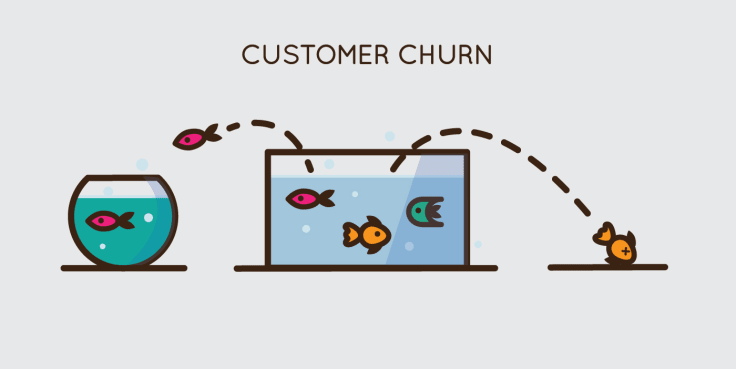
**PREDICTING CUSTOMER CHURN IN THE TELECOM SECTOR**



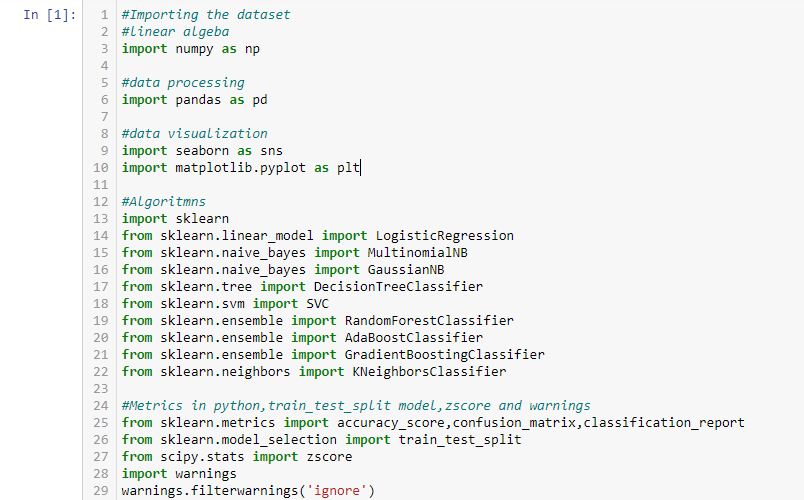
In this blog-post, I will go through the whole process of creating a machine learning model on the famous Telecom customer churn dataset, which is used by many people all over the world. It provides information on whether a particular customer will churn or not in the telecom sector, summarized according to Internet Service, Paperless Billing, Payment Method, Monthly Charges and many more factors.

In this project, we will examine customer data from IBM Sample Data Sets with the aim of building and comparing several customer churn prediction models.

**What is Customer Churn?**

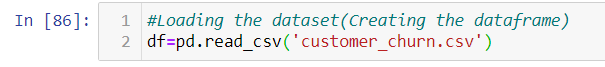
***Customer churn occurs when customers or subscribers stop doing business with a company or service. Also known as customer attrition, customer churn is a critical metric because it is much***[***less expensive***](http://www.invespcro.com/blog/customer-acquisition-retention/)***to retain existing customers than it is to acquire new customers – earning business from new customers means working leads all the way through the sales funnel, utilizing your marketing and sales resources throughout the process. Customer retention, onthe other hand, is generally more cost-effective as we’ve already earned the trust and loyalty of existing customers.***

**Importing the Libraries**

****

*Exhibit 1*

**Getting the Data**

****

*Exhibit 2*

# Data Exploration/Analysis

# C:\Users\Neeti\Documents\3.png

*Exhibit 3*

# The dataset consists of 7043 rows and 21 columns(features) including the target variable.

# C:\Users\Neeti\Documents\4.png

*Exhibit 4*

# The features are listed above in the screenshot including the dependent feature. Let’s check the datatypes of the features.

# C:\Users\Neeti\Documents\5.png

*Exhibit 5*

# C:\Users\Neeti\Documents\6.png

*Exhibit 6*

# C:\Users\Neeti\Documents\7.png

*Exhibit 7*

# From the table above, we can note a few things. First of all, we **need to convert a lot of features into numeric** ones later on, so that the machine learning algorithms can process them. Furthermore, we can see that the **features have widely different ranges**, that we will need to convert into roughly the same scale.

