Matthew Thomas Prescher

Final Project

CIS 3330

12/12/2023

Introduction

Data set can be found at: https://www.kaggle.com/datasets/blastchar/telco-customer-churn

The dataset contains various attributes of customers of Telco. Telco offers a wide variety of services including, phone, internet and TV. All information of a customer can be found in the dataset. My job is to disect the information to see what can be used to predict the Churn of a customer. By understanding what attributes affect Churn, Telco can use that infomration to try and change it services to consist of more loyal and supportive Customer base. This report consist of my findings on the data set and include the following: Intial Data View, Descriptive Analysis, Predictive Analysis and Findings. Each section contains information that will help in understanding my thought process and how I came the final result of my findings.

In [2]:
```
#Import all packages
%matplotlib inline
import pandas as pd
import warnings
import seaborn as sns
import statsmodels.api as sm
import statsmodels.formula.api as sm_api
import matplotlib.pylab as plt
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
from sklearn.metrics import precision_score
from sklearn.metrics import f1_score
from sklearn.neighbors import KNeighborsClassifier
from sklearn.naive_bayes import GaussianNB, MultinomialNB
from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier, DecisionTreeRegressor
from sklearn.ensemble import RandomForestClassifier
from sklearn.svm import SVC
from vaderSentiment.vaderSentiment import SentimentIntensityAnalyzer
from ucimlrepo import fetch_ucirepo
from dmba import plotDecisionTree, classificationSummary, regressionSummary
```

Initial Data View (IDV)

ID1

In [3]:
```
#import csv file
df = pd.read_csv('TelcoCustomerChurn.csv')
print(df.shape)
for i in df.columns:
    print(i)
df.head()
```

```
(7043, 21)
customerID
gender
SeniorCitizen
Partner
Dependents
tenure
PhoneService
MultipleLines
InternetService
OnlineSecurity
OnlineBackup
DeviceProtection
TechSupport
StreamingTV
StreamingMovies
Contract
PaperlessBilling
PaymentMethod
MonthlyCharges
TotalCharges
Churn
```

Out[3]:

	customerID	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetService	OnlineSecurity	...	DeviceProtection	TechSupport	StreamingTV	StreamingMovies	Contract	PaperlessB
0	7590-VHVEG	Female	0	Yes	No	1	No	No phone service	DSL	No	...	No	No	No	No	Month-to-month	
1	5575-GNVDE	Male	0	No	No	34	Yes	No	DSL	Yes	...	Yes	No	No	No	One year	
2	3668-QPYBK	Male	0	No	No	2	Yes	No	DSL	Yes	...	No	No	No	No	Month-to-month	
3	7795-CFOCW	Male	0	No	No	45	No	No phone service	DSL	Yes	...	Yes	Yes	No	No	One year	
4	9237-HQITU	Female	0	No	No	2	Yes	No	Fiber optic	No	...	No	No	No	No	Month-to-month	

5 rows × 21 columns

IDV2

In [4]:
```
#creating dummy variables for gender, tenure, Contract, Partner,Dependents

cls_df = df[['SeniorCitizen','gender','tenure','Partner','Dependents'
             ,'Contract','TechSupport','Churn']]
cls_df= pd.get_dummies(cls_df, columns = ['Partner','Dependents'
            ,'Contract','TechSupport'])
cls_df[['Churn']] = cls_df[['Churn']].replace({'Yes':1,'No':0})
cls_df['gender'] = cls_df['gender'].replace({'Male':0,'Female':1})
cls_df.head()
```

Out[4]:

	SeniorCitizen	gender	tenure	Churn	Partner_No	Partner_Yes	Dependents_No	Dependents_Yes	Contract_Month-to-month	Contract_One year	Contract_Two year	TechSupport_No	TechSupport_No internet service	TechSupport_Yes
0	0	1	1	0	0	1	1	0	1	0	0	1	0	0
1	0	0	34	0	1	0	1	0	0	1	0	1	0	0
2	0	0	2	1	1	0	1	0	1	0	0	1	0	0
3	0	0	45	0	1	0	1	0	0	1	0	0	0	1
4	0	1	2	1	1	0	1	0	1	0	0	1	0	0

The small excerpt from the data in IDV1 shows us that most of our data is categorical. This relays that we will need to create dummy variables for the fields that may seem significant. Churn will also be changed from yes and no to binary values. 1 is equal to yes and 0 is equal to no. I am not creating a dummy variable for churn as I want to keep it isolated to one column.This will help with understanding correlations and facilitate a more comprehensive analysis.

In IDV2 I am 'cleaning' the data to be able to process it and use it in our analysis. By creating dummy variables we are able to give a mathematical significance to our variables for visualizations and correlation tables. This also allows us to make use of Classification Methods. In this analysis we will be making use of Decision Trees, and Nearest Neighbors.

