part4-churn

December 13, 2019

1 Part 4: Churn Prediction

Every company puts its efforts into knowing who their best customer are and then it also work hard on retaining them. That's what makes **Retention Rate** is one of the most critical metrics.

Retention Rate is an indication of how good is your product market fit (PMF). If your PMF is not satisfactory, you should see your customers churning very soon. One of the powerful tools to improve Retention Rate (hence the PMF) is Churn Prediction. By using this technique, you can easily find out who is likely to churn in the given period.

In this notebook, we will use a Telco dataset and go over following steps to develop churn prediction: * Exploratory data analysis * Feature engineering * Investigating how the features affect Retention by using Logistic Regression * Building a classification model with XGBoost

1.1 Exploratory Data Analysis

We start with checking out how our data looks like and visualize how it interacts with our label (churned or not?).

```
   customerID  gender  SeniorCitizen Partner Dependents  tenure PhoneService  \
0  7590-VHVEG  Female              0     Yes         No       1           No
1  5575-GNVDE    Male              0      No         No      34          Yes
2  3668-QPYBK    Male              0      No         No       2          Yes
3  7795-CFOCW    Male              0      No         No      45           No
4  9237-HQITU  Female              0      No         No       2          Yes

      MultipleLines InternetService OnlineSecurity  … DeviceProtection  \
0  No phone service             DSL             No  …               No
1                No             DSL            Yes  …              Yes
2                No             DSL            Yes  …               No
3  No phone service             DSL            Yes  …              Yes
4                No     Fiber optic             No  …               No

  TechSupport StreamingTV StreamingMovies        Contract PaperlessBilling  \
0          No          No              No  Month-to-month              Yes
1          No          No              No        One year               No
2          No          No              No  Month-to-month              Yes
3         Yes          No              No        One year               No
4          No          No              No  Month-to-month              Yes
```

```
             PaymentMethod MonthlyCharges  TotalCharges Churn
0           Electronic check          29.85         29.85    No
1               Mailed check          56.95        1889.5    No
2               Mailed check          53.85        108.15   Yes
3  Bank transfer (automatic)          42.30       1840.75    No
4           Electronic check          70.70        151.65   Yes

[5 rows x 21 columns]
```

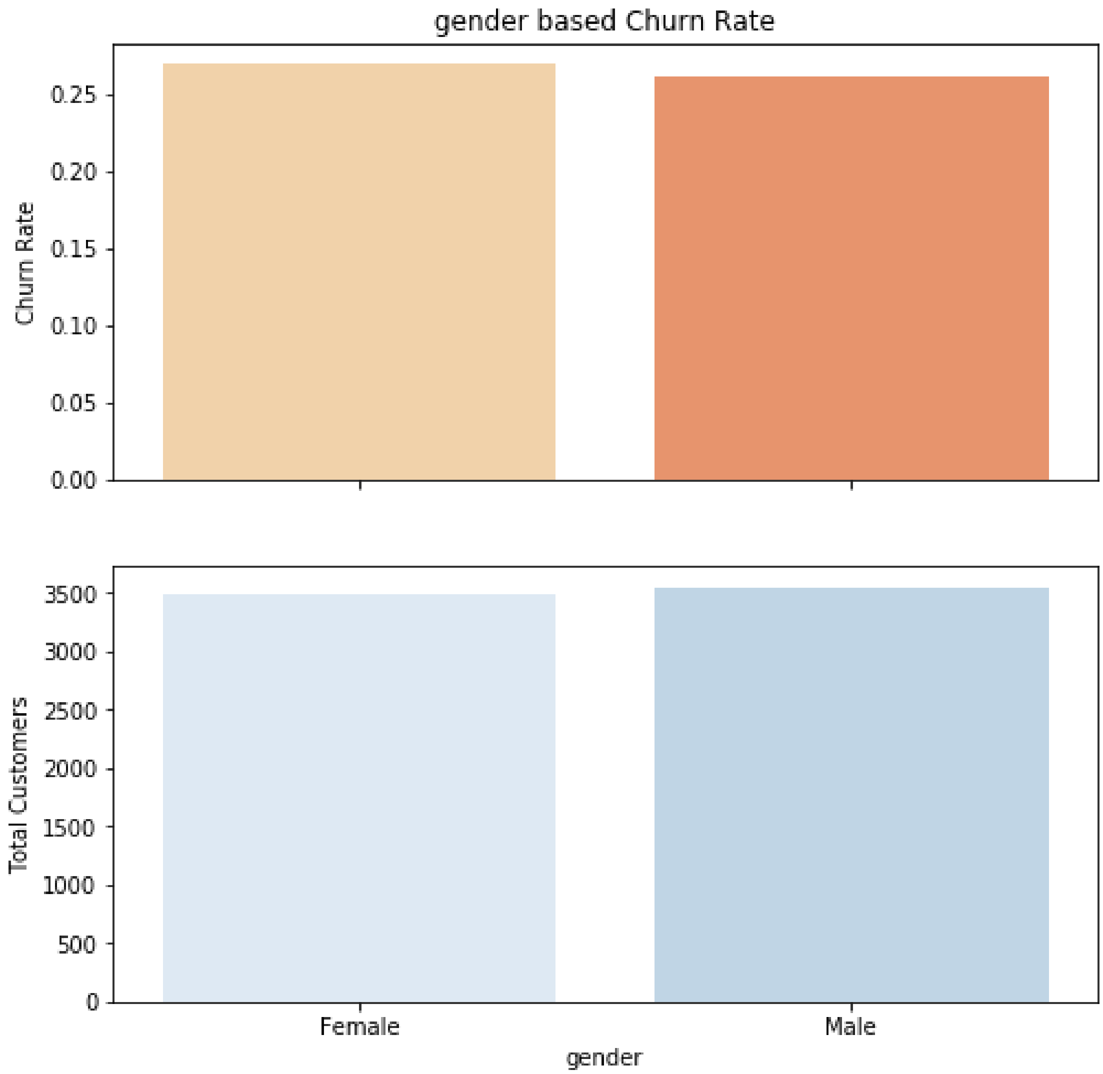
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7043 entries, 0 to 7042
Data columns (total 21 columns):
customerID          7043 non-null object
gender              7043 non-null object
SeniorCitizen       7043 non-null int64
Partner             7043 non-null object
Dependents          7043 non-null object
tenure              7043 non-null int64
PhoneService        7043 non-null object
MultipleLines       7043 non-null object
InternetService     7043 non-null object
OnlineSecurity      7043 non-null object
OnlineBackup        7043 non-null object
DeviceProtection    7043 non-null object
TechSupport         7043 non-null object
StreamingTV         7043 non-null object
StreamingMovies     7043 non-null object
Contract            7043 non-null object
PaperlessBilling    7043 non-null object
PaymentMethod       7043 non-null object
MonthlyCharges      7043 non-null float64
TotalCharges        7043 non-null object
Churn               7043 non-null object
dtypes: float64(1), int64(2), object(18)
memory usage: 1.1+ MB
```

Our data fall under two categories: * Categorical features: gender, streaming tv, payment method &, etc. * Numerical features: tenure, monthly charges, total charges

Now starting from the categorical ones, we shed light on all features and see how helpful they are to identify if a customer is going to churn.

Gender Let's start with how Churn rate looks with respect to Gender:

```
   gender      mean
0  Female  0.269209
1    Male  0.261603
```

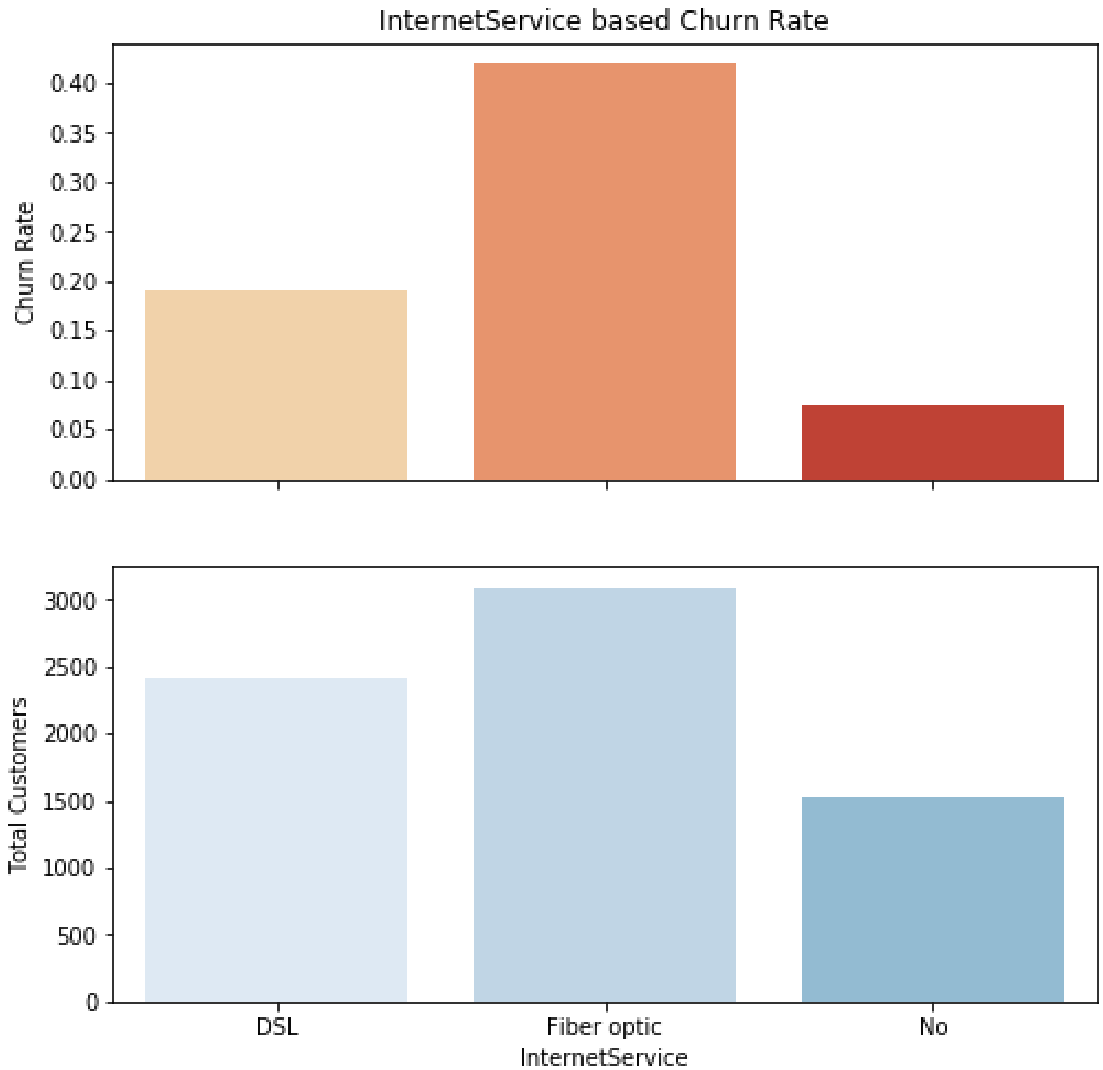


Female customers are more likely to churn vs. male customers, but the difference is minimal (~0.8%).

Let's replicate this for all categorical columns.

InternetService

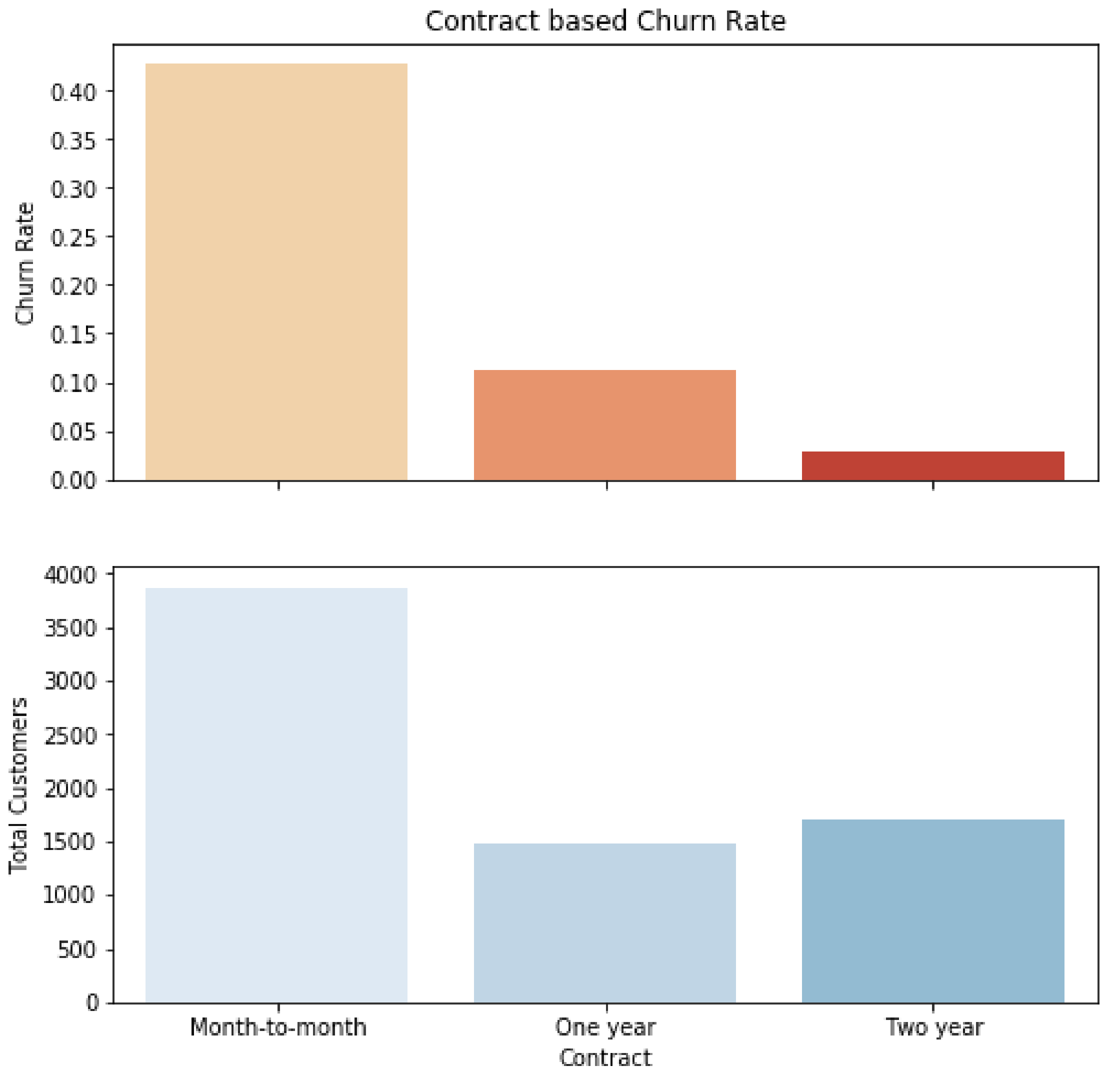
```
  InternetService      mean
0             DSL  0.189591
1     Fiber optic  0.418928
2              No  0.074050
```



This chart reveals customers who have Fiber optic as Internet Service are more likely to churn. I normally expect Fiber optic customers to churn less due to they use a more premium service. But this can happen due to high prices, competition, customer service, and many other reasons.

Contract

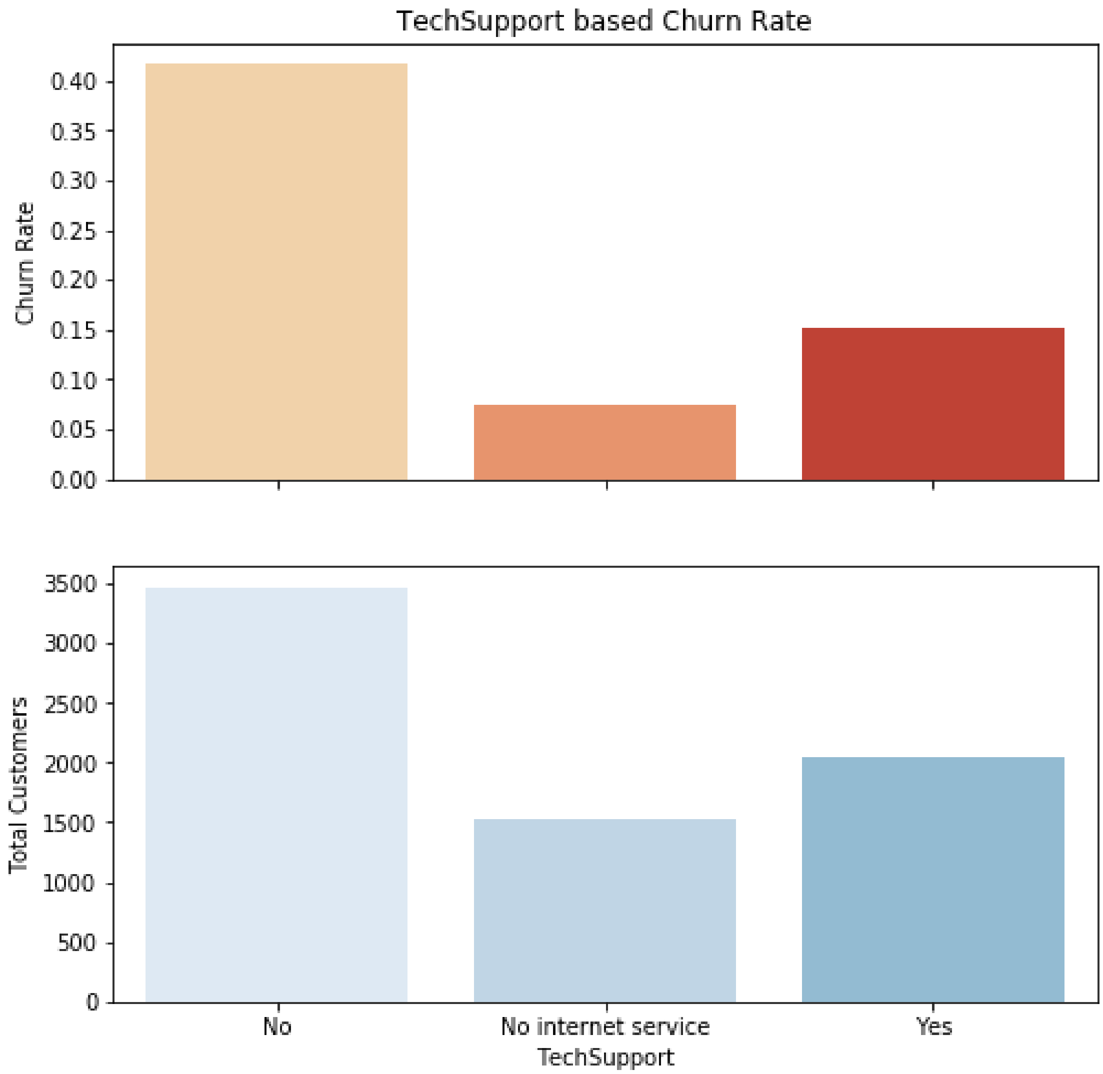
	Contract	mean
0	Month-to-month	0.427097
1	One year	0.112695
2	Two year	0.028319



As expected, the shorter contract means higher churn rate.

Tech Support

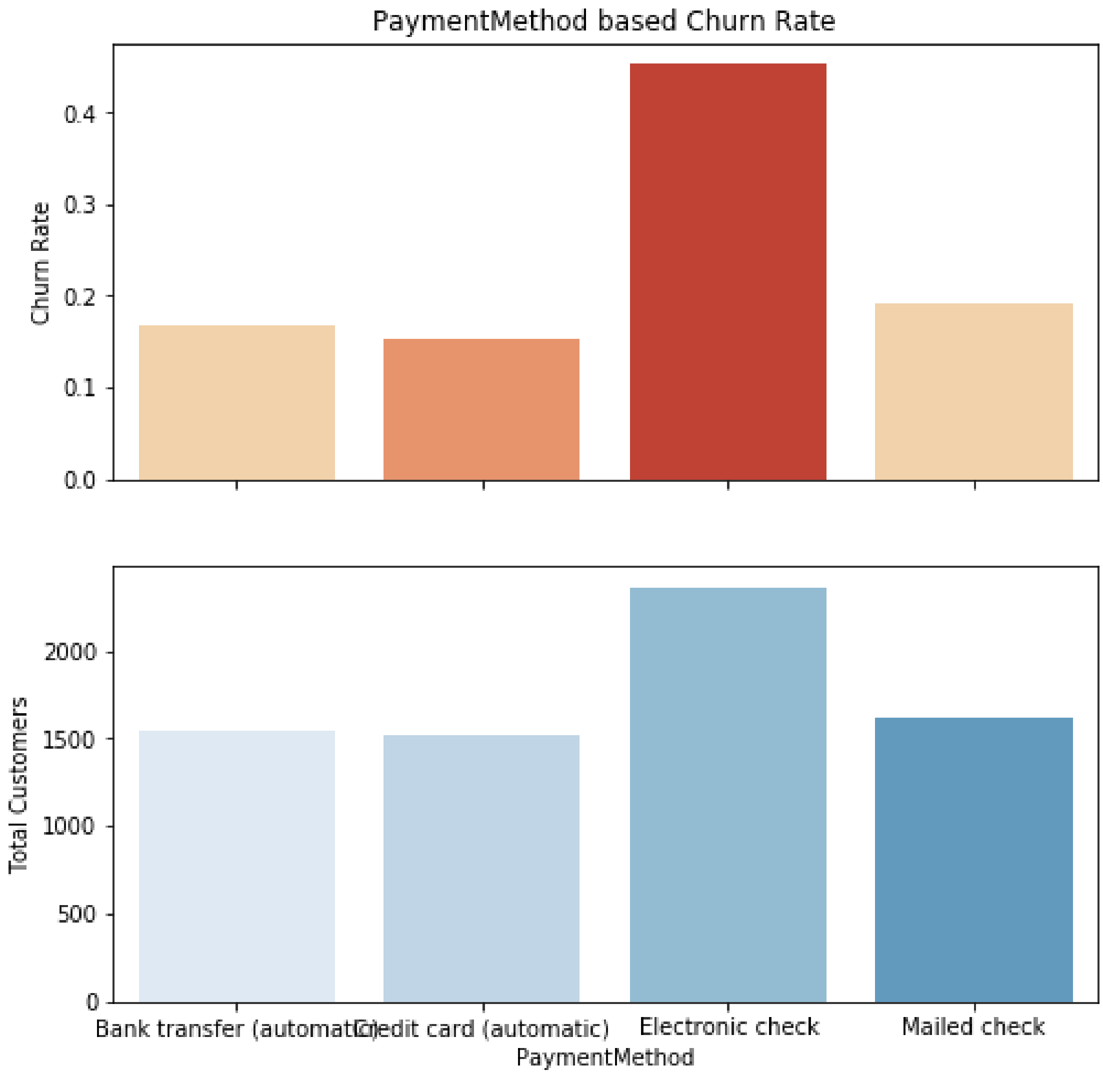
```
         TechSupport      mean
0                 No  0.416355
1  No internet service  0.074050
2                Yes  0.151663
```



Customers don't use Tech Support are more like to churn (~25% difference).

Payment Method

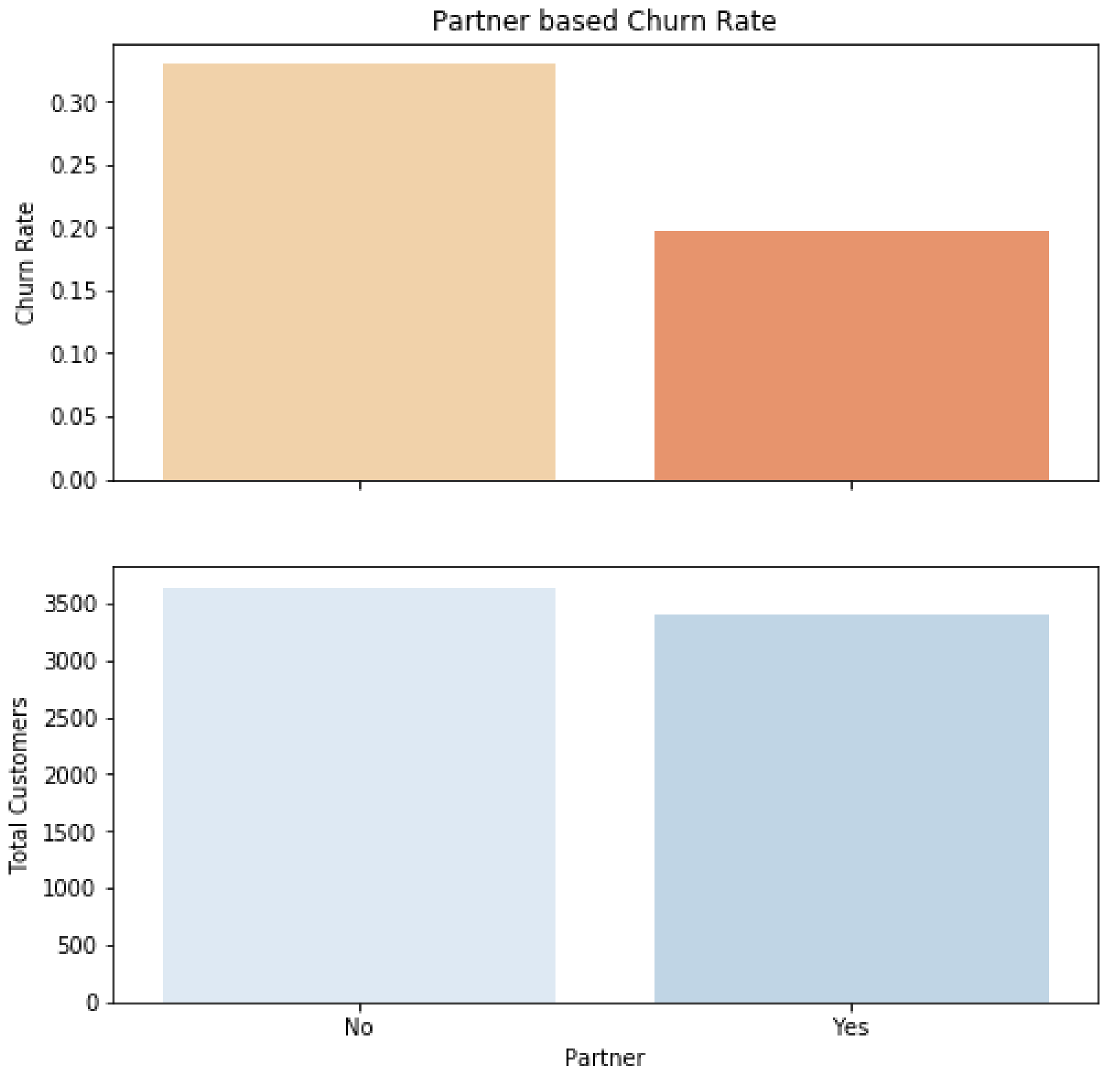
	PaymentMethod	mean
0	Bank transfer (automatic)	0.167098
1	Credit card (automatic)	0.152431
2	Electronic check	0.452854
3	Mailed check	0.191067



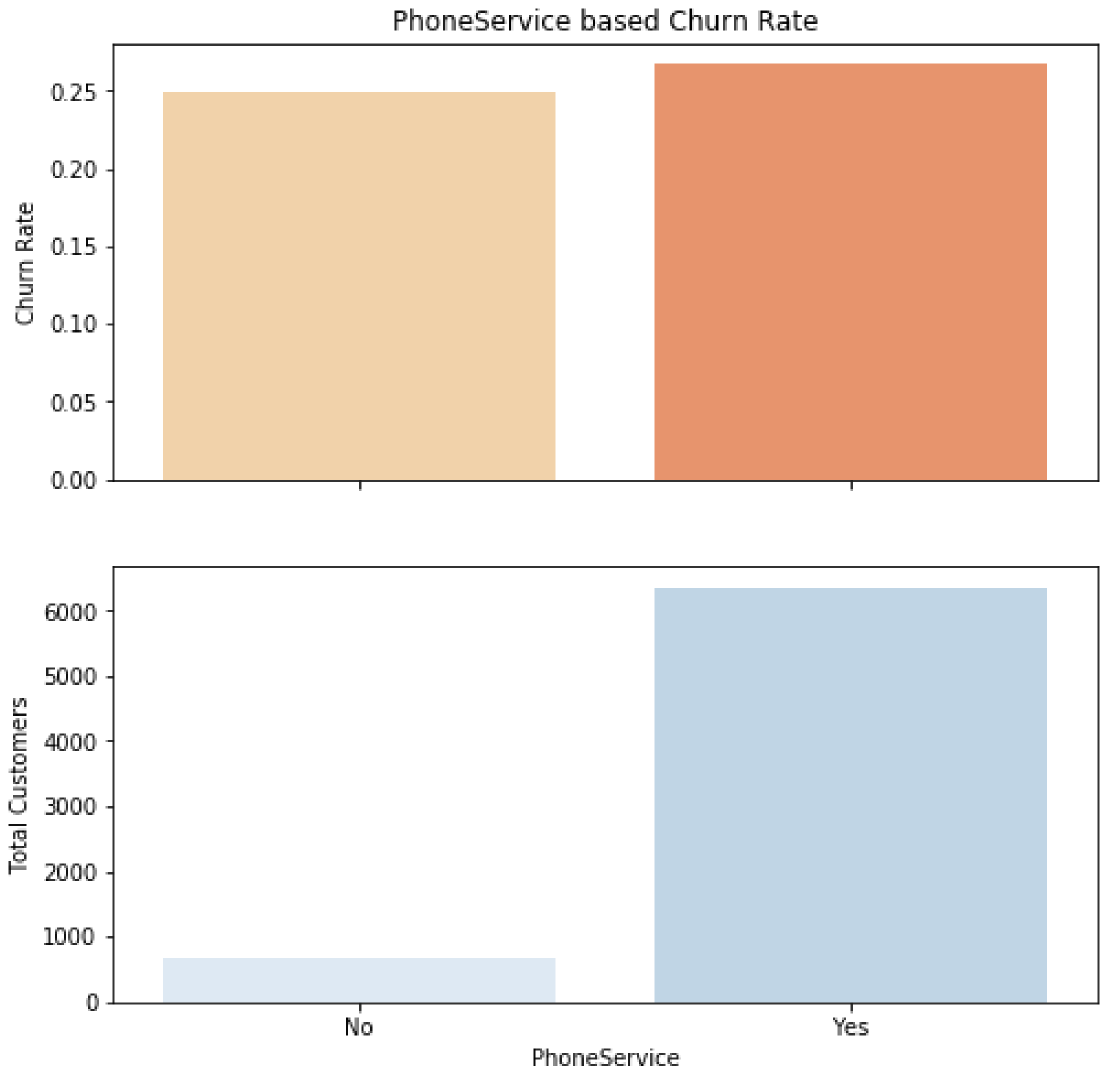
Automating the payment makes the customer more likely to retain in your platform (~30% difference).

Others

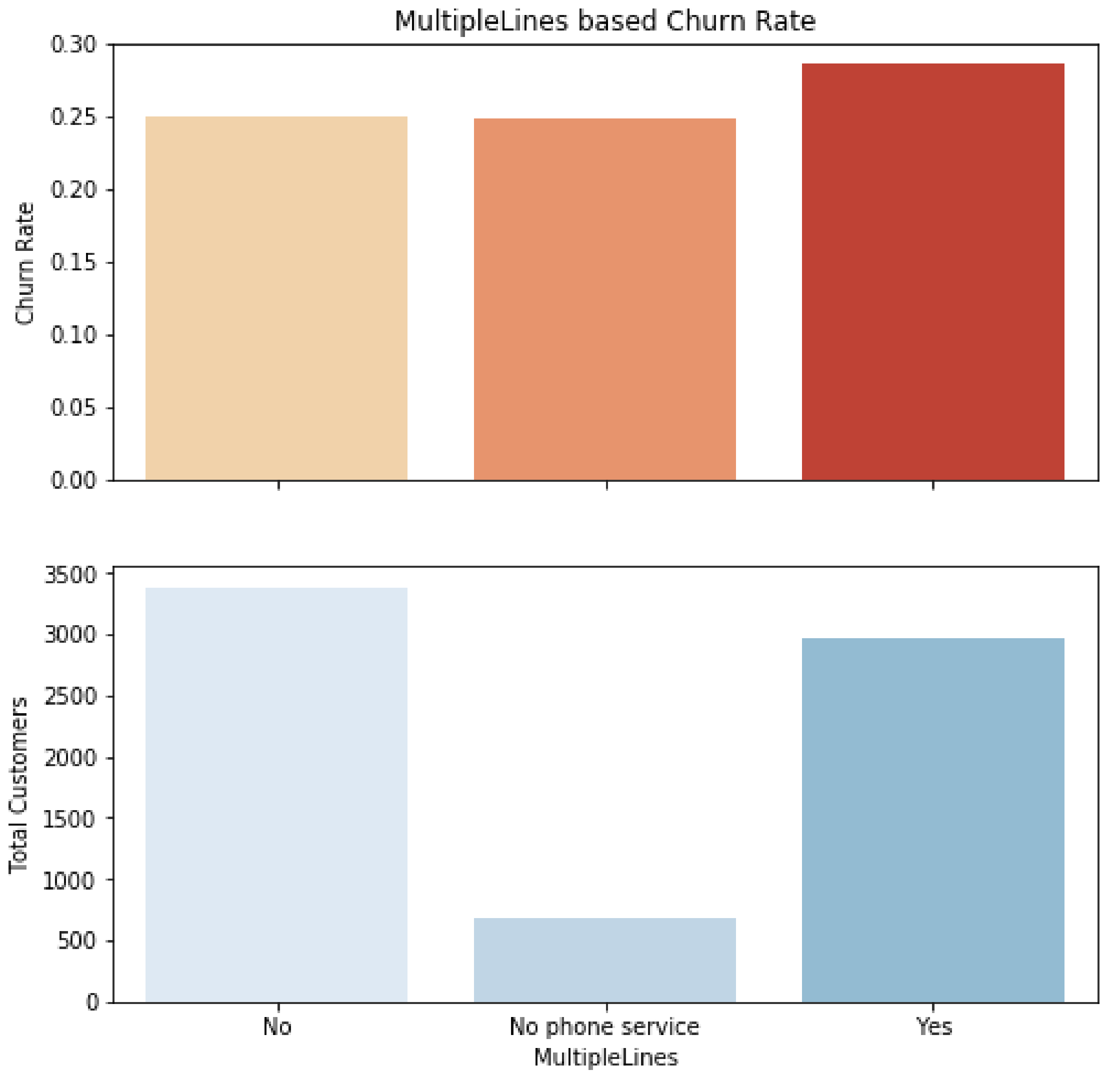
```
  Partner      mean
0      No  0.329580
1     Yes  0.196649
```



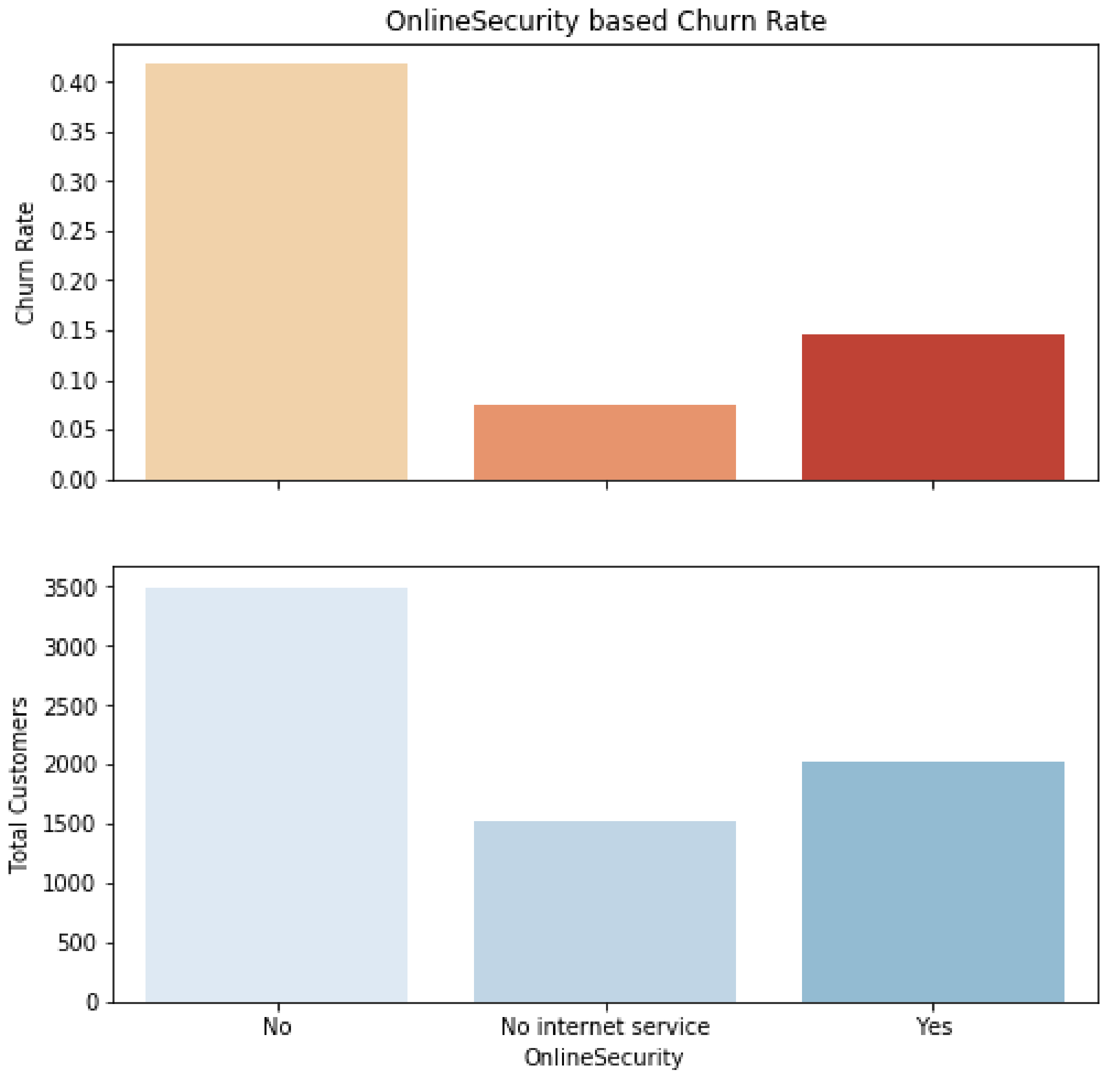
	PhoneService	mean
0	No	0.249267
1	Yes	0.267096



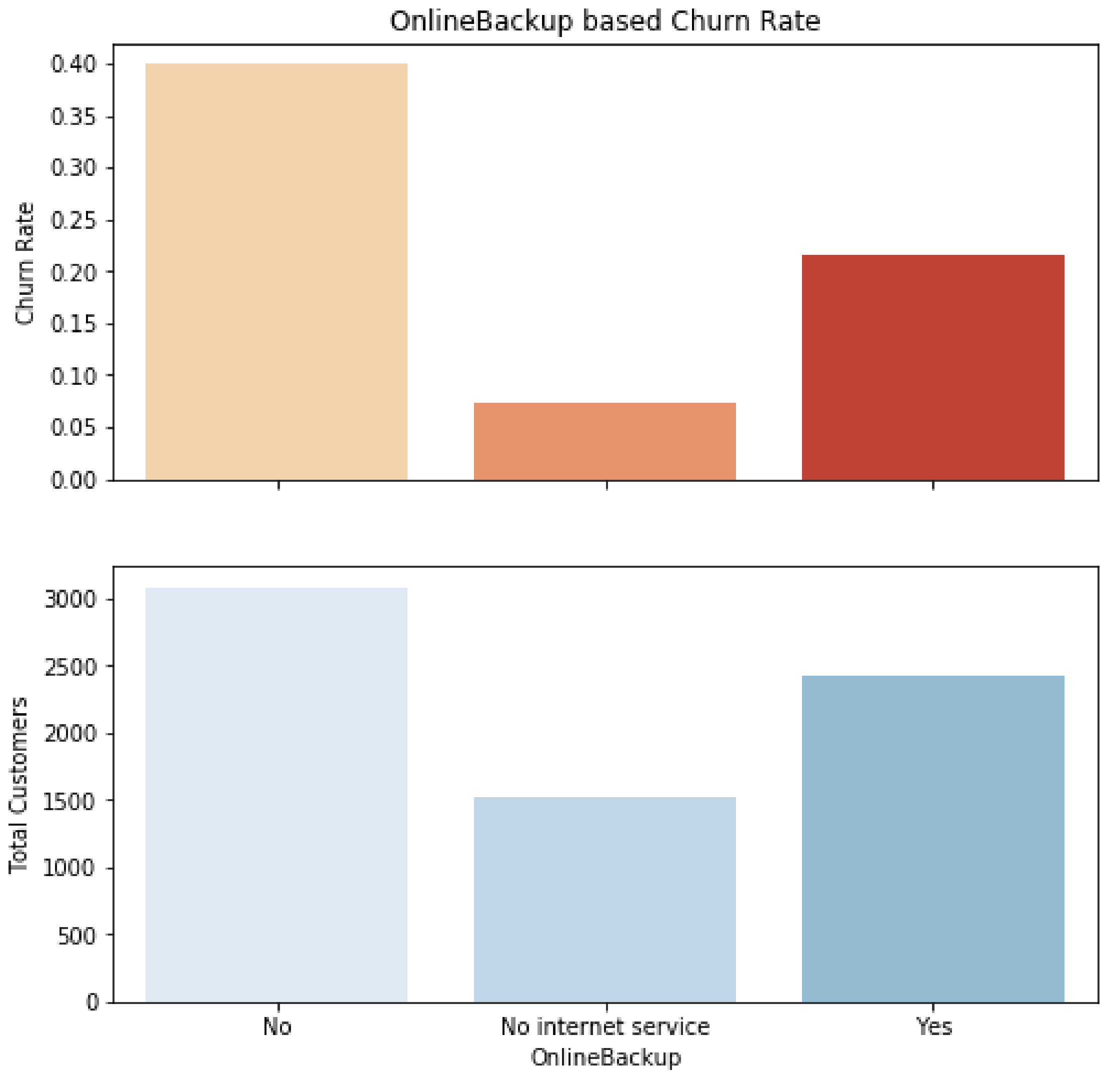