# Let’s check if there are any null values present in the dataset.

# C:\Users\Neeti\Documents\8.png

*Exhibit 8*

# There are no null values in the dataset as we have seen from above. I am also checking it with a heatmap.

# C:\Users\Neeti\Documents\9.png

*Exhibit 9*

# By looking at the heatmap, we can say that there are no null values present. If any null values were present, then we could have seen some short white spaces in between the heatmap.

# Data Pre-processing

# Converting features into numeric:

# For categorical type of data,I use LabelEncoder and for continuous type, I use OrdinalEncoder. Let’s start the conversion process. Firstly, I will check the value counts of the features and then I will convert it into numeric.

# C:\Users\Neeti\Documents\10.png

*Exhibit 10*

# ‘Gender’ and ‘SeniorCitizen’ –both features are categorical and I have encoded the object data into numeric form but ‘SeniorCitizen’ feature is already encoded into 0 and 1.

# Now I will perform encoding of all the features wherever required.

# Partner:

# C:\Users\Neeti\Documents\11.png

*Exhibit 11*

# MultipleLines:

# C:\Users\Neeti\Documents\12.png

*Exhibit 12*

# InternetService and StreamingTV:

# C:\Users\Neeti\Documents\13.png

*Exhibit 13*

# Till now, I am handling all the categorical data. Let’s explore some more data.

# Contract and PaperlessBilling:

# C:\Users\Neeti\Documents\14.png

*Exhibit 14*

# PaymentMethod:

# C:\Users\Neeti\Documents\15.png

*Exhibit 15*

# I think all the data are encoded and are ready for Machine Learning algorithms. But before moving forward once I will check the datatypes of the features again.

# C:\Users\Neeti\Documents\16.png

*Exhibit 16*

# From the above, I can see ‘TotalCharges’ feature is still no encoded, the data in the column are mostlycontinuous type data, so I go for OrdinalEncoding.

# C:\Users\Neeti\Documents\17.png

*Exhibit 17*

# Now, I will drop the ‘customerID’ feature as it is not playing any significant role.

# C:\Users\Neeti\Documents\18.png

*Exhibit 18*

# C:\Users\Neeti\Documents\19.png

*Exhibit 19*

# I have dropped the ‘customerID’ column from the dataset and checked the first five rows of the DataFrame. Finally, my DataFrame is ready for Machine Learning processes. Moving forward, I will see the statistical summary of the newly encoded data and also the correlation matrix.

# Statistical Summary:

# C:\Users\Neeti\Documents\n.png

*Exhibit 20*

# Correlation matrix:

# C:\Users\Neeti\Documents\21.png

*Exhibit 21*

# C:\Users\Neeti\Documents\22.png

*Exhibit 22*

# From the correlation matrix, it is seen that ‘MonthlyCharges’, ‘PaperlessBilling’, ‘SeniorCitizen’, ‘InternetService’, ‘MultipleLines’, ‘TotalCharges’, ‘PhoneService’, ‘Gender’ is positively correlated with the target variable ‘Churn’.

# I have checked if any outlier is present using boxplots, but there are no outliers present. And also I have checked for any skewed data but there is no skewness present in the data.

# Splitting the dependant and independent variable in x and y:

# C:\Users\Neeti\Documents\s.png

*Exhibit 23*

# Scaling the data to a standard form:

# C:\Users\Neeti\Documents\scale.png

*Exhibit 24*

# EDA Concluding Remark

# After going through data analysis and data processing, I can conclude that the raw data that I have received is now cleaned and is ready for Machine Learning process. Steps that I follow are listed below:

# At the beginning, I have analysed the data by checking its shape, its datatypes and information regarding presence of null values.

# After checking for null values, I have seen that there are no null values present. So there is no need of using SimpleImputer method.

# While checking for datatypes, I have seen many object type columns are present, I have encoded them into numeric so that Machine can understand. The encoding process that I have used for categorical type data is LabelEncoder and for continuous type data I have used OrdinalEncoder.

# Then I have checked the statistical summary of the dataset and also checked the correlation between the features and the target variable using heatmap and correlation matrix.

# When my DataFrame gets ready for Machine Learning process, I have splitted the independent variables and target variable into x and y.