Descriptive Analysis (DA)

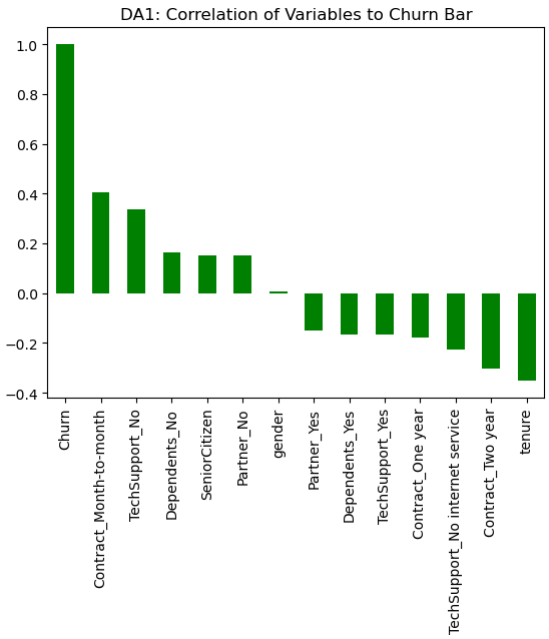
Correlation Matrix and Bar graph

Resource:

https://stackoverflow.com/questions/35420642/how-to-plot-a-graph-for-correlation-co-efficient-between-each-attributes-of-a-da

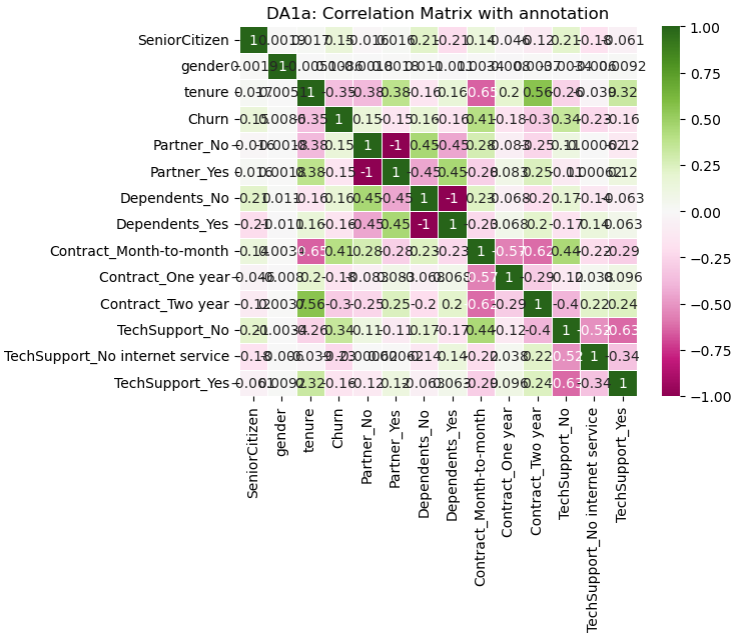
https://www.geeksforgeeks.org/sort-correlation-matrix-in-python/

In [9]:
```
##correlation becomes hard to visualize with so many aggregated components
corr = cls_df.corr()
#sets correlation so that all values are correlations to Churn
corr['Churn'].sort_values(ascending = False).plot(kind ='bar',color= 'g',
                                                  title = 'DA1: Correlation of Variables to Churn Bar')
plt.show()
```



In [34]:
```
ax = plt.axes()
sns.heatmap(corr,annot = True, linewidth = .5, cmap = 'PiYG', ax = ax)
ax.set_title('DA1a: Correlation Matrix with annotation')
```

Out[34]:
```
Text(0.5, 1.0, 'DA1a: Correlation Matrix with annotation')
```

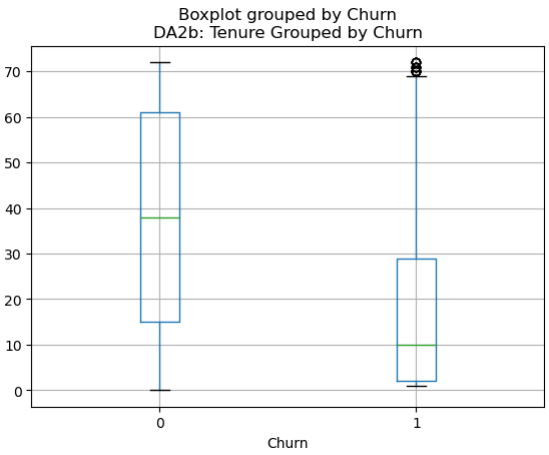


The correlation matrix (DA1c) is transformed into a easily readable correlation bar graph (DA1). In doing so it is easier to distinguish which variables are most correlated with Churn. We can see that tenure and Contract_Month-to-month are moderatly correlated to Churn and are above the rest of the variables. Now, using descriptive analytics we can see and identify there relationships more closely. Not only are we able to see what attributes are valuable to Churn but which are not. Multiple lines, Phone Service, Streaming TV, Device Protection and gender are not significant in the determination of if someone churns. These have been dropped in an effort to only include variables that are significant. However, gender should remain for although not significant in correlation it is useful in understanding our customer base.