```
      MultipleLines      mean
0                No  0.250442
1  No phone service  0.249267
2               Yes  0.286099
```



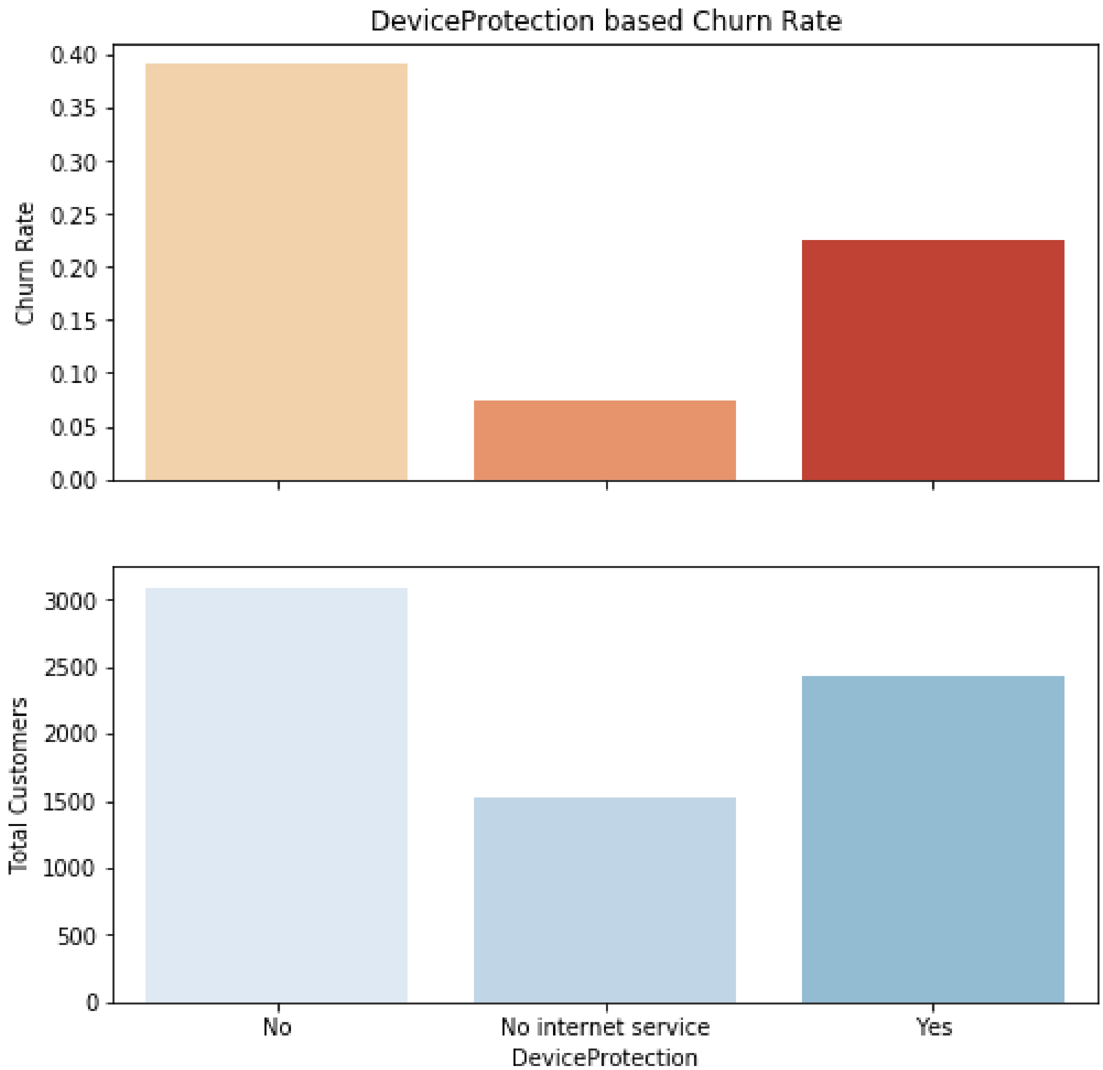

