Machine Learning Engineer Nanodegree

Capstone Project

Mohamed El-Deeb
November 11, 2019

I. Definition

Project Overview

With the wide spread of smart phones telco companies are now a more important than ever and have a high number of customer but as the telco market matures and big companies compete in prices, service and quality of service every little advantage is needed for any company and since customers are the most important aspect that drives these companies it is highly important to retain customers in the company hence a churn model to help telco companies predict which customers are in danger of leaving the company is of extreme importance now more than ever.
The primary motivation behind this model is the dire need of businesses to retain existing customers, coupled with the high cost associated with acquiring new ones. A review of the field has revealed a lack of efficient, rule-based Customer Churn Prediction (CCP) approaches in the telecommunication sector in my country.

The proposed solution is to create classification model to classify customers whether they are going to churn or not and try to profile the churn types. I am going try different models using methods to tune these models.

Problem Statement

Given a dataset of customer features, an algorithm needs to be developed to classify whether a customer will churn or not and try to determine what are the most contributing features in their churn while also investigating the reason behind the churn of said customers and if possible introduce a solution for different types of churn for the business.

Metrics

This study will be evaluated with regards to the model ability to accurately predict the customers that are in danger of churning though typically classification model are measured using accuracy, which is defined as the number of correct predictions from all the predictions made but since the business is mainly concerned with lost customer since the lost customers present a threat to revenue and the cost of regaining a customer is always greater that the cost of keeping one and since the data is unbalanced.

So I will use I will use both F-Score and sensitivity (Recall) as my main evaluation metrics and accuracy and precision as supporting metrics my model

Where Precision is the number of True Positives (TP) divided by the number of True Positives (TP) and False Positives (FP) where a low precision indicates a large number of False Positives.

$$precision = \frac{TP}{TP + FP}$$

Recall is the number of True Positives (TP) divided by the number of True Positives (TP) and the number of False Negatives (FN). A low recall indicates a large number of False Negatives.

$$recall = \frac{TP}{TP + FN}$$

Since we would rather make false positive predictions than false negative ones in our case I will use the f-beta metric where I will set beta as 1.5 to weight the metric towards recall more that precision.

$$(1 + \beta^2) \cdot \frac{precision \cdot recall}{(\beta^2 \cdot precision) + recall}$$

II. Analysis

Data Exploration

I will be using the Telco Customer Churn "Focused customer retention programs" Dataset [See here] from Kaggle. This was uploaded for examining customer retention and predicting churn and will be well suited to this study. The data contains more than 7043 row each row represents a customer and 21 features including the target label as a flag.

They contain both numeric and categorical features that represents different customer's attributes that may contribute to customer churn they are as follows:

- **customerID -** A unique identifier for each cutomer
- **Gender -** Whether the customer is a male or a female
- **SeniorCitizen -** Whether the customer is a senior citizen or not (1, 0)
- **Partner -** Whether the customer has a partner or not (Yes, No)
- **Dependents -** Whether the customer has dependents or not (Yes, No)
- **Tenure -** Number of months the customer has stayed with the company
- **PhoneService -** Whether the customer has a phone service or not (Yes, No)
- **MultipleLines -** Whether the customer has multiple lines or not (Yes, No, No phone service)
- **InternetService -** Customer's internet service provider (DSL, Fiber optic, No)
- **OnlineSecurity -** Whether the customer has online security or not (Yes, No, No internet service)
- **OnlineBackup -** Whether the customer has online backup or not (Yes, No, No internet service)
- **DeviceProtection -** Whether the customer has device protection or not (Yes, No, No internet service)
- **TechSupport -** Whether the customer has tech support or not (Yes, No, No internet service)
- **StreamingTV -** Whether the customer has streaming TV or not (Yes, No, No internet service)
- **StreamingMovies -** Whether the customer has streaming movies or not (Yes, No, No internet service)
- **Contract -** The contract term of the customer (Month-to-month, One year, Two year)
- **PaperlessBilling -** Whether the customer has paperless billing or not (Yes, No)
- **PaymentMethod -** The customer's payment method (Electronic check, Mailed check, Bank transfer (automatic), Credit card (automatic))

- **MonthlyCharges -** The amount charged to the customer monthly
- **TotalCharges -** The total amount charged to the customer
- **Churn -** Whether the customer churned or not (Yes or No)

An imbalance can be observed in the target label as the number of lost customers are 1890 rows while the number of non-churn customers are 5174 rows.

