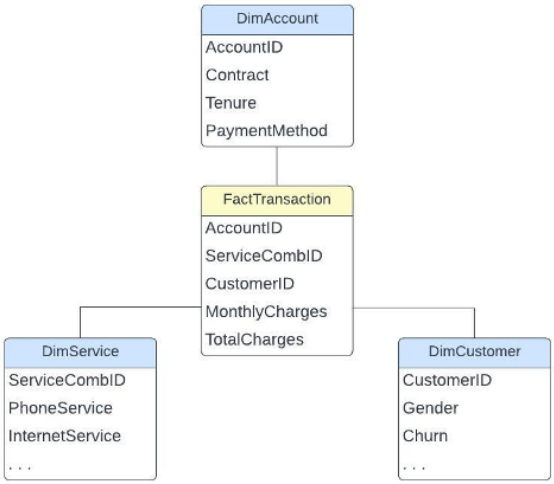

Figure 5. Star Schema

3.3.1.2 Data Insertion

The CSV file was imported as a dataframe in the Visual Studio - Jupyter Notebook environment using relevant Python packages.

3.3.1.3 Data Description

By analyzing the distribution of the target variable of this assignment – Churn *(Figure 6),* we can state that it is imbalanced as the class values are not represented equally. It seems that 27% of the customers left the business last month while 73% remained.

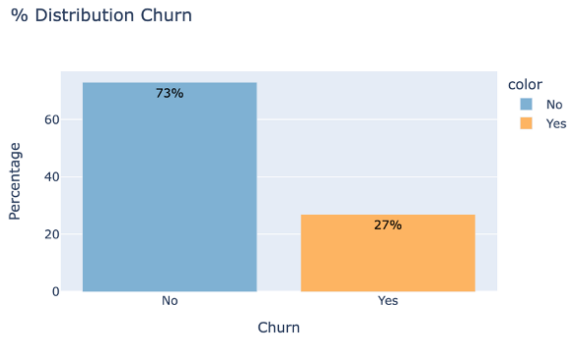

Figure 6. Target Distribution

Since the imbalance nature of the dependent variable can lead to biased results, we took additional actions to balance it in the Data Preparation phase of the methodology.

3.3.1.4 Data Exploration

The data cleaning task of the dataframe involved handling missing values, reformatting, converting data types, and detecting potential outliers. The empty values of the "TotalCharges" variable were replaced with 0 as they were associated with the customers having a 0-month tenure period. When it comes to finding potential abnormal values, we could not detect anything after applying the previously mentioned statistical techniques.

The bivariate and multivariate analysis steps of the data analysis task, done with various graphical techniques (histograms, density plots, clustered column charts) suggested several correlations between the target and the independent variables.

Customers with a higher tenure are less likely to churn while recent clients are more likely to leave the business. Also, purchasers with highly monthly charges (70-100) are more likely to churn, compared with clients that had lower monthly charges (*see Figure 7*).

Furthermore, the relationship between “Tenure” and “Monthly Charges” attributes indicates that as the tenure increases, the monthly charges also increase linearly for both groups. However, the line representing customers who churned is consistently above the line representing customers who did not churn. This suggests that clients who churned are charged more than customers who did not churn, for the same tenure. Several reasons could explain this case, such as the purchasers who churned having a higher rate plan.

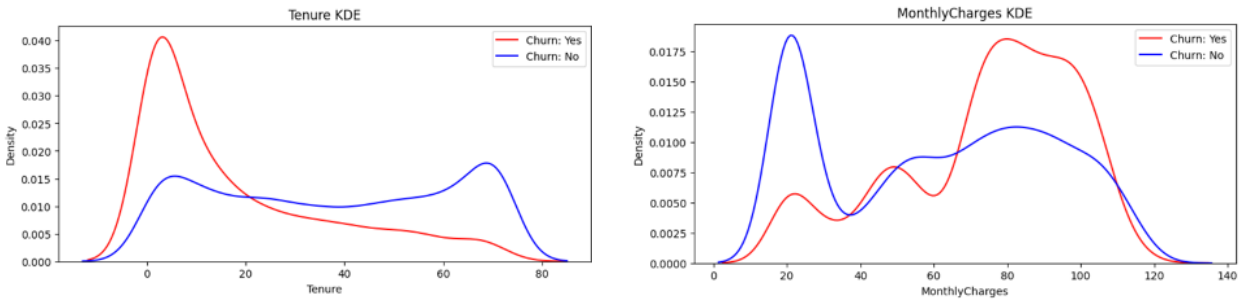

Figure 7. Bivariate Analysis between Churn and Tenure, Monthly Charges

The bivariate step of the data analysis task was also expressed using several statistical correlation coefficient methods, depending on the data type and properties of the attributes.