```
        OnlineSecurity      mean
0                   No  0.417667
1  No internet service  0.074050
2                  Yes  0.146112
```



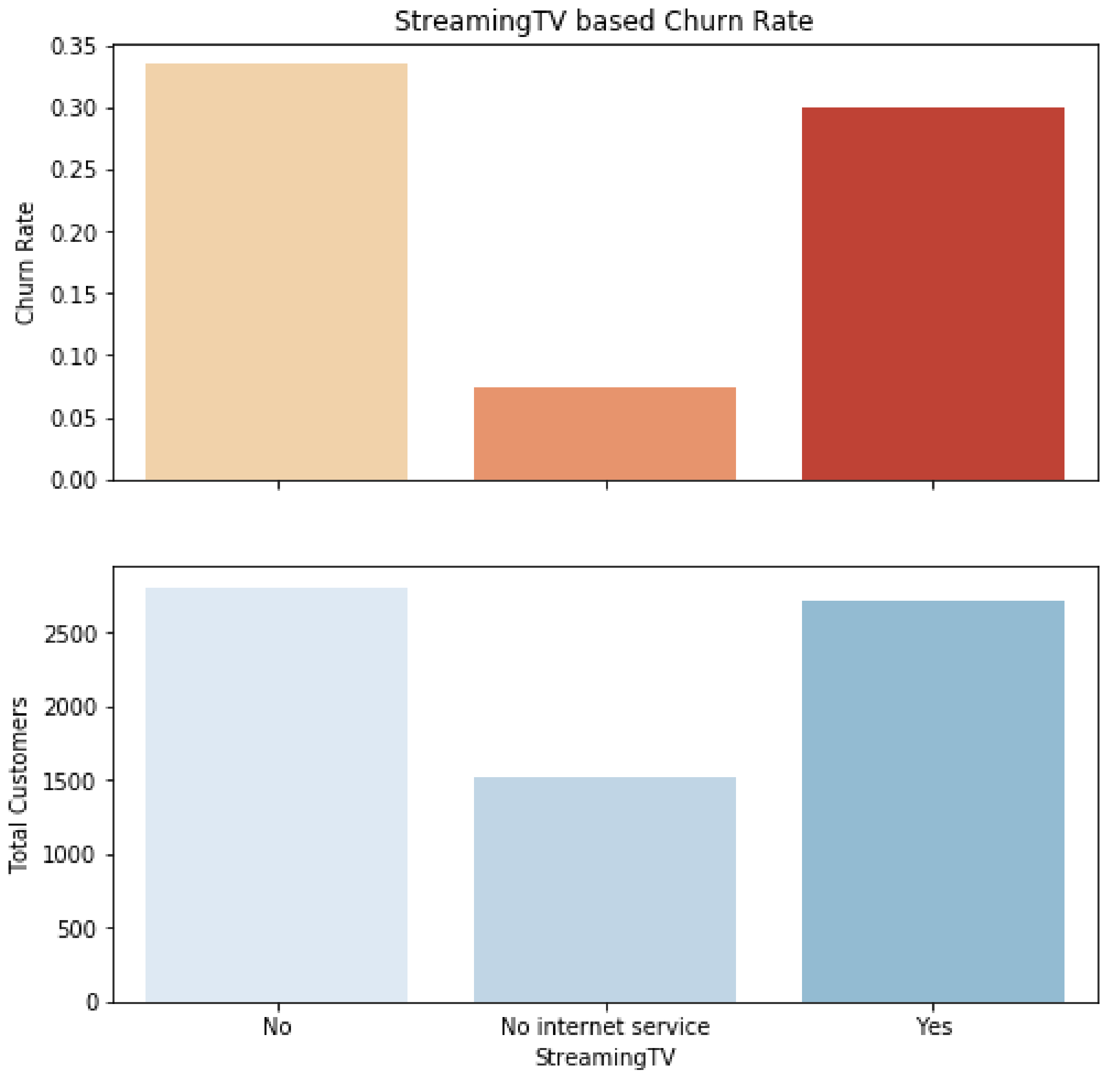