Data Profiling:

Overview

Dataset info

Number of variables	21
Number of observations	7043
Total Missing (%)	0.0%
Total size in memory	1.1 MiB
Average record size in memory	168.0 B

Variables types

Numeric	2
Categorical	17
Boolean	1
Date	0
Text (Unique)	1
Rejected	0
Unsupported	0

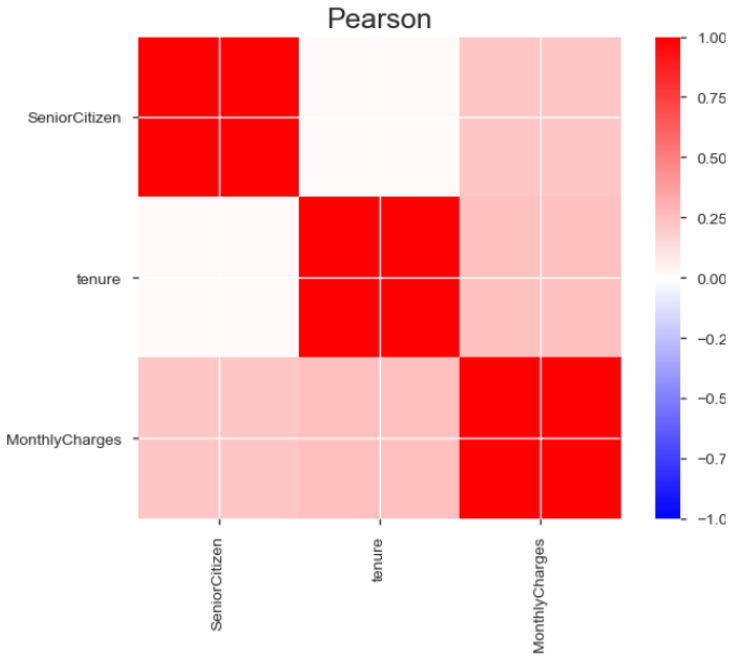
Warnings

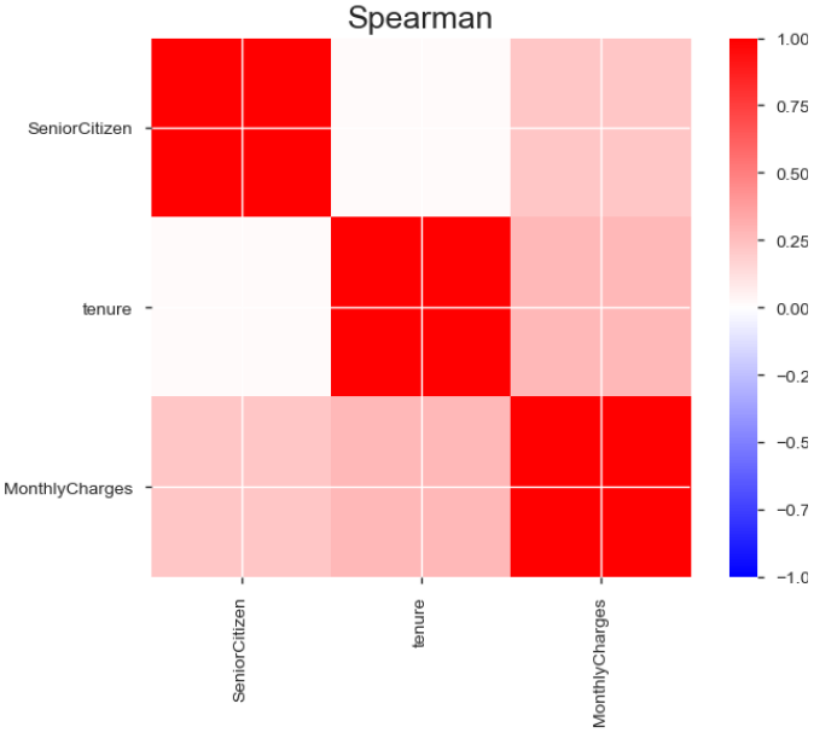
- `TotalCharges` has a high cardinality: 6531 distinct values Warning

Sample

	customerID	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetService
0	7590-VHVEG	Female	0	Yes	No	1	No	No phone service	DSL
1	5575-GNVDE	Male	0	No	No	34	Yes	No	DSL
2	3668-QPYBK	Male	0	No	No	2	Yes	No	DSL
3	7795-CFOCW	Male	0	No	No	45	No	No phone service	DSL
4	9237-HQITU	Female	0	No	No	2	Yes	No	Fiber optic