Since Pearson’s correlation coefficient assumes a Gaussian distribution for the numerical data and “Tenure” and “Monthly Charges“ do not follow such a shape, we used the Spearman’s coefficient and generated the scores, all showing a weak relationship with the target. Regarding the categorical attributes, by using Cramer’s V technique we received a best moderate score for “Contract” (0.40). *Table 1* contains the ranking of the best correlation coefficient scores. As the reader can observe from these results, we cannot declare a strong relationship between “Churn” and any independent variable.

Contract	0.40
Tenure	-0.35
Online Security	0.34
Monthly Charges	0.19

Table 2. Spearman’s coefficient and Cramer’s V Ranking: Churn – numerical and categorical attributes

Overall, the outcome of the Data Understanding phase consists of several discovered graphical patterns concerning the relationship of “Churn” with “Monthly Charges” and “Tenure”. Their significance can be further analyzed using statistical techniques.

3.3.2 Data Preparation

3.3.2.1 Data Cleaning

As mentioned, Machine learning (ML) algorithms require working with numbers. As such, our data-set consists of both numerical and categorical features. Luckily, it does not make use of arbitrary text, meaning that for the categorical features they are one of a limited amount of values. As such we can encode these values. We applied *Label encoding* for features with binary values, such a ‘Yes’ or ‘No’, this was the case for a column like: “isSenior”. However, for other non-binary categorical columns such as “InternetService”, this would not work as there are 3 possible values, namely: ”DSL”, “Fiber_optic” or “No”. As such, a label encoder would create a range of values for the labels. This however could make the ML algorithm assume that two nearby values are similar, when they are not, they are rather distinct. To address this, we used *One Hot Encoding*. This process creates dummy binary attributes for columns. This does result in a larger amount of columns in our data frame than at first, but that is an essential step nonetheless.

3.3.2.2 Feature Scaling

In our case, we went with the first option, Normalization. This is due to the fact that when we examined the distribution of all of our numerical features, we discovered that none of them followed a normal distribution. As can be seen in the KDE (Kernel Density Estimator) plots below.

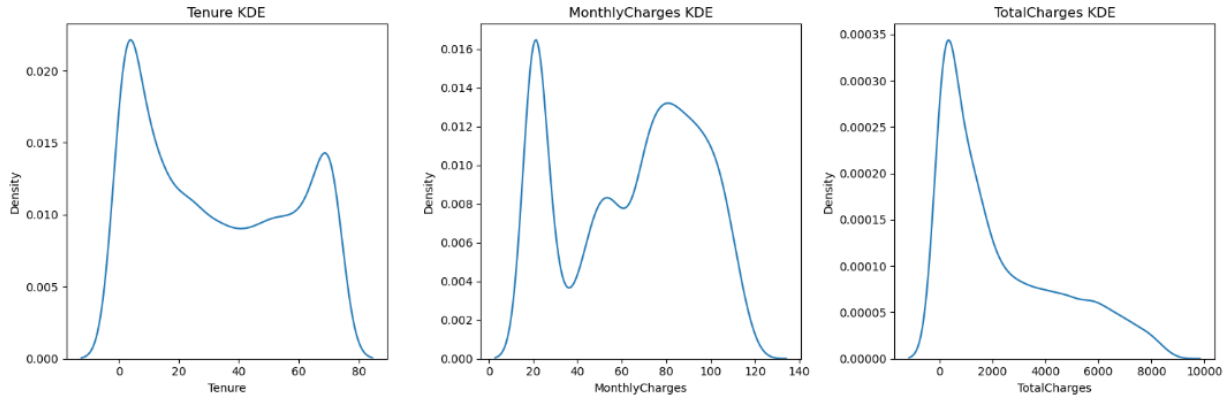

Figure 8. KDEs for Tenure, MonthlyCharges and TotalCharges

As a result, we chose normalization over standardization because standardization requires that the data be normally distributed (Hartmann, K., Krois, J., Waske, B., 2018). This is also supported by an inspection of the ranges of values of each variable, as the scales vary a lot. As a result, using normalization we rescale the data to a range of [0,1].
In retrospect, we may have used different ways to fix the skewness of TotalCharges, such as log or square-root scaling.