```
          OnlineBackup      mean
0                   No  0.399288
1  No internet service  0.074050
2                  Yes  0.215315
```



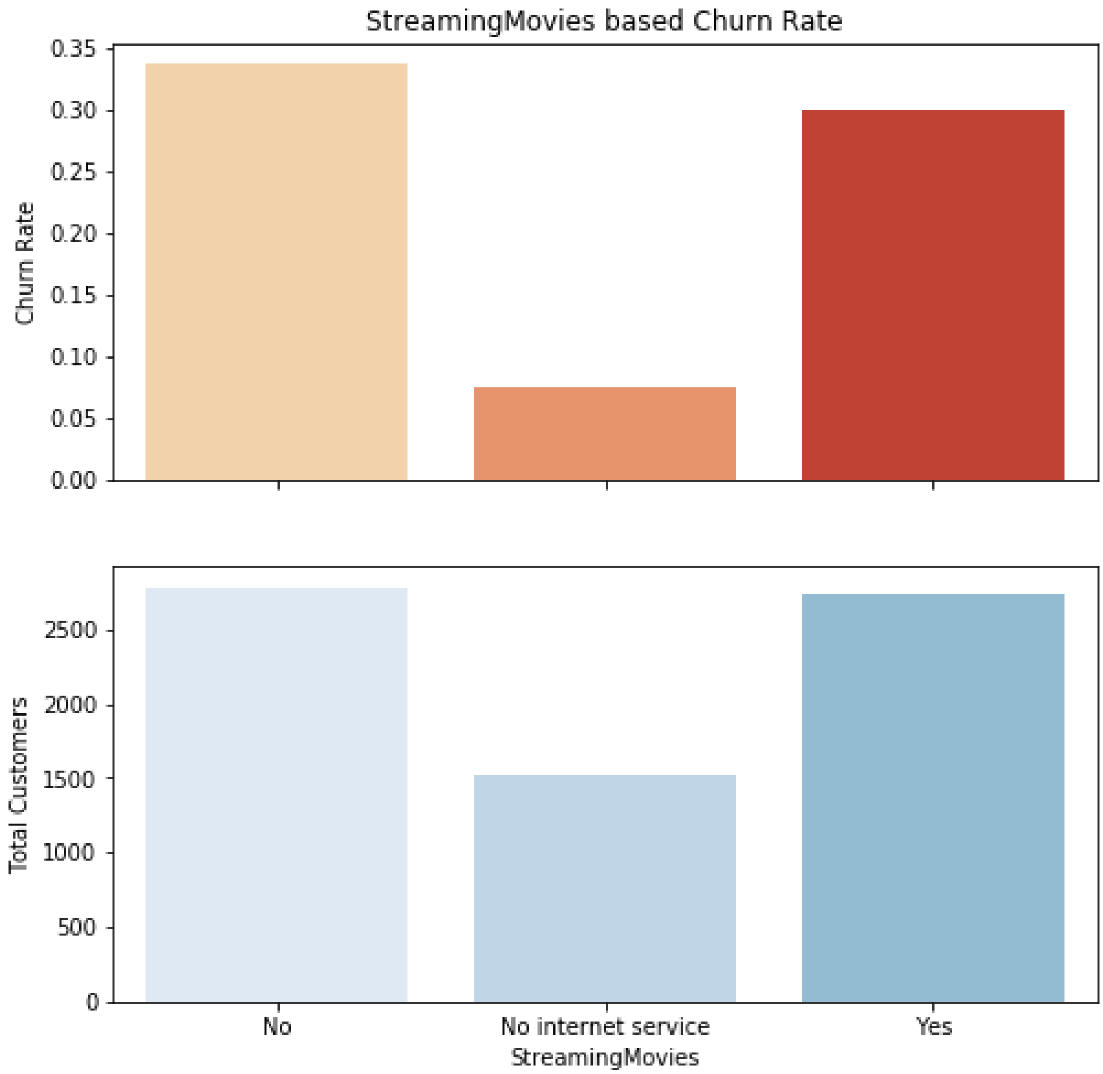

```
       DeviceProtection      mean
0                    No  0.391276
1   No internet service  0.074050
2                   Yes  0.225021
```



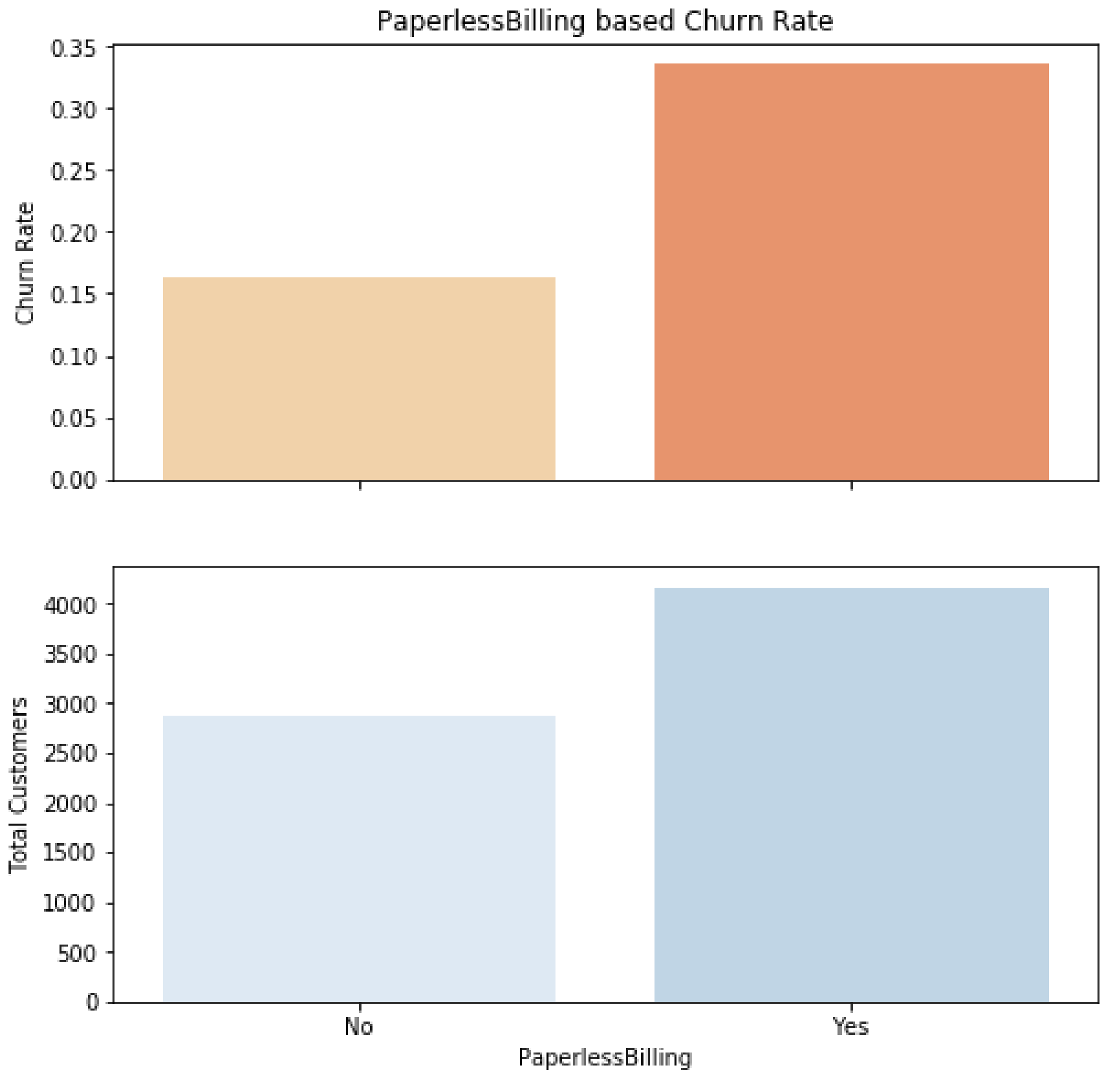
	StreamingTV	mean
0	No	0.335231
1	No internet service	0.074050
2	Yes	0.300702



	StreamingMovies	mean
0	No	0.336804
1	No internet service	0.074050
2	Yes	0.299414



	PaperlessBilling	mean
0	No	0.163301
1	Yes	0.335651

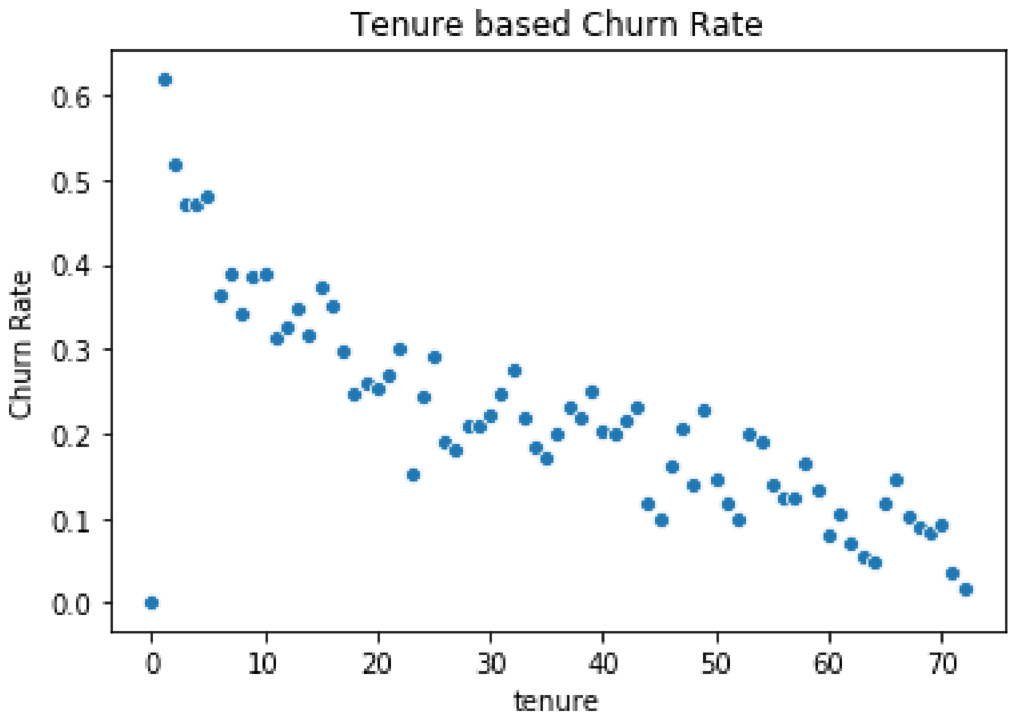


Other indicative columns are: Partner, Online Security, Online Backup, Paperless Billing.

We are done with the categorical features. Let's see how numerical features look like.

Tenure To see the trend between Tenure and average Churn Rate, let's build a scatter plot:

```
[Text(0, 0.5, 'Churn Rate'), Text(0.5, 1.0, 'Tenure based Churn Rate')]
```

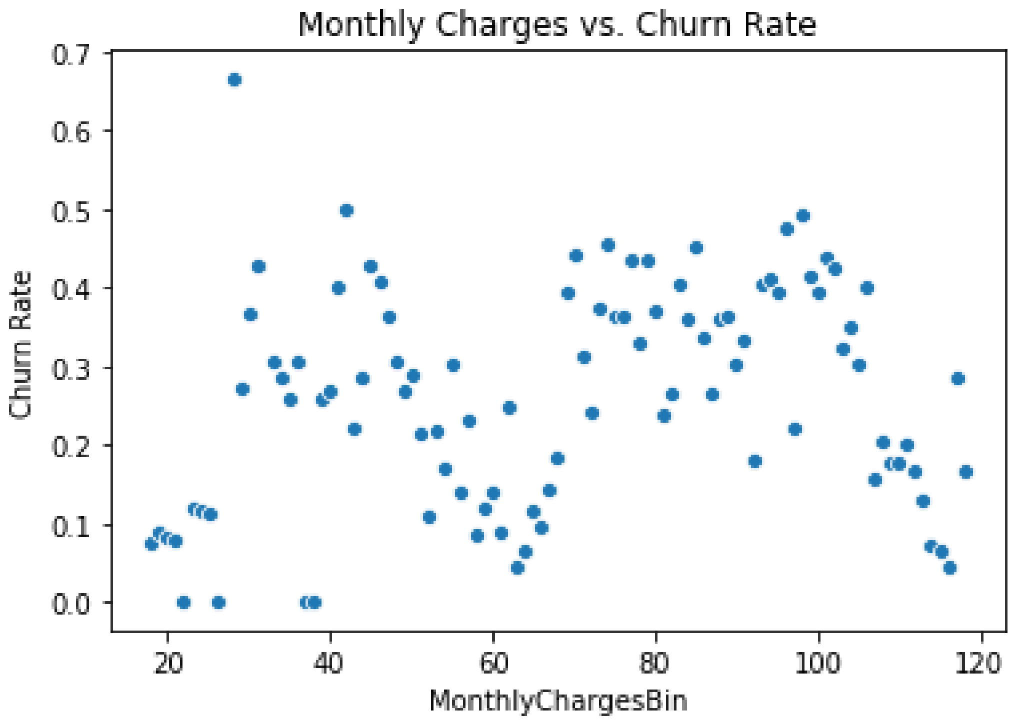


Super apparent that the higher tenure means lower Churn Rate. We are going to apply the same for Monthly and Total Charges.

Monthly Charges