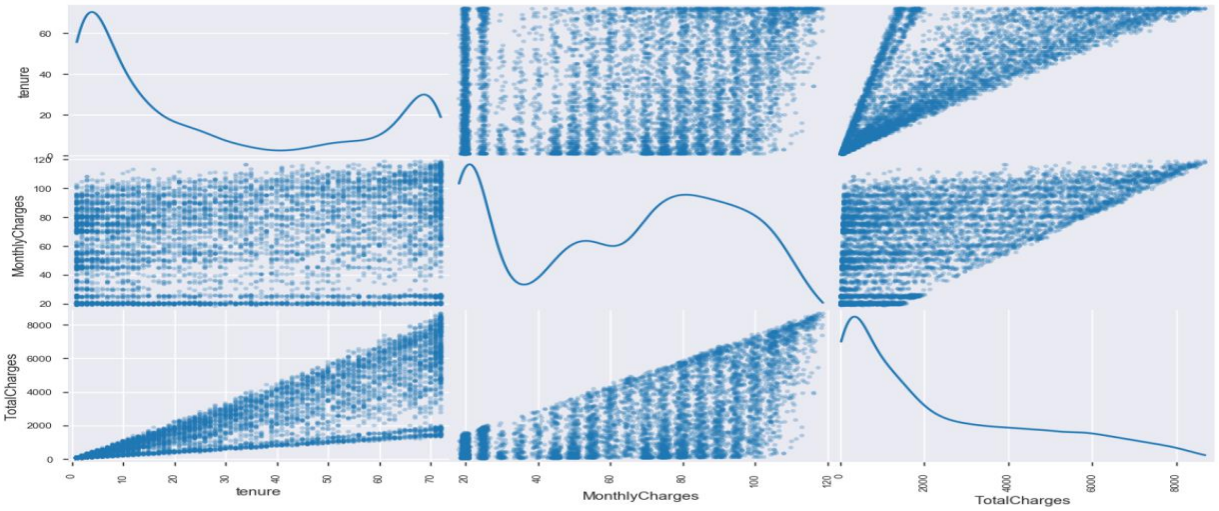
Correlations

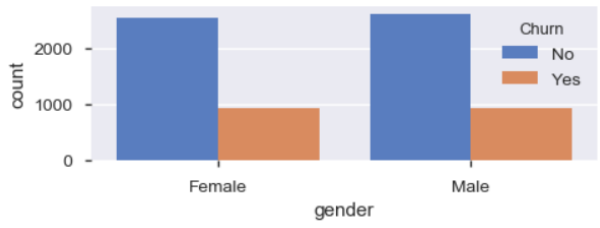


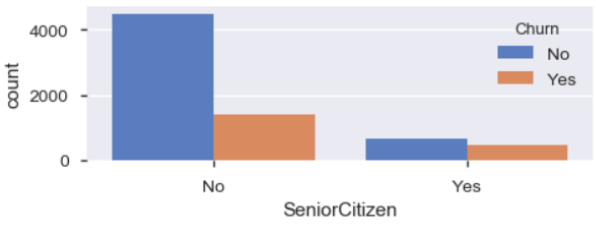


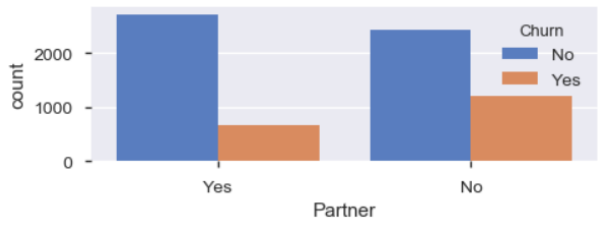
```
customerID                , Rows:  7043, NAs: 0                , precentage: 0.0
gender                    , Rows:  7043, NAs: 0                , precentage: 0.0
SeniorCitizen             , Rows:  7043, NAs: 0                , precentage: 0.0
Partner                   , Rows:  7043, NAs: 0                , precentage: 0.0
Dependents                , Rows:  7043, NAs: 0                , precentage: 0.0
tenure                    , Rows:  7043, NAs: 0                , precentage: 0.0
PhoneService              , Rows:  7043, NAs: 0                , precentage: 0.0
MultipleLines             , Rows:  7043, NAs: 0                , precentage: 0.0
InternetService           , Rows:  7043, NAs: 0                , precentage: 0.0
OnlineSecurity            , Rows:  7043, NAs: 0                , precentage: 0.0
OnlineBackup              , Rows:  7043, NAs: 0                , precentage: 0.0
DeviceProtection          , Rows:  7043, NAs: 0                , precentage: 0.0
TechSupport               , Rows:  7043, NAs: 0                , precentage: 0.0
StreamingTV               , Rows:  7043, NAs: 0                , precentage: 0.0
StreamingMovies           , Rows:  7043, NAs: 0                , precentage: 0.0
Contract                  , Rows:  7043, NAs: 0                , precentage: 0.0
PaperlessBilling          , Rows:  7043, NAs: 0                , precentage: 0.0
PaymentMethod             , Rows:  7043, NAs: 0                , precentage: 0.0
MonthlyCharges            , Rows:  7043, NAs: 0                , precentage: 0.0
TotalCharges              , Rows:  7043, NAs: 11               , precentage: 0.1561834445548772
Churn                     , Rows:  7043, NAs: 0                , precentage: 0.0
columns with NAs : ['TotalCharges']
```