3.3.2.3 Feature Selection

There are many techniques and approaches for feature selection. The one that we mostly replied on was removing highly correlated features that we could observe from Cramer V's correlation matrix (Catolino Gemma, *Classification,* March 2022). We did also use Low-Variance Feature Selection, which returned that Total charges should be dropped, but from the insights that we gained from our Data Understanding process we decided to leave it. We also considered using an Exhaustive Feature Selection technique, but this was out of the scope of this assignment. Since, as a result of the feature encoding, there are simply too many combinations of features possible.

As mentioned above, we used the Cramer V's correlation matrix. As our target was to evaluate the correlation between categorical features (of nominal values). As a result, this is what our result Correlation matrix looked like, after removing all features that had a correlation with another over the *0.1* threshold.

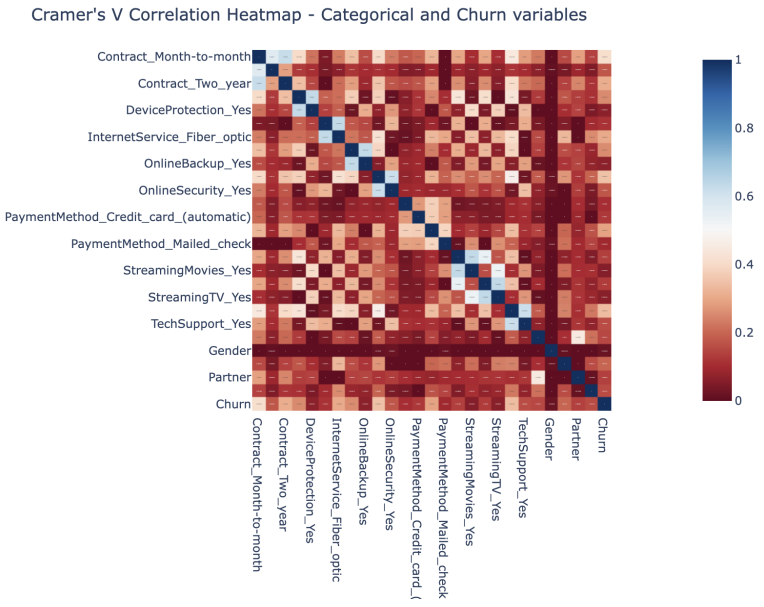

Figure 9. Cramer's V Correlation Heatmap- Categorical and Churn variables

3.3.2.4 Data Balancing

Data imbalance is a problem that needs to be addressed, since our target variable of churn is in the minority class. With 1869 out of 7043 records are 'churned', therefore our target minority class only encompasses ~23% of the entire dataset.

We decided to focus on techniques ROS and SMOTE, as we felt that we do not have enough data to try to continue with random under sampling. As such, moving forward, for the model section we only will be evaluating using the Random Over Sampler and SMOTE.

3.3.3 Data Modeling & Evaluation

Finally, with our newly cleaned dataset, we were able to begin the data modeling process. This is a crucial point in the CRISP-DM process as it involves the actual development of the predictive model that will best generalize for yet unseen data.

3.3.3.1 Baseline

Evaluating a baseline model before applying any balancing methods or tuning hyperparameters is very important. This gives us the point of reference when evaluating the performance of our binary classifier. A lack of a baseline model, would make it potentially difficult to assess whether our more sophisticated and developed model actually performs better than our baseline. In order to achieve this, we assigned all the points to the majority class and computed the Accuracy (0.73) and Recall (0) scores.

3.3.3.2 First Version – Unenhanced ML Candidates

All the algorithms described in the Methodology chapter were used to build 3 different models. After evaluating their performance, we concluded that the Logistic Regression model presented the best results out of the algorithms, having a higher Recall (0.54) and MCC value (0.47) (see Figure x).