```
count    7043.000000
mean       64.761692
std        30.090047
min        18.250000
25%        35.500000
50%        70.350000
75%        89.850000
max       118.750000
Name: MonthlyCharges, dtype: float64
```

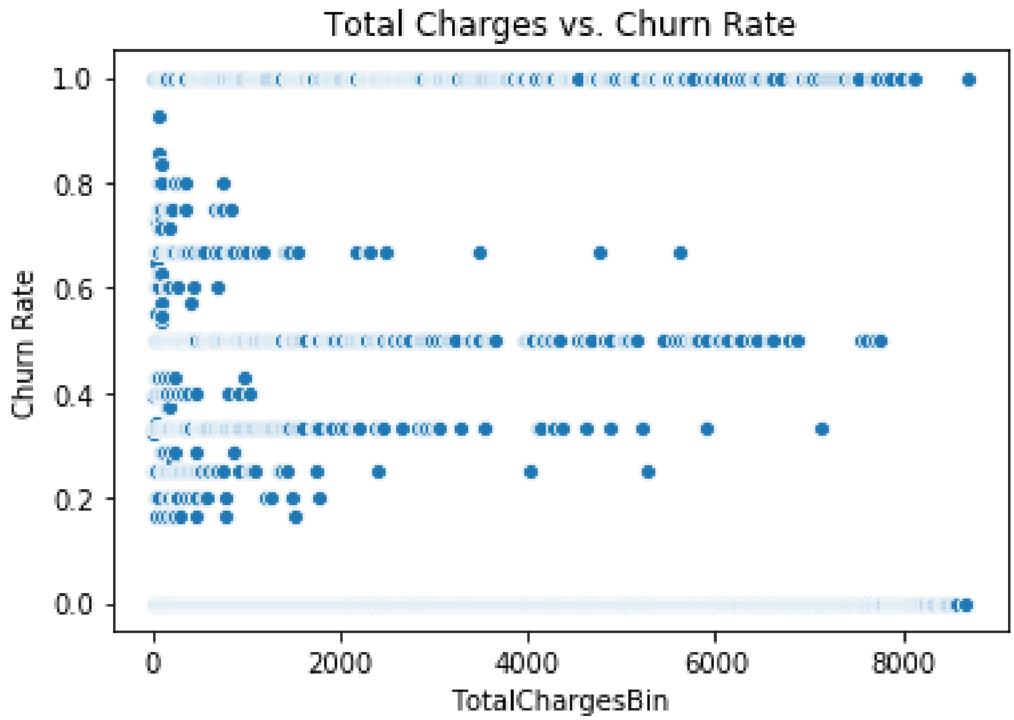
```
[Text(0, 0.5, 'Churn Rate'), Text(0.5, 1.0, 'Monthly Charges vs. Churn Rate')]
```



Total Charges

```
0      29
1    1889
2     108
3    1840
4     151
Name: TotalCharges, dtype: int32
```

```
[Text(0, 0.5, 'Churn Rate'), Text(0.5, 1.0, 'Total Charges vs. Churn Rate')]
```



Unfortunately, there is no trend between Churn Rate and Monthly & Total Charges.

1.2 Feature Engineering

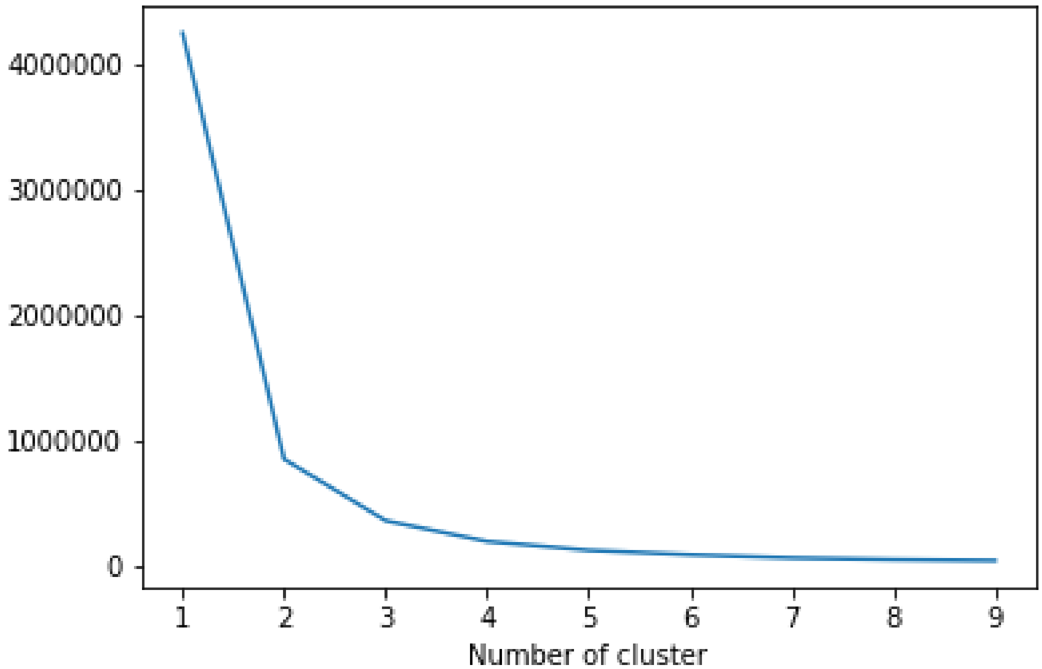
In this section, we are going to transform our raw features to extract more information from them. Our strategy is as follows: 1. Group the numerical columns by using clustering techniques 1. Apply Label Encoder to categorical features which are binary 1. Apply get_dummies() to categorical features which have multiple value

1.2.1 Numerical Columns

As we know from the EDA section, We have three numerical columns: * Tenure * Monthly Charges * Total Charges

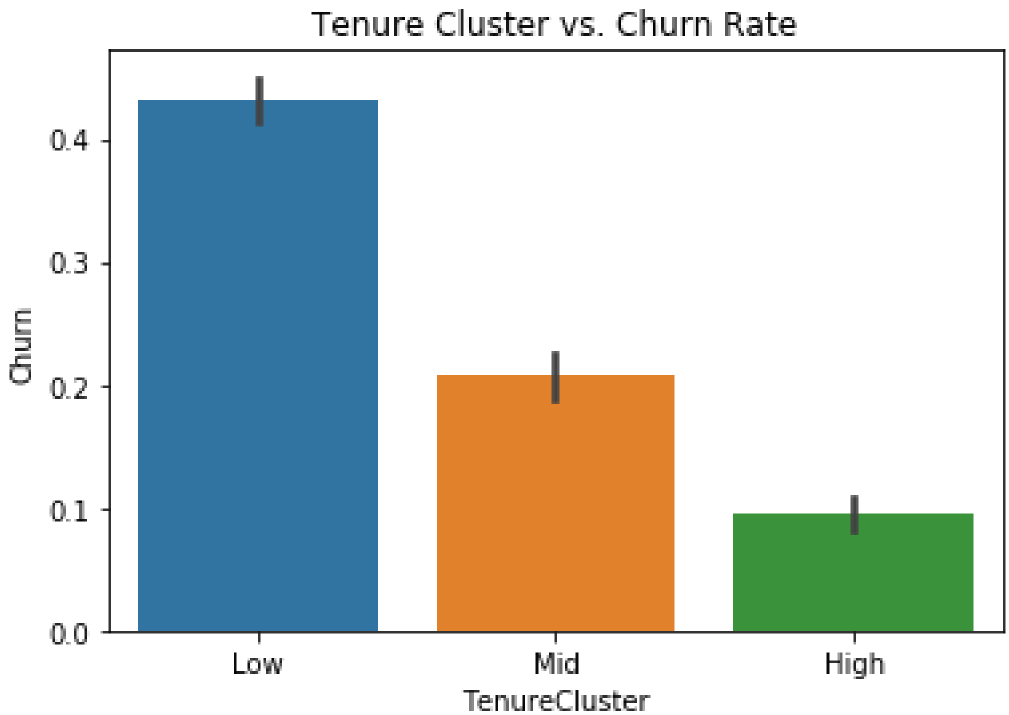
We are going to apply the following steps to create groups: * Using Elbow Method to identify the appropriate number of clusters * Applying K-means logic to the selected column and change the naming * Observe the profile of clusters

Tenure Cluster

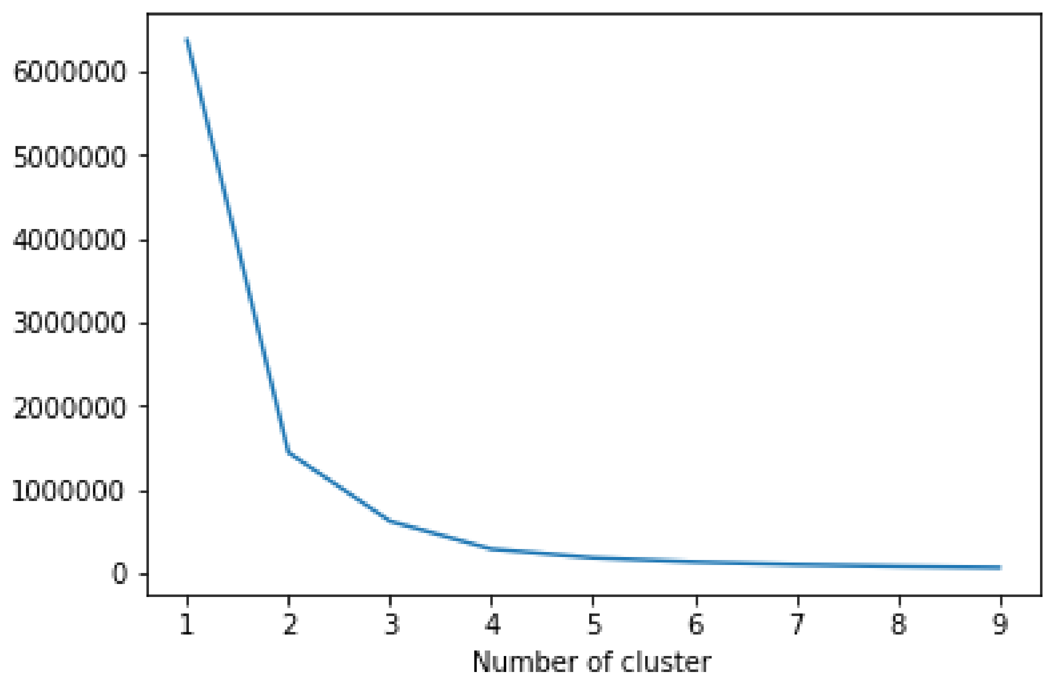


According to elbow method, for tenure we optimal choice is 3 clusters. We could go with other number if business requires so.

	count	mean	std	min	25%	50%	75%	max
TenureCluster								
High	2239.0	63.048682	7.478229	49.0	56.0	64.0	70.0	72.0
Low	2941.0	7.801428	6.227163	0.0	2.0	6.0	13.0	21.0
Mid	1863.0	34.288782	7.992701	22.0	27.0	34.0	41.0	48.0

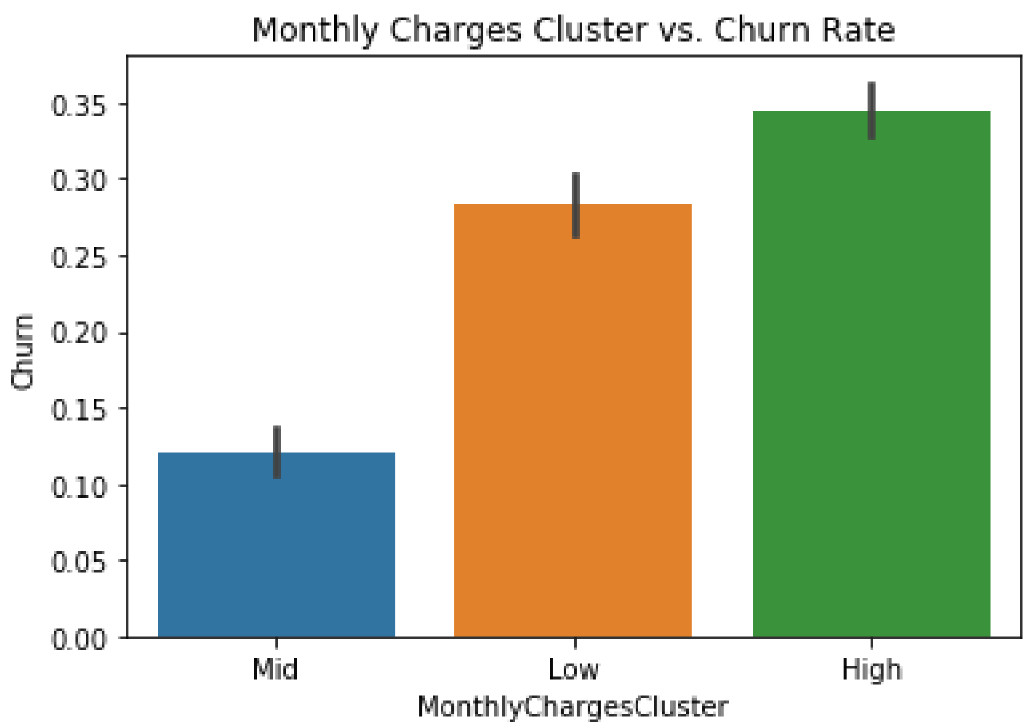


Monthly Charges This is how it looks after applying the same for Monthly & Total Charges:



According to elbow method, for MonthlyCharges we optimal choice is 3 clusters. We could go with other number if business requires so.

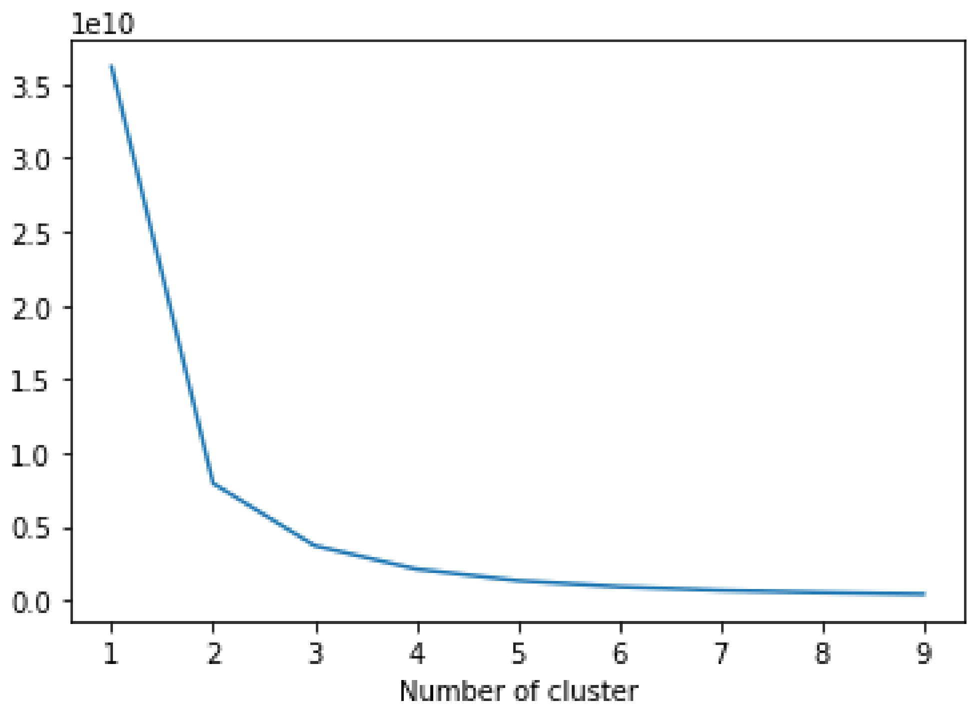
```
                      count       mean        std  min   25%   50%   75%  \
MonthlyChargesCluster
High                 2912.0  39.717720  23.984937  0.0  17.0  41.0  63.0
Low                  2239.0  25.930326  23.381947  0.0   4.0  18.0  46.0
Mid                  1892.0  28.686047  23.827175  0.0   7.0  23.0  49.0