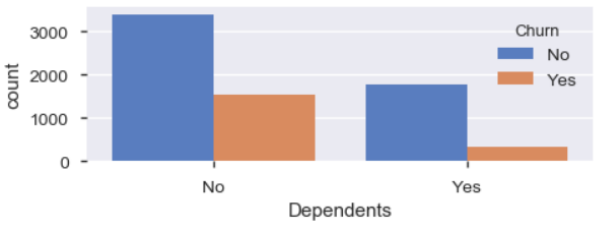
Exploratory Visualization

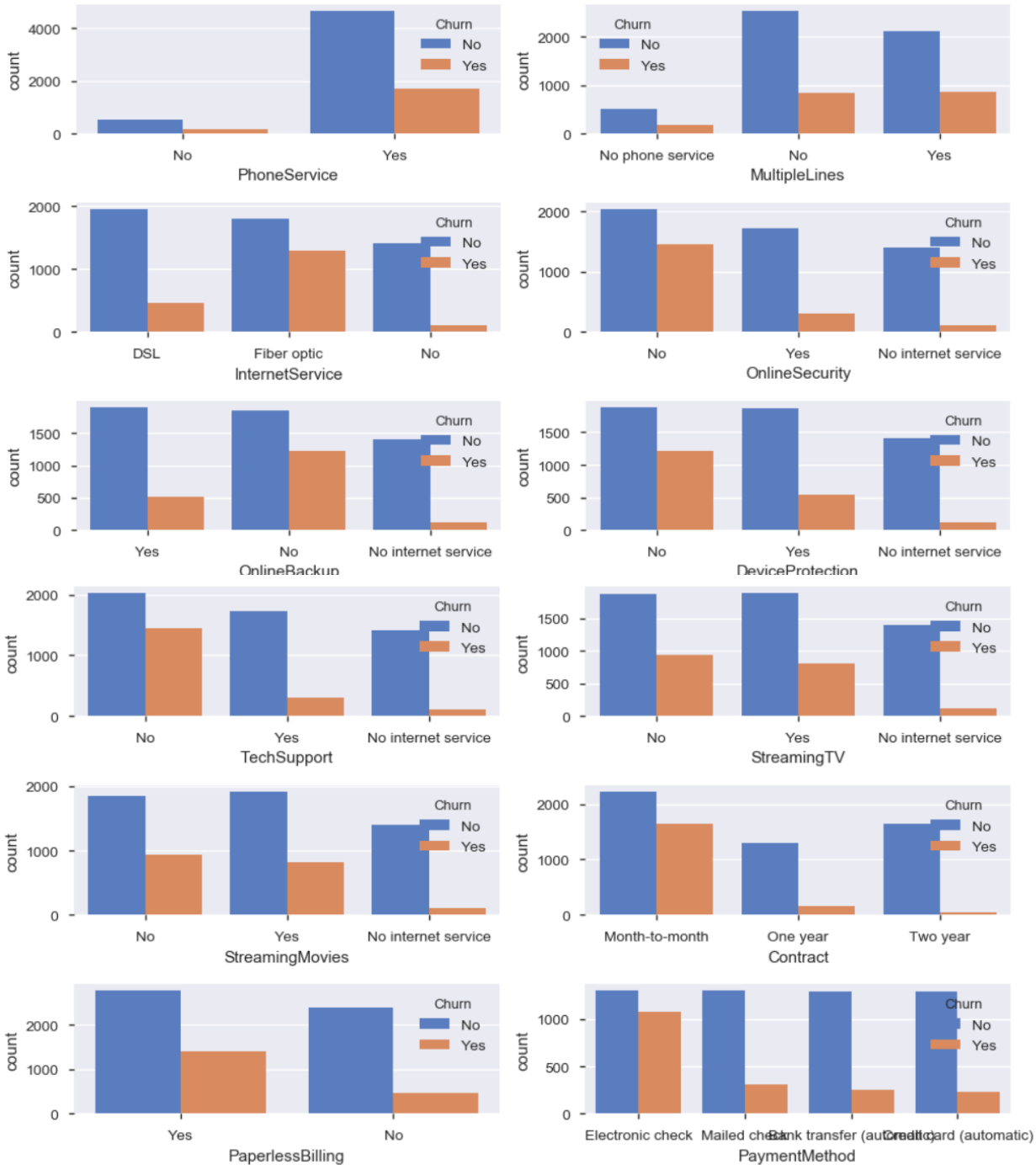











count
Churn
No
Yes
PhoneService
No
Yes
MultipleLines
No phone service
No
Yes
InternetService
DSL
Fiber optic
No
OnlineSecurity
No
Yes
No internet service
OnlineBackup
Yes
No
No internet service
DeviceProtection
No
Yes
No internet service
TechSupport
No
Yes
No internet service
StreamingTV
No
Yes
No internet service
StreamingMovies
No
Yes
No internet service
Contract
Month-to-month
One year
Two year
PaperlessBilling
Yes
No
PaymentMethod
Electronic check
Mailed check
Bank transfer (automatic)
Credit card (automatic)

Algorithms and Techniques

Unsupervised Algorithms:

- **XGBoost** is an optimized distributed gradient boosting library designed to be highly efficient, flexible and portable. It implements machine learning algorithms under the Gradient Boosting framework. XGBoost provides a parallel tree boosting (also known as GBDT, GBM) that solve many data science problems in a fast and accurate way. The same code runs on major distributed environment (Hadoop, SGE, MPI) and can solve problems beyond billions of examples [1].