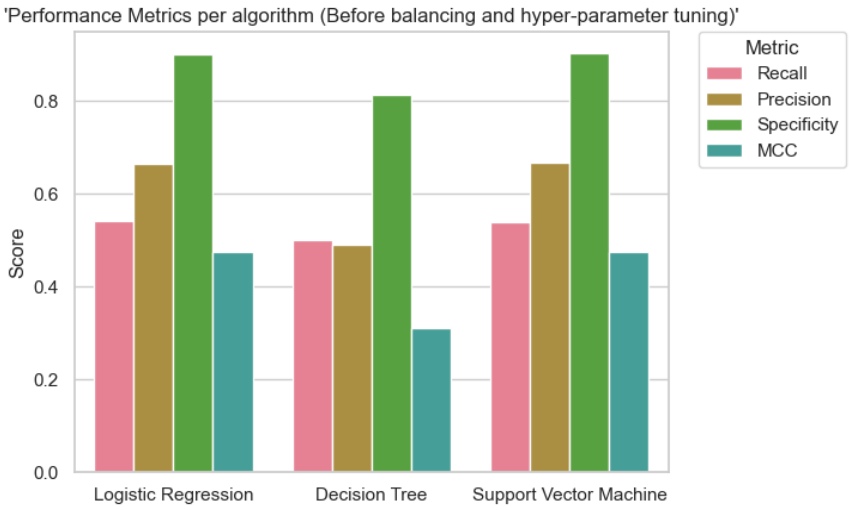

Figure 10. First Version- Unenhanced ML candidates

3.3.3.3 Second Version – Hyperparameter tuned and Imbalance techniques ML Candidates

We performed the Hyperparameter Tuning process after separately using SMOTE and Random Oversampling techniques, to enhance the candidates' performance (see Figure x). Concerning the Random Oversampling method, the findings indicated a massive increase of the Recall scores for all 3 models but also a slight increase of the MCC values. SVM is proven to be the model having the best combination between Recall (0.80), Precision (0.51) and MCC (0.47) scores.

Regarding the results gained by using the SMOTE technique, we observed a worse overall performance for all 3 models.

By looking at the figures above and comparing with the first version of the models, we could also observe that the Specificity scores decreased, meaning that the application of the imbalanced techniques successfully worked (the classifiers do not only predict the majority class).

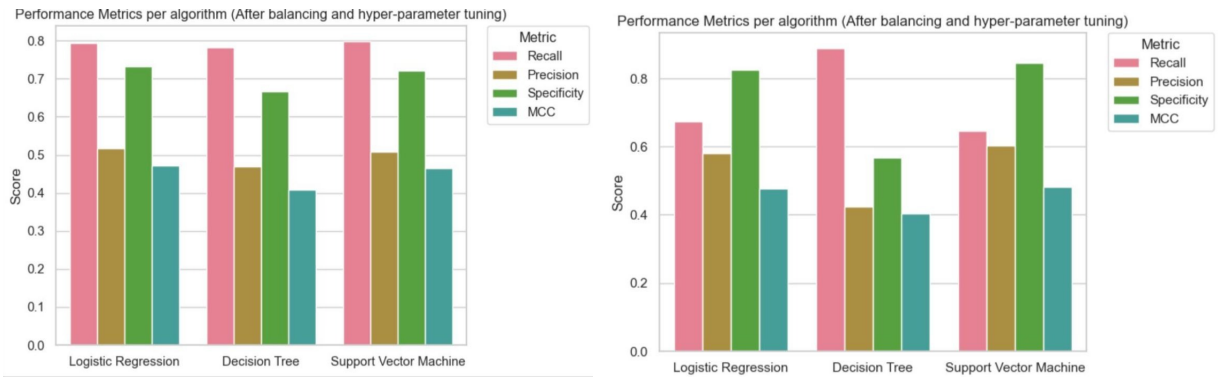

Figure 11. Second Version - Enhanced ML candidates: Random Oversampling & SMOTE

Since our goal was to maximize the Recall and also have the best possible overall performance between the other metrics, we decided to select the SVM tuned model, with the applied Random Oversampling technique as the best candidate.