                      max
MonthlyChargesCluster
High                 72.0
Low                  72.0
Mid                  72.0
```



Total Charges Total charges after converting to numeric hace few NA values. Those are customers that just signed up and didn't receice their first invoice yet or only received single invoice.

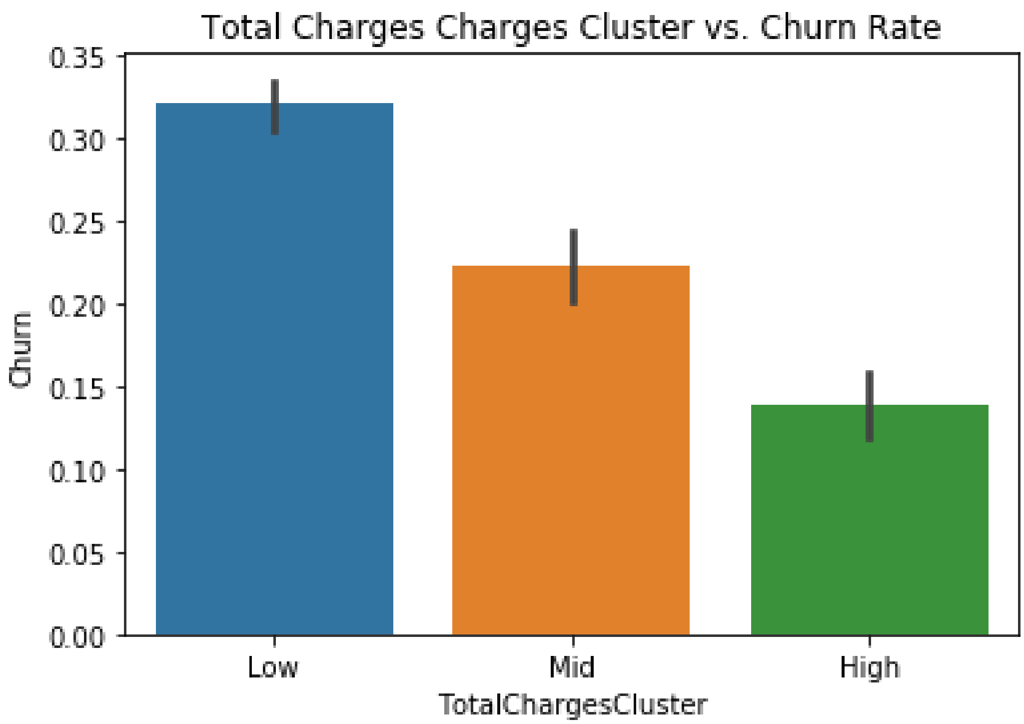
```
     tenure  MonthlyCharges  TotalCharges
92        0           20.25           NaN
138       0           25.75           NaN
425       0           19.85           NaN
```

```
488        0           25.35          NaN
566        0           20.00          NaN
681        0           19.70          NaN
1977       0           52.55          NaN
2116       0           56.05          NaN
3016       0           73.35          NaN
3029       0           61.90          NaN
4252       0           80.85          NaN

      tenure  MonthlyCharges  TotalCharges
92         0           20.25         20.25
138        0           25.75         25.75
425        0           19.85         19.85
488        0           25.35         25.35
566        0           20.00         20.00
681        0           19.70         19.70
1977       0           52.55         52.55
2116       0           56.05         56.05
3016       0           73.35         73.35
3029       0           61.90         61.90
4252       0           80.85         80.85
```



According to elbow method, for MonthlyCharges we optimal choice is 3 clusters. We could go with other number if business requires so.

```
                      count       mean        std   min   25%   50%   75%  \
TotalChargesCluster
High                 1259.0  64.373312   7.420728  43.0  59.0  66.0  71.0
Low                  4171.0  18.173100  19.185982   0.0   3.0  12.0  24.0
Mid                  1613.0  44.106634  13.433636  19.0  33.0  43.0  54.0

                      max
TotalChargesCluster
High                 72.0
Low                  72.0
Mid                  72.0
```



1.2.2 Categorical Columns

Before using categorical columns we need to convert them from lables to numbers. Two approaches are availiable: * Label Encoder converts categorical columns to numerical by simply assigning integers to distinct values. For instance, the column gender has two values: Female & Male. Label encoder will convert it to 1 and 0. * get_dummies() method creates new columns out of categorical ones by assigning 0 & 1s

Let's use both to handle remaining columns.

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 7043 entries, 0 to 7042
Data columns (total 24 columns):
```

```
customerID               7043 non-null object
gender                   7043 non-null object
SeniorCitizen            7043 non-null int64
Partner                  7043 non-null object
Dependents               7043 non-null object
tenure                   7043 non-null int64
PhoneService             7043 non-null object
MultipleLines            7043 non-null object
InternetService          7043 non-null object
OnlineSecurity           7043 non-null object
OnlineBackup             7043 non-null object
DeviceProtection         7043 non-null object
TechSupport              7043 non-null object
StreamingTV              7043 non-null object
StreamingMovies          7043 non-null object
Contract                 7043 non-null object
PaperlessBilling         7043 non-null object
PaymentMethod            7043 non-null object
MonthlyCharges           7043 non-null float64
TotalCharges             7043 non-null float64
Churn                    7043 non-null int64
TenureCluster            7043 non-null object
MonthlyChargesCluster    7043 non-null object
TotalChargesCluster      7043 non-null object
dtypes: float64(2), int64(3), object(19)
memory usage: 1.7+ MB
```

Check out how the data looks like for the selected columns:

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 7043 entries, 0 to 7042
Data columns (total 51 columns):
customerID                                 7043 non-null object
gender                                     7043 non-null int32
SeniorCitizen                              7043 non-null int64
Partner                                    7043 non-null int32
Dependents                                 7043 non-null int32
tenure                                     7043 non-null int64
PhoneService                               7043 non-null int32
PaperlessBilling                           7043 non-null int32
MonthlyCharges                             7043 non-null float64
TotalCharges                               7043 non-null float64
Churn                                      7043 non-null int64
MultipleLines_No                           7043 non-null uint8
MultipleLines_No phone service             7043 non-null uint8
MultipleLines_Yes                          7043 non-null uint8
InternetService_DSL                        7043 non-null uint8
InternetService_Fiber optic                7043 non-null uint8
InternetService_No                         7043 non-null uint8
```

```
OnlineSecurity_No                          7043 non-null uint8
OnlineSecurity_No internet service         7043 non-null uint8
OnlineSecurity_Yes                         7043 non-null uint8
OnlineBackup_No                            7043 non-null uint8
OnlineBackup_No internet service           7043 non-null uint8
OnlineBackup_Yes                           7043 non-null uint8
DeviceProtection_No                        7043 non-null uint8
DeviceProtection_No internet service       7043 non-null uint8
DeviceProtection_Yes                       7043 non-null uint8
TechSupport_No                             7043 non-null uint8
TechSupport_No internet service            7043 non-null uint8
TechSupport_Yes                            7043 non-null uint8
StreamingTV_No                             7043 non-null uint8
StreamingTV_No internet service            7043 non-null uint8
StreamingTV_Yes                            7043 non-null uint8
StreamingMovies_No                         7043 non-null uint8
StreamingMovies_No internet service        7043 non-null uint8
StreamingMovies_Yes                        7043 non-null uint8
Contract_Month-to-month                    7043 non-null uint8
Contract_One year                          7043 non-null uint8
Contract_Two year                          7043 non-null uint8
PaymentMethod_Bank transfer (automatic)    7043 non-null uint8
PaymentMethod_Credit card (automatic)      7043 non-null uint8
PaymentMethod_Electronic check             7043 non-null uint8
PaymentMethod_Mailed check                 7043 non-null uint8
TenureCluster_High                         7043 non-null uint8
TenureCluster_Low                          7043 non-null uint8
TenureCluster_Mid                          7043 non-null uint8
MonthlyChargesCluster_High                 7043 non-null uint8
MonthlyChargesCluster_Low                  7043 non-null uint8
MonthlyChargesCluster_Mid                  7043 non-null uint8
TotalChargesCluster_High                   7043 non-null uint8
TotalChargesCluster_Low                    7043 non-null uint8
TotalChargesCluster_Mid                    7043 non-null uint8
dtypes: float64(2), int32(5), int64(3), object(1), uint8(40)
memory usage: 1.1+ MB
```

	gender	Partner	TenureCluster_High	TenureCluster_Low	TenureCluster_Mid
0	0	1	0	1	0
1	0	0	0	1	0
2	1	0	0	1	0
3	1	0	0	1	0
4	1	1	0	1	0

As we can see easily, gender & Partner columns became numerical ones, and we have three new columns for TenureCluster.

It is time to fit a logistic regression model and extract insights to make better business decisions.

1.3 Logistic Regression

Predicting churn is a binary classification problem. Customers either churn or retain in a given period. Along with being a robust model, Logistic Regression provides interpretable outcomes too. As we did before, let's sort out our steps to follow for building a Logistic Regression model: 1. Prepare the data (inputs for the model) 1. Fit the model and see the model summary

And the summary looks like below:

```
                 Generalized Linear Model Regression Results
==============================================================================
Dep. Variable:                  Churn   No. Observations:                 7043
Model:                            GLM   Df Residuals:                     7013
Model Family:                Binomial   Df Model:                           29
Link Function:                  logit   Scale:                          1.0000
Method:                          IRLS   Log-Likelihood:                -2901.2
Date:                Thu, 12 Dec 2019   Deviance:                       5802.4
Time:                        17:09:20   Pearson chi2:                 7.61e+03
No. Iterations:                   100   Covariance Type:             nonrobust
================================================================================
===========================
                                            coef    std err          z
P>|z|      [0.025      0.975]
--------------------------------------------------------------------------------
---------------------------
Intercept                                 0.2509      0.276      0.908
0.364      -0.291       0.792
gender                                   -0.0249      0.065     -0.383
0.702      -0.152       0.103
SeniorCitizen                             0.2236      0.085      2.638
0.008       0.057       0.390
Partner                                   0.0011      0.078      0.013
0.989      -0.152       0.154
Dependents                               -0.1386      0.090     -1.539
0.124      -0.315       0.038
tenure                                   -0.0644      0.008     -7.668
0.000      -0.081      -0.048
PhoneService                              0.2292      0.403      0.569
0.569      -0.560       1.018
PaperlessBilling                          0.3476      0.075      4.647
0.000       0.201       0.494
MonthlyCharges                           -0.0336      0.032     -1.055
0.292      -0.096       0.029
TotalCharges                              0.0001   9.98e-05      1.260
0.208   -6.98e-05       0.000
MultipleLines_No                         -0.1126      0.130     -0.869
0.385      -0.367       0.141
MultipleLines_No_phone_service            0.0217      0.160      0.136
0.892      -0.291       0.335
```

```
MultipleLines_Yes                            0.3418      0.283      1.208
0.227      -0.213       0.896
InternetService_DSL                         -0.5957      0.226     -2.637
0.008      -1.039      -0.153
InternetService_Fiber_optic                  1.0419      0.577      1.804
0.071      -0.090       2.174
InternetService_No                          -0.1953      0.091     -2.145
0.032      -0.374      -0.017
OnlineSecurity_No                            0.3267      0.108      3.023
0.002       0.115       0.538
OnlineSecurity_No_internet_service          -0.1953      0.091     -2.145
0.032      -0.374      -0.017
OnlineSecurity_Yes                           0.1195      0.261      0.458
0.647      -0.392       0.631
OnlineBackup_No                              0.2216      0.107      2.075
0.038       0.012       0.431
OnlineBackup_No_internet_service            -0.1953      0.091     -2.145
0.032      -0.374      -0.017
OnlineBackup_Yes                             0.2246      0.260      0.863
0.388      -0.286       0.735
DeviceProtection_No                          0.1464      0.107      1.365
0.172      -0.064       0.357
DeviceProtection_No_internet_service        -0.1953      0.091     -2.145
0.032      -0.374      -0.017
DeviceProtection_Yes                         0.2998      0.260      1.151
0.250      -0.211       0.810
TechSupport_No                               0.3129      0.108      2.903
0.004       0.102       0.524
TechSupport_No_internet_service             -0.1953      0.091     -2.145
0.032      -0.374      -0.017
TechSupport_Yes                              0.1332      0.262      0.509
0.611      -0.380       0.646
StreamingTV_No                              -0.0566      0.048     -1.177
0.239      -0.151       0.038
StreamingTV_No_internet_service             -0.1953      0.091     -2.145
0.032      -0.374      -0.017
StreamingTV_Yes                              0.5027      0.339      1.481
0.139      -0.162       1.168
StreamingMovies_No                          -0.0575      0.048     -1.186
0.236      -0.152       0.038
StreamingMovies_No_internet_service         -0.1953      0.091     -2.145
0.032      -0.374      -0.017
StreamingMovies_Yes                          0.5036      0.339      1.484
0.138      -0.162       1.169
Contract_Month_to_month                      0.7777      0.118      6.611
0.000       0.547       1.008
Contract_One_year                            0.0953      0.121      0.788
0.431      -0.142       0.332
```

```
Contract_Two_year                          -0.6222      0.148     -4.195
0.000      -0.913      -0.331
PaymentMethod_Bank_transfer__automatic_     0.0308      0.097      0.316
0.752      -0.160       0.221
PaymentMethod_Credit_card__automatic_      -0.0538      0.099     -0.546
0.585      -0.247       0.139
PaymentMethod_Electronic_check              0.3224      0.087      3.727
0.000       0.153       0.492
PaymentMethod_Mailed_check                 -0.0485      0.097     -0.500
0.617      -0.239       0.142
TenureCluster_High                          0.5670      0.187      3.028
0.002       0.200       0.934
TenureCluster_Low                          -0.1745      0.172     -1.017
0.309      -0.511       0.162
TenureCluster_Mid                          -0.1417      0.119     -1.190
0.234      -0.375       0.092
MonthlyChargesCluster_High                  0.0592      0.169      0.351
0.726      -0.272       0.390
MonthlyChargesCluster_Low                   0.0749      0.127      0.591
0.555      -0.174       0.324
MonthlyChargesCluster_Mid                   0.1168      0.195      0.600
0.548      -0.265       0.498
TotalChargesCluster_High                    0.3622      0.206      1.754
0.079      -0.042       0.767
TotalChargesCluster_Low                    -0.2718      0.177     -1.535
0.125      -0.619       0.075
TotalChargesCluster_Mid                     0.1605      0.122      1.316
0.188      -0.079       0.400
================================================================================
===========================
```

We have two important outcomes from this report. When you prepare a Churn Prediction model, you will be faced with the questions below: 1. Which characteristics make customers churn or retain? 1. What are the most critical ones? What should we focus on?

For the first question, you should look at the 4th column (P>|z|). If the absolute p-value is smaller than 0.05, it means, that feature affects Churn in a statistically significant way. Examples are: * SeniorCitizen * InternetService_DSL * OnlineSecurity_NO

Then the second question. We want to reduce the Churn Rate, where we should start? The scientific version of this question is;

> *Which feature will bring the best ROI if I increase/decrease it by one unit?*

That question can be answered by looking at the coef column. Exponential coef gives us the expected change in Churn Rate if we change it by one unit. If we apply the code below, we will see the transformed version of all coefficients:

```
Intercept                                  1.285160
gender                                     0.975395
```

```
SeniorCitizen                            1.250605
Partner                                  1.001054
Dependents                               0.870603
tenure                                   0.937642
PhoneService                             1.257575
PaperlessBilling                         1.415616
MonthlyCharges                           0.966963
TotalCharges                             1.000126
MultipleLines_No                         0.893466
MultipleLines_No_phone_service           1.021935
MultipleLines_Yes                        1.407524
InternetService_DSL                      0.551151
InternetService_Fiber_optic              2.834684
InternetService_No                       0.822587
OnlineSecurity_No                        1.386373
OnlineSecurity_No_internet_service       0.822587
OnlineSecurity_Yes                       1.126925
OnlineBackup_No                          1.248082
OnlineBackup_No_internet_service         0.822587
OnlineBackup_Yes                         1.251792
DeviceProtection_No                      1.157608
DeviceProtection_No_internet_service     0.822587
DeviceProtection_Yes                     1.349626
TechSupport_No                           1.367448
TechSupport_No_internet_service          0.822587
TechSupport_Yes                          1.142521
StreamingTV_No                           0.945018
StreamingTV_No_internet_service          0.822587
StreamingTV_Yes                          1.653236
StreamingMovies_No                       0.944158
StreamingMovies_No_internet_service      0.822587
StreamingMovies_Yes                      1.654743
Contract_Month_to_month                  2.176561
Contract_One_year                        1.100015
Contract_Two_year                        0.536769
PaymentMethod_Bank_transfer__automatic_  1.031241
PaymentMethod_Credit_card__automatic_    0.947615
PaymentMethod_Electronic_check           1.380499
PaymentMethod_Mailed_check               0.952640
TenureCluster_High                       1.763036
TenureCluster_Low                        0.839915
TenureCluster_Mid                        0.867882
MonthlyChargesCluster_High               1.060993
MonthlyChargesCluster_Low                1.077803
MonthlyChargesCluster_Mid                1.123842
TotalChargesCluster_High                 1.436438
TotalChargesCluster_Low                  0.762027
```

```
TotalChargesCluster_Mid                          1.174086
dtype: float64
```

As an example, one unit change in Monthly Charge (coef. 0.965881) means ~3.4% improvement in the odds for churning if we keep everything else constant. From the table above, we can quickly identify which features are more important. Now, everything is ready for building our classification model.

1.4 Binary Classification Model with XGBoost

To fit XGBoost to our data, we should prepare features (X) and label(y) sets and do the train & test split.

```
Accuracy of XGB classifier on training set: 0.84
Accuracy of XGB classifier on test set: 0.82
```

By using this simple model, we have achieved 83% accuracy,

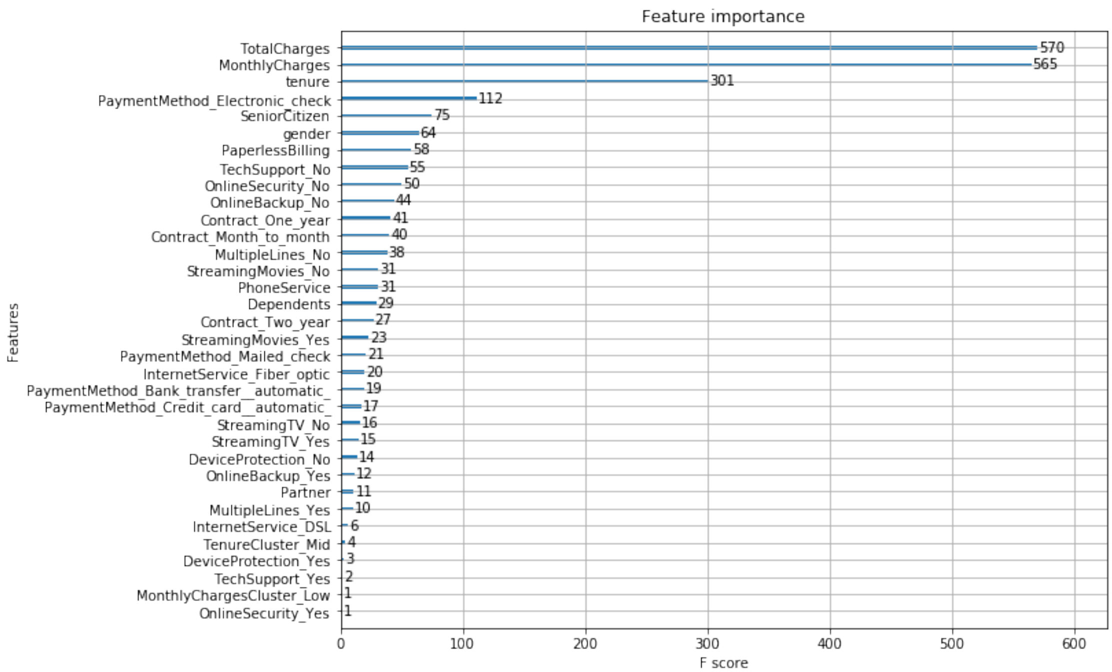
Our actual Churn Rate in the dataset was 26.5% (reflects as 73.5% for baseline model performance). This shows our model is a useful one. Better to check our classification model to see where exactly our model fails.

```
              precision    recall  f1-score   support

           0       0.86      0.92      0.89       265
           1       0.69      0.53      0.60        88

   micro avg       0.82      0.82      0.82       353
   macro avg       0.77      0.73      0.74       353
weighted avg       0.82      0.82      0.82       353
```

We can interpret the report above as if our model tells us, 100 customers will churn, 70 of it will churn (0.70 precision). And actually, there are around 170 customers who will churn (0.58 recall). Especially recall is the main problem here, and we can improve our model's overall performance by: * Adding more data (we have around 2000 rows for this example) * Adding more features * More feature engineering * Trying other models * Hyper-parameter tuning

Moving forward, let's see how our model works in detail. First off, we want to know which features our model exactly used from the dataset. Also, which were the most important ones? For addressing this question, we can use the code below:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1bcd234e588>
```



We can see that our model assigned more importance to **TotalCharges** and **MonthlyCharges** compared to others.

Finally, the best way to use this model is assigning Churn Probability for each customer, create segments, and build strategies on top of that. Below we get the churn probability from our model:

	customerID	proba
0	7590-VHVEG	0.631970
1	6713-OKOMC	0.189523
2	7469-LKBCI	0.013183
3	8779-QRDMV	0.885242
4	1680-VDCWW	0.034147

Now we know if there are likely to churn customers in our best segments and we can build actions based on it!