- **k-means** clustering is a method of vector quantization, originally from signal processing, that is popular for cluster analysis in data mining. k-means clustering aims to partition n observations into k clusters in which each observation belongs to the cluster with the nearest mean, serving as a prototype of the cluster. This results in a partitioning of the data space into Voronoi cells. k-Means minimizes within-cluster variances (squared Euclidean distances) [2].

Supervised Algorithms:

- **Naïve Bayes** Naïve Bayes classifiers are highly scalable, requiring a number of parameters linear in the number of variables (features/predictors) in a learning problem. Maximum-likelihood training can be done by evaluating a closed-form expression, which takes linear time, rather than by expensive iterative approximation as used for many other types of classifiers [3].

- **Support Vector Machines** (SVMs), also known as support vector networks, are a family of extremely powerful models which use method-based learning and can be used in classification and regression problems. They aim at finding decision boundaries that separate observations with differing class memberships. In other words, SVM is a discriminative classifier formally defined by a separating hyperplane [4].

- **Random forest** is an ensemble approach that can also be thought of as a form of nearest neighbor predictor. Ensembles are a divide-and-conquer approach used to improve performance. The principle behind ensemble methods is that a group of "weak learners" can come together to form a "strong learner." Individually, each classifier is a "weak learner," while taken together they are a "strong learner" [5].

Benchmark

Looking on some kernels on kaggle it can be observed that accuracy ranges from 84% to 73% with kernels using relatively simple models as logistic regression (See here) an accuracy of 75% and sensitivity of 75%, other tree based models were used and they perform rather well.

III. Methodology

(approx. 3-5 pages)

Data Preprocessing

Data Cleansing:

- **Data Formatting,** Looking at the different variables I found two variables that needed to be converted to other types these variables are "TotalCharges" which needed to be converted to numerical values and "SeniorCitizen" which needed to be converted from binary to categorical.

- **Dropping Nulls,** Nulls needed to be detected, measured and handled in the dataset I found only a neglectable percentage of one variable to contain Nulls which then I proceeded to clean by dropping the rows containing these null values.

Feature Engineering:

- **Feature deriving,** I derived three new feature num_of_services, sustain_score and internet_speed from existing features I added the number of services subscribed by each user to create the numerical variable "num_of_services". I ranked the contracts types using the time for each contract as a bases { "Month-to-month": 1, "One year": 2, 'Two year': 3 } to get the ordinal variable "sustain_score". I ranked the internet speed based on the type of infrastructure provided to each user as follows "DSL":1,"Fiber optic": 2,"No":0} to get the ordinal variable "internet_speed".

Data Preparation:

- **One-hot Encoding,** As shown above, there are some categorical features, One-hot encoding transforms categorical features to a numerical format that works better with learning algorithms. It creates a dummy variable for each possible category of each non-numeric feature. For example, after one-hot encoding, the "gender" feature is encoded into "gender_male" and "gender_female", each being a Boolean column with values 0 or 1.

- **Normalization,** As shown above, there are some numerical features in the dataset taking values in various ranges. Feature scaling or normalization is a method used to standardize the range of numerical features. In order to make sure that each feature is treated equally when applying supervised learners, *StandardScaler* from the scikit-learn library.

Implementation

I loaded the needed libraries for preprocessing, cleansing and data profiling then I used pandas profiling to profile the data and did some simple data exploration using the following code:

```
#import the nececassary libraries
import pandas as pd
import numpy as np
import pandas_profiling as pp
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
```

```
# load tha data into pandas
df = pd.read_csv('.\\telco-customer-churn\\WA_Fn-UseC_-Telco-Customer-Churn.csv')
```

```
# take a look at the columns
df.columns
```

```
# take a look at the column types
df.info()
```

```
# Display a profile for our data
pp.ProfileReport(df)
```

Then I cleaned the dataset from nulls using a function that I created and converted some features to more appropriate types as shown in the following code:

```
# Convert TotalCharges to numeric and SeniorCitizen to categorical
df["TotalCharges"] = pd.to_numeric(df["TotalCharges"],errors='coerce')
df['SeniorCitizen'] = df.SeniorCitizen.map({0: 'No', 1: 'Yes'},na_action='ignore')
df.describe()
```

```
# create a fcuntion to find and display NAs
def count_NAs(dfna):
    NaList = []
    numOfRows = len(dfna)
    for i in dfna.columns:
        numOfNAs = dfna[i].isna().sum()
        print(f'{i:<30}, Rows: {numOfRows:>5}, NAs: {numOfNAs:<25}, precentage: {numOfNAs/numOfRows * 100.000}')
        if numOfNAs != 0:
           NaList.append(i)
    print(f'columns with NAs : {NaList}')
    return NaList
```

```
# Display NAs
NAs = count_NAs(df)
```

```
# Drop rows containing nulls
df = df.dropna()
```