3.3.3.4 Final classifier

Having the best-tuned classifier in place, we retrained it on the full training set and tested it on the test set (see Figure 12). The findings below indicate a similar performance to the one obtained using cross-validation. Comparing these results with the baseline, we can state that our classifier is better performing since we have a slightly higher Accuracy and a better Recall/Sensitivity.

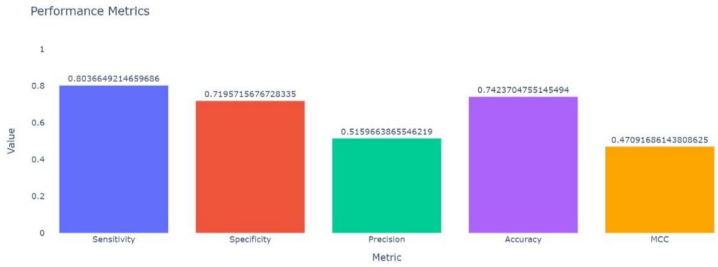

Figure 12. Final Model. Evaluation Metrics including Accuracy

4. Discussion

As mentioned before, SVM or Support Vector Machine is a linear model for classification. It was a good choice of algorithm for our purposes. Similarly, to other algorithms, there are certain drawbacks. Due to our constraints of time and processing power, we made certain choices for receiving results faster. As such, the accuracy of our model is potentially not as high as it could be, a topic further explored in Future Work.In short, there is an accuracy and speed tradeoff. As such, a different kernel function could be better suited to generalize for our dataset. During our hyper parameter tuning process, we targeted maximizing the recall (Sensitivity**)**, which was done successfully as seen in the previous figure, seeing a jump from under 0.6 to ~0.8 using ROS SVM. Another issue that can often occur with SVM is the problem of overfitting. It is not so easy to spot overfitting for SVM as it is for Decision trees and Logistic regressions. Therefore, we paid close attention during our performance evaluation process and applied cross validation as opposed to a simple train, test and validation split. As with this we were able to evaluate performance over different partition arrangements.

The Logistic Regression model was the second best performing algorithm after SVM, while the Decision Tree was the last. Even though Logistic Regression had a close performance to SVM, we decided to pick the model optimizing the Recall score.

Given all of this, how does our exploration help us answer our main research question? We set out to evaluate, to what extent machine learning can be used to predict customer churn. In general, we find that machine learning can be a good method to do so. Our model based on the Support Vector Machine (SVM) is suitable for prediction churning customers. We quantified this by our target metrics, which were significant enough to consider. It is also important to mention that a machine learning model can analyze large amounts of data in a quicker and more efficient manner than that of manual analysis. It is capable of identifying patterns and relationships that an ordinary human, even with enough domain knowledge would not be able to recognise or detect, which could lead to more insight than other techniques for detecting churn. Such as defining business rules (Kniazieva, June 2022). As this is a time consuming and complex process, which still may not always capture the nuances of customer behavior.

As such, we find that this also is supporting our business verdict. If Telco is considering a solution for identifying customers that are at risk of churning, they should consider creating retention campaigns powered by a supervised machine learning model that would predict which customers are at risk of churning, such that a customer reactivation campaign can begin before they are lost.

5. Conclusion

In conclusion, we find that we successfully were able to deliver on our research and business objectives. Namely, following the CRISP DM model, we followed each phase which was a natural way to progress in the development of a machine learning model for a real world scenario. Throughout, we explored and explained in detail each of the algorithms (Decision Trees, Logistic Regression and Support Vector Machines) we used and were able to evaluate the results of each model. As such, these models were trained to maximize the Recall metric, which captures the model's ability to identify true positives, which are customers who are likely to leave the company. As such, we found that our SVM model was the most capable of predicting this with a score of {score here}. So, in terms of the business value proposition, we conclude that they may consider using such an algorithm to predict which of their customers are at risk of leaving. In terms of research value, we consequently found that machine learning can be used for prediction of such customer behaviors; such as churn.

6. Future work

As this was a project for our Introduction to Machine Learning course, this means that our resources (e.g. time) were limited. So, for future work we believe that we could apply other techniques (such as Exhaustive Feature Selection), that way it is possible that we would have better results. We also could look at other methods or algorithms to consider, such as using Random Forests for feature selection. It is also possible to apply other algorithms which use unsupervised learning techniques, such as k-means clustering, to provide further insight.