# Then I scaled my data into a standard form using StandardScaler.

# Building Machine Learning Models

# Now I will train several Machine Learning models and compare their results. Later on, I will use cross validation.

# C:\Users\Neeti\Documents\mnew.png

*Exhibit 25*

# C:\Users\Neeti\Documents\anew.png

*Exhibit 26*

# As we can see,SVC(Support Vector Classifier) is giving a good accuracy of 82% among all the classification models at a random state 59. But, let us check, how SVC performs, when we use cross validation.

# K Fold Cross-Validation:

# K-Fold Cross Validation randomly splits the training data into **Ksubsets called folds**. Let’s image we would split our data into 4 folds (K = 5). Our model would be trained and evaluated 5 times, using a different fold for evaluation everytime, while it would be trained on the remaining 4 folds.

# The image below shows the process, using 4 folds (K = 4). Every row represents one training + evaluation process. In the first row, the model get’s trained on the first, second and third subset and evaluated on the fourth. In the second row, the model get’s trained on the second, third and fourth subset and evaluated on the first. K-Fold Cross Validation repeats this process till every fold acted once as an evaluation fold.

# https://miro.medium.com/max/2298/1*HzpaubLj_o-zt1klnB81Yg.png

*Exhibit 27*

# The code below performs K-Fold Cross Validation on our SVC model, using 5 folds (K = 5 or cv=5).

# C:\Users\Neeti\Documents\cross.png

*Exhibit 28*

# After cross validation, we get the actual accuracy of the model,i.e. 80%. Before it was 82% because of overfitting.Now,I will try to increase its performance even further in the following section.

# Support Vector Classifier:

# “Support Vector Machine”(SVM) is supervised machine learning algorithm which can be used for both classification and regression challenges. Support Vectors are simply the co-ordinates of individual observation. The SVM classifier is a frontier which best segregates the two classes (hyper-plane/ line).

# https://miro.medium.com/max/1626/1*kzdqdDUTwNsAkVZNLQAPvQ.png

*Exhibit 29*

# Hyperparameter tuning:

# A Machine Learning model is defined as a mathematical model with a number of parameters that need to be learned from the data. By training a model with existing data, we are able to fit the model parameters.However, there is another kind of parameters, known as Hyperparameters, that cannot be directly learned from the regular training process. They are usually fixed before the actual training process begins. These parameters express important properties of the model such as its complexity or how fast it should learn.

# Models can have many hyperparameters and finding the best combination of parameters can be treated as a search problem. Two best strategies for Hyperparameter tuning are:

# GridSearchCV:In GridSearchCV approach, machine learning model is evaluated for a range of hyperparameter values. This approach is called GridSearchCV, because it searches for best set of hyperparameters from a grid of hyperparameters values.

# RandomizedSearchCV:RandomizedSearchCV solves the drawbacks of GridSearchCV, as it goes through only a fixed number of hyperparameter settings. It moves within the grid in random fashion to find the best set hyperparameters. This approach reduces unnecessary computation.