Then comes the visuals I visualized all categorical variables against the target and all numerical variables against each other

Visuals

```
# Display all categoral columns against target variable
col_to_plot = ['gender','SeniorCitizen','Partner', 'Dependents','PhoneService', 'MultipleLines', 'InternetService',
               'OnlineSecurity', 'OnlineBackup', 'DeviceProtection', 'TechSupport',
               'StreamingTV', 'StreamingMovies', 'Contract','PaperlessBilling',
               'PaymentMethod']
f, axes = plt.subplots(round(len(col_to_plot)/2),2, figsize=(10, 15))
for i,x in zip(col_to_plot,axes.flat):
    sns.countplot(data = df ,x = str(i) ,ax = x ,hue = 'Churn'  ,palette="muted")
f.show()
plt.tight_layout()
```

```
# Display numerical columns asgainst each other
pd.plotting.scatter_matrix(df, alpha = 0.3, figsize = (14,8), diagonal = 'kde')
```

Then I prepared the data for clustering and I defined a function to work with both kmeans and GaussianMixture the following are the results of both clusters methods

```
clustering using GaussianMixture
Number of clusters: 2 with score: 0.23566957428368457
Number of clusters: 3 with score: 0.2363869278819532
Number of clusters: 4 with score: 0.2453773458379687
Number of clusters: 5 with score: 0.23034297571791112
Number of clusters: 6 with score: 0.19610343525687268
Number of clusters: 7 with score: 0.20408185708010623
Number of clusters: 8 with score: 0.17178651823094002
Number of clusters: 9 with score: 0.22669738799818237
Number of clusters: 10 with score: 0.2100273224544588
Number of clusters: 11 with score: 0.21687879265784893
Number of clusters: 12 with score: 0.2104569573299668
Number of clusters: 13 with score: 0.22532787869258641
Number of clusters: 14 with score: 0.2165536648388327
cluster with Max score: 14 with score: 0.2453773458379687
Cluster Centers
```

:

	SeniorCitizen	Partner	Dependents	tenure	MonthlyCharges	TotalCharges	num_of_services	sustain_score	data_speed
0	0.0	0.0	0.0	31.0	21.0	666.0	1.0	2.0	0.0
1	0.0	0.0	0.0	10.0	65.0	613.0	4.0	1.0	1.0
2	1.0	1.0	0.0	33.0	83.0	2900.0	5.0	1.0	2.0
3	0.0	1.0	0.0	52.0	83.0	4351.0	6.0	2.0	2.0

```
clustering using Kmeans
Number of clusters: 2 with score: 0.26629442394203806
Number of clusters: 3 with score: 0.2689058944442241
Number of clusters: 4 with score: 0.2815320288943176
Number of clusters: 5 with score: 0.2613547221332882
Number of clusters: 6 with score: 0.27306503032952056
Number of clusters: 7 with score: 0.2667844916233723
Number of clusters: 8 with score: 0.28149014274676804
Number of clusters: 9 with score: 0.2891574257061582
Number of clusters: 10 with score: 0.29367095176929725
Number of clusters: 11 with score: 0.27850643244439294
Number of clusters: 12 with score: 0.2982723317584113
Number of clusters: 13 with score: 0.29417412774865626
Number of clusters: 14 with score: 0.30637190065299946
cluster with Max score: 14 with score: 0.30637190065299946
Cluster Centers
```

	SeniorCitizen	Partner	Dependents	tenure	MonthlyCharges	TotalCharges	num_of_services	sustain_score	data_speed
0	0.0	1.0	0.0	21.0	37.0	826.0	2.0	1.0	1.0
1	0.0	1.0	0.0	24.0	83.0	1997.0	5.0	1.0	2.0
2	1.0	1.0	0.0	57.0	95.0	5431.0	7.0	2.0	2.0
3	0.0	0.0	0.0	54.0	90.0	4811.0	7.0	2.0	2.0
4	0.0	1.0	1.0	63.0	88.0	5558.0	7.0	3.0	1.0

- And though kmeans have a higher score I would still go with GMM since it has fewer segments which is more easier for the business to understand also GMM is a lot more flexible in terms of cluster covariance
- k-means is actually a special case of GMM in which each cluster's covariance along all dimensions approaches 0. This implies that a point will get assigned only to the cluster closest to it. With GMM, each cluster can have unconstrained covariance structure. Think of rotated and/or elongated distribution of points in a cluster, instead of spherical as in kmeans. As a result, cluster assignment is much more flexible in GMM than in k-mean