Also, it must be mentioned that the findings of our model are limited to the quality of the data that we train and test it on. So, we believe that if we had a more dense and extensive dataset, we would consequently be able to create more accurate predictions on whether a customer would churn. In return we would be able to provide the company with a more detailed profile of the customers that are at risk of churning. One of the options for this to be possible is by combining more datasets with similar classes or by selecting more data.

7. References

Batra, M., Agrawal, R. (2018). Comparative Analysis of Decision Tree Algorithms. In: Panigrahi, B., Hoda, M., Sharma, V., Goel, S. (eds) Nature Inspired Computing. Advances in Intelligent Systems and Computing, vol 652. Springer, Singapore. https://doi.org/10.1007/978-981-10-6747-1_4 (Date accessed, April 26th, 2023)

Bhuvaneswari Gopalan (December, 2020) What is Entropy and Information Gain? How are they used to construct decision trees?, *Numpy Ninja.* https://www.numpyninja.com/post/what-is-entropy-and-information-gain-how-are-they-used-to-construct-decision-trees (Date accessed, April 26th, 2023)

Brownlee Jason (April, 2014) Logistic Regression Tutorial for Machine Learning, *Machine Learning Mastery.* https://machinelearningmastery.com/logistic-regression-tutorial-for-machine-learning (Date accessed, April 26th, 2023)

Carter Tom (September, 2014), An introduction to information theory and entropy. https://csustan.csustan.edu/~tom/Lecture-Notes/Information-Theory/info-lec.pdf (Date accessed, April 26th, 2023)

Catolino Gemma (April 2023), *Artificial Neural Networks.* *https://tilburguniversity.instructure.com/courses/11206/files/folder/Lecture%207?preview=2389692* (Date accessed, April 26th, 2023)

Catolino Gemma (March 2023), *Decision Trees.* *https://tilburguniversity.instructure.com/courses/11206/files/folder/Lecture%205?preview=2341410* (Date accessed, April 26th, 2023)

Catolino Gemma & Pecorelli Fabiano (February 2023), *Classification.* *https://tilburguniversity.instructure.com/courses/11206/files/folder/Lecture%203?preview=2290771* (Date accessed, April 26th, 2023)

Catolino Gemma & Pecorelli Fabiano (February 2023), *Data Quality & Feature Engineering.* *https://tilburguniversity.instructure.com/courses/11206/files/folder/Lecture%202?preview=2275582* (Date accessed, April 26th, 2023)

Catolino Gemma & Pecorelli Fabiano (January 2023), *Introduction to Machine Learning.*

https://tilburguniversity.instructure.com/courses/11206/files/folder/Lecture%201?preview=2262066 (Date accessed, April 26th, 2023)

Catolino Gemma & Pecorelli Fabiano (March 2023), *Regression.* *https://tilburguniversity.instructure.com/courses/11206/files/folder/Lecture%204?preview=2325608* (Date accessed, April 26th, 2023)

Chamalka Kaumadie (June, 2019). Decision Tree in Machine Learning, *Medium*. https://kaumadiechamalka100.medium.com/decision-tree-in-machine-learning-c610ef087260. (Date accessed, April 26th, 2023)

Chandrasekaran Maran (November, 2021), Logistic Regression for Machine Learning, *Capital One.* https://www.capitalone.com/tech/machine-learning/what-is-logistic-regression/ (Date accessed, April 26th, 2023)

Cortes Corinna , Vapnik Vladimir (1995), Support-Vector Networks, *Kluwer Academic Publishers*. http://image.diku.dk/imagecanon/material/cortes_vapnik95.pdf (Date accessed, April 26th, 2023)

Dietterich Tom (September, 1995), Overfitting and Under Computing in Machine Learning, *AMC Digital Library.* https://dl.acm.org/doi/pdf/10.1145/212094.212114 (Date accessed, April 26th, 2023)

Ellerman David (June. 2017), Logical Information Theory: New Logical Foundations for Information Theory, *Philosophy Department, University of California at Riverside.* http://philsci-archive.pitt.edu/13213/1/Logic-to-information-theory3.pdf (Date accessed, April 26th, 2023)