# Now we will tune the hyperparameters and check if we can increase the model's accuracy.

# C:\Users\Neeti\Documents\g.png

*Exhibit 30*

# The accuracy after hyperparameter tuning is also 80%. The best parameters are {‘C’:1,’kernel’:’rbf’}.

# Now,I will fit the hyperparameters into the SVC model. After that I will save the model.

# C:\Users\Neeti\Documents\save.png

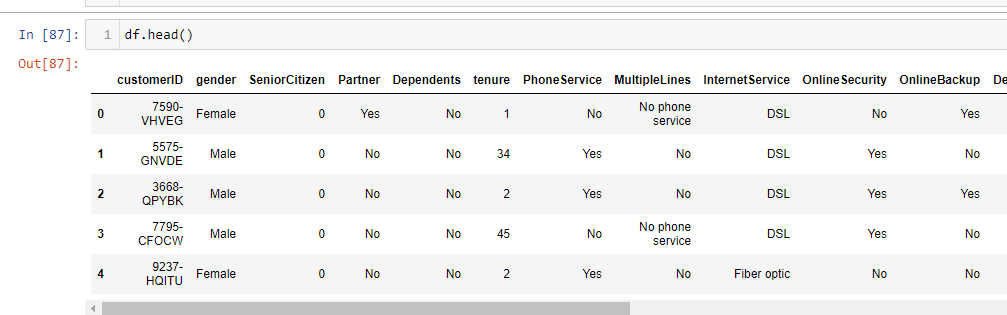
*Exhibit 31*

# Conclusion

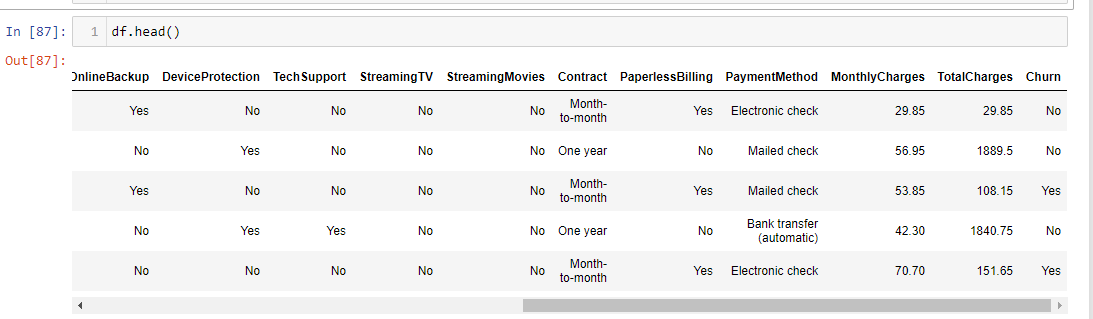
We started with the data exploration where we got a feeling for the dataset, checked about missing data and learned which features are important. During this process we used seaborn and matplotlib to do the visualizations. During the data pre-processing part, we converted features into numeric ones,grouped values into categories and dropped feature which does not play an important role in the process. Afterwards we started training 5 different machine learning models, picked one of them (SVC) and applied cross validation on it. Then we discussed how SVC works, looked at the importance it assigns to the different features and tuned its performance through optimizing it’s hyperparameter values.

Below you can see a before and after picture of the “df” dataframe:

**Before:**

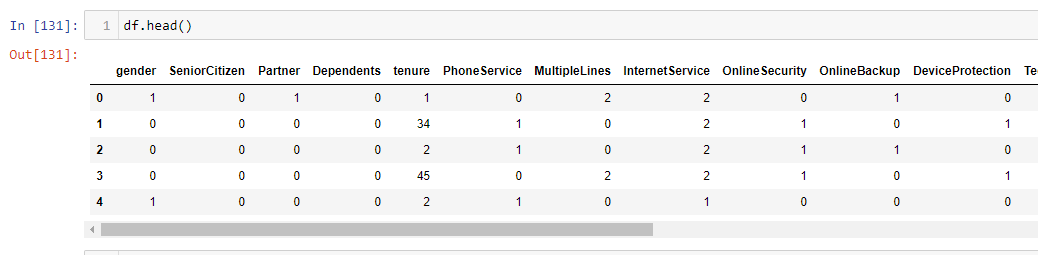
****

*Exhibit 32*

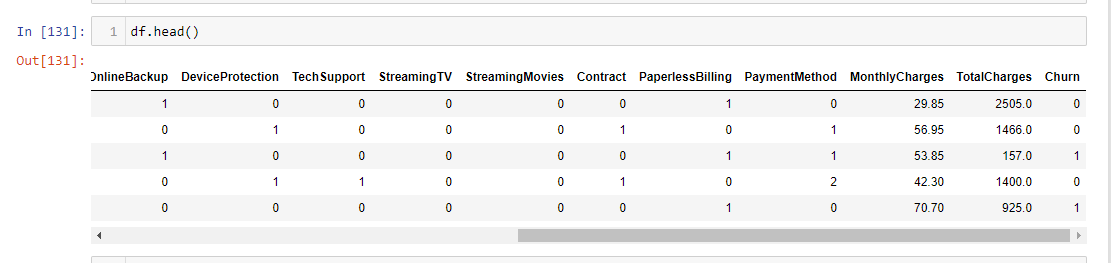
****

*Exhibit 33*

**After:**

****

*Exhibit 34*

****

*Exhibit 35*

# Naturally, there is still room for improvement, like doing a more extensive feature engineering, by comparing and plotting the features against each other and identifying and removing the noisy features.