Then I prepare the data for modeling by converting the categorical variables to dummy variables and normalizing numerical values and split the data for training and testing using the following code:

preprare data for the model

```
# Normalize numerical columns
numerics = list(df.select_dtypes(include=[np.number]))
scaled_df = pd.DataFrame(data = df)
scaled_df[numerics] = StandardScaler(with_mean=True, with_std=True).fit_transform(df[numerics])

# Seperate target column from the rest of the dataframe and apply binary transformation
churn = scaled_df['Churn'].apply(lambda x: 1 if x == 'Yes' else 0)
scaled_df = scaled_df.drop(['Churn','customerID'],axis=1)

# Create dummy variables for categorical features
features_final = pd.get_dummies(scaled_df)

# Count number of features in the input dataframe
len(features_final.columns)
```

```
# Split the 'features' and 'churn' data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(features_final,
                                                    churn,
                                                    test_size = 0.2,
                                                    random_state = 0)

# Show the results of the split
print("Training set has {} samples.".format(X_train.shape[0]))
print("Testing set has {} samples.".format(X_test.shape[0]))
```

At last I built a dummy model as a baseline

```
# Establish a baseline line naive model
dummy = DummyClassifier().fit(X_train, y_train)
dummy_pred = dummy.predict(X_test)

# checking unique labels
print('Unique predicted labels: ', (np.unique(dummy_pred)))

# checking accuracy
print('Test score: ', accuracy_score(y_test, dummy_pred))

print('Test fscore: ',fbeta_score(y_test, dummy_pred, 2))
```

Then I built, tuned and compared my models to chose which one to take into consideration

Refinement

GridSearchCV is used to fine tune the chosen model. By specifying a parameter grid with hyperparameter values to experiment with, we can use GridSearchCV to evaluate all possible parameter combinations using cross validation. The best combination of parameters which give the best evaluation score is determined.

In the case of Random Forest, one hyperparameter to tune is 'n_estimators', which is the number of trees to build before taking the maximum majority or averages of predictions.

Higher number of trees gives better performance but makes the model slower. Possible values of 10, 20, 40, 50, 75, 100, 120, 140 are specified. The other hyperparameter is 'min_samples_split', which is the minimum number of samples required to split an internal node. A smaller number makes the model more prone to capturing noise in train data. Possible values of 2, 5, 6, 7 and 10 are passed to the parameter grid.

I ran grid search on all my models to optimize them as much as possible

IV. Results

Model Evaluation and Validation

Random Forest

```
Unoptimized model
------
Accuracy score on testing data: 0.7711
F-score on testing data: 0.5075

Optimized Model
------
Final accuracy score on the testing data: 0.7768
Final F-score on the testing data: 0.5849
```

Support Vector Machine

```
Unoptimized model
------
Accuracy score on testing data: 0.7555
F-score on testing data: 0.7033

Optimized Model
------
Final accuracy score on the testing data: 0.7050
Final F-score on the testing data: 0.7302
```

XGboost

```
Unoptimized model
------
Accuracy score on testing data: 0.7882
F-score on testing data: 0.6658

Optimized Model
------
Final accuracy score on the testing data: 0.7683
Final F-score on the testing data: 0.7185
```

I used F2 score since I wanted my main metrics to be fscore and recall so I weighted recall more than precision in the f score metric

And I believe XGboost is the best model for our problem since he has the highest f score and since XGboost is efficient and we can use it to derive feature importance which is really valuable for the business.

Justification

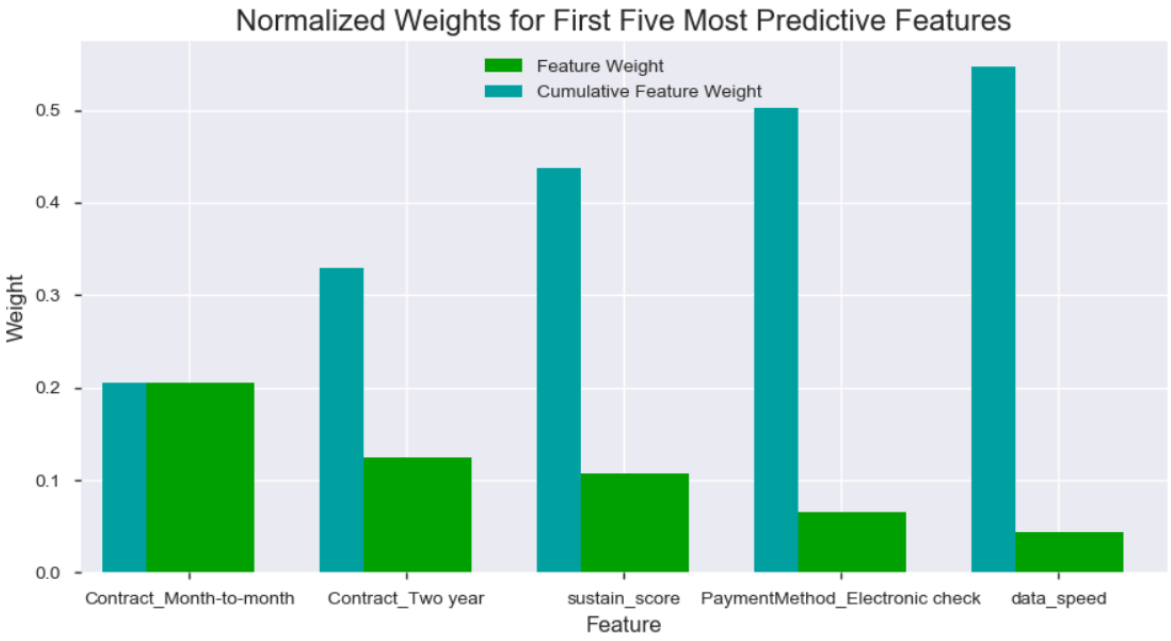
The following is the output for the dummy baseline model:

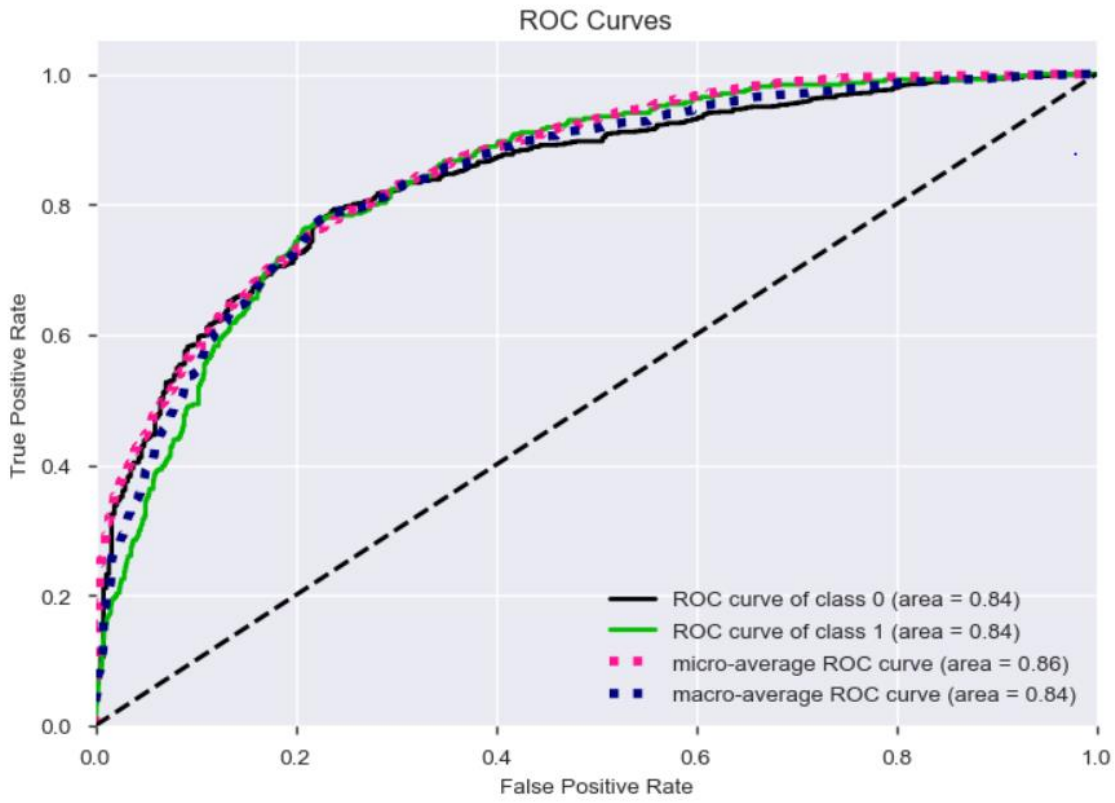
```
Unique predicted labels:  [0 1]
Test score:  0.5074626865671642
Test fscore:  0.4224376731301939
```

It can be observed that the xgboost is rather good compared with this output and that it produces a much better results

V. Conclusion

Free-Form Visualization





Reflection

To summarize, this project outlines a machine learning approach to help implement an effective customer churn prediction mode. A dataset containing both numerical and categorical features of the customers, as well as a binary label of whether the customer left or not is used to build a binary classification model.

The categorical features are transformed to numerical values by one-hot encoding. The numerical features are normalized so that the learners can treat each feature equally and features were derived from existing features.

Clustering was used to segment the customers to be able to identify higher tier customers and give them more importance for the business to take action to retain them.

Upsampling was used to solve the imbalance in the data and different classification algorithms including random forest, xgboost and SVM was used, tuned and compared and measured using the F2 score in order to put more weights on recall.

Tuning was done via GridSearchCV and the final model shows excellent F2 score much better than the dummy benchmark classifier) and appears to be generalizable to unseen data. With this classification model

Feature importance analysis reveals the top 5 features, which consist about 95% of the total importance.

Improvement

I do believe that more tuning can be used to achieve better results I could try random search with more ranges for models I also believe that a more in depth analysis of the different segments can yield more insights into customer behavior also if a time series column for customer history is provided I think it would bring even more insights in how to prevent customers from churning and it was really challenging to use xgboost as I had to research for a long time to be able to install, use and tun the model.