Gallo Amy(October, 2014), The Value of Keeping the Right Customers, *Harvard Business Review.* https://hbr.org/2014/10/the-value-of-keeping-the-right-customers (Date accessed, April 26th, 2023)

Géron Aurélien, Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow, 2nd Edition, O'Reilly Media, Inc., June 2019

Guo Anyi (September, 2021), Pearson vs. Spearman Correlation: What's the difference?, *Medium.*

https://anyi-guo.medium.com/correlation-pearson-vs-spearman-c15e581c12ce (Date accessed, April 26th, 2023)

Gupta Prashant (May, 2017), Decision Trees in Machine Learning, *Towards Data Science.* https://towardsdatascience.com/decision-trees-in-machine-learning-641b9c4e8052 (Date accessed, April 26th, 2023)

Hartmann, K., Krois, J., Waske, B. (2018): E-Learning Project SOGA: Statistics and Geospatial Data Analysis. Department of Earth Sciences, Freie Universitaet Berlin. *https://www.geo.fu-berlin.de/en/v/soga/Basics-of-statistics/Continous-Random-Variables/The-Standard-Normal-Distribution/Standardizing-a-Normally-Distributed-Variable/index.html* (Date accessed, April 26th, 2023)

Kniazieva Yuliia (June, 2022), Pattern Recognition and Machine Learning: A Perfect Match?, *Label your Data.* *https://labelyourdata.com/articles/pattern-recognition-in-machine-learning* (Date accessed, April 26th, 2023)

Kumar Ajitesh (April, 2023), Machine Learning – Sensitivity vs Specificity Difference, *Data Analytics.* *https://vitalflux.com/ml-metrics-sensitivity-vs-specificity-difference/* (Date accessed, April 26th, 2023)

Kumar, Narender (March, 2023), Logistic Regression Explained with Examples. *Spark By {Examples}.*

https://sparkbyexamples.com/machine-learning/logistic-regression-explained-with-examples/. (Date accessed, April 26th, 2023)

Logistic Regression in Machine Learning (March, 2023), *GeeksForGeeks.* https://www.geeksforgeeks.org/understanding-logistic-regression/ (Date accessed, April 26th, 2023)

Machine Learning - Logistic Regression, *Tutorials point.* https://www.tutorialspoint.com/machine_learning_with_python/machine_learning_with_python_classification_algorithms_logistic_regression.htm (Date accessed, April 26th, 2023)

Nielsen Micheal (December, 2019) Improving the way neural networks learn, *Neural networks and deep learning.* http://neuralnetworksanddeeplearning.com/chap3.html (Date accessed, April 26th, 2023)

Pant Ayush (January, 2022), Introduction to Logistic Regression, *Towards Data Science.* https://towardsdatascience.com/introduction-to-logistic-regression-66248243c148 (Date accessed, April 26th, 2023)

Pecorelli Fabiano (March 2023), *Support Vector Machines.* *https://tilburguniversity.instructure.com/courses/11206/files/folder/Lecture%206?preview=2350414* (Date accessed, April 26th, 2023)

Rashida048 (April, 2022), Simple Explanation on How Decision Tree Algorithm Makes Decisions, *Regenerative.* https://regenerativetoday.com/simple-explanation-on-how-decision-tree-algorithm-makes-decisions/#:~:text=The%20decision%20tree%20algorithm%20works,decisions%20based%20on%20the%20conditions (Date accessed, April 26th, 2023)

Schober, Patrick MD, PhD, MMedStat; Boer, Christa PhD, MSc; Schwarte, Lothar A. MD, PhD, MBA (May, 2018), Correlation Coefficients: Appropriate Use and Interpretation, *Anesthesia & Analgesia.* https://journals.lww.com/anesthesia-analgesia/fulltext/2018/05000/correlation_coefficients__appropriate_use_and.50.aspx (Date accessed, April 26th, 2023)

1.1. Linear Models, *scikit.* https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression (Date accessed, April 26th, 2023)

1.4. Support Vector Machines, *scikit.* https://scikit-learn.org/stable/modules/svm.html (Date accessed, April 26th, 2023)