Box Plot

In [50]:
```
cls_df.boxplot(column = ['tenure'],by='Churn').set_title('DA2b: Tenure Grouped by Churn')
```

Out[50]:
```
Text(0.5, 1.0, 'DA2b: Tenure Grouped by Churn')
```

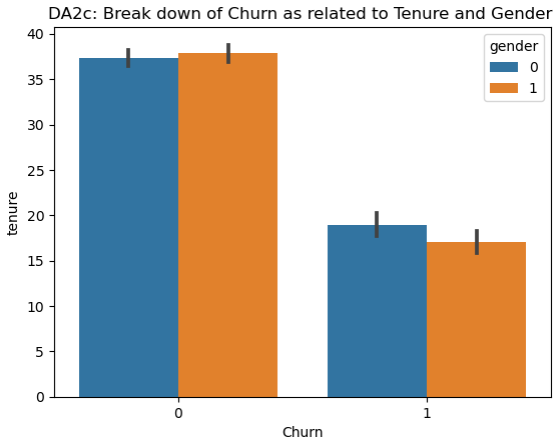


The box plot above, DA2b, showing tenure in correlation with Churn. Tenure in the dataset is described as how long a customer has been with the company in months. The box plot gives us a visual representation that the longer someone stays with the company they less likely they are to Churn. This is not always the case as we can see that the range of Churn (value 1) is near the highest point of those who do not Churn. Also there are some outliers of those who have been with the company for a long time and still Churn. However, from the box plot we can see that individuals are less likely to Churn the longer thier tenure.

Bar Plots

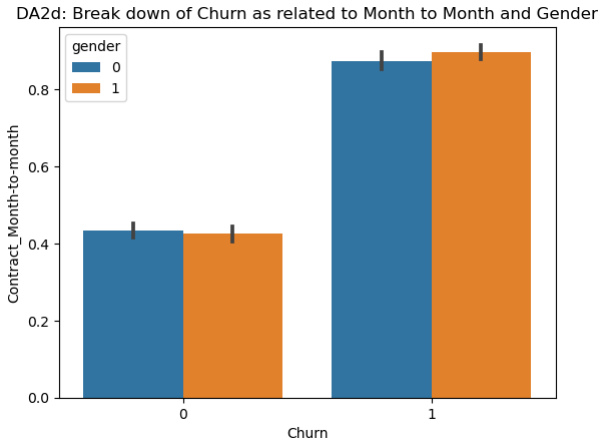
In [51]:
```
sns.barplot(x='Churn',y='tenure',hue='gender',data = cls_df).set_title(
    'DA2c: Break down of Churn as related to Tenure and Gender ')
```

Out[51]:
```
Text(0.5, 1.0, 'DA2c: Break down of Churn as related to Tenure and Gender ')
```



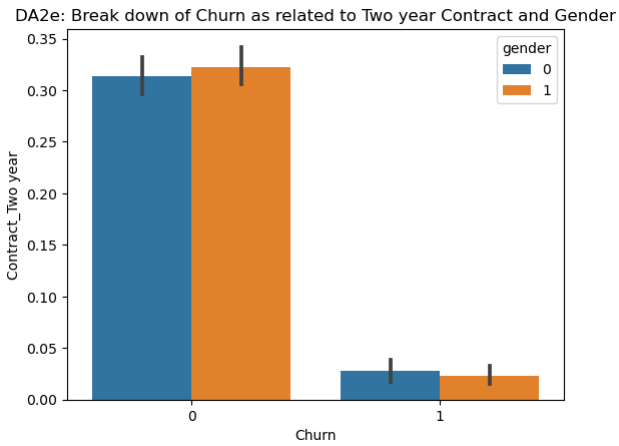
In [52]:
```
sns.barplot(x='Churn',y = 'Contract_Month-to-month',hue='gender',data= cls_df).set_title(
    'DA2d: Break down of Churn as related to Month to Month and Gender')
```

Out[52]:
```
Text(0.5, 1.0, 'DA2d: Break down of Churn as related to Month to Month and Gender')
```



In [37]:
```
sns.barplot(x='Churn',y='Contract_Two year', hue = 'gender', data = cls_df).set_title(
    'DA2e: Break down of Churn as related to Two year Contract and Gender')
```

Out[37]:
```
Text(0.5, 1.0, 'DA2e: Break down of Churn as related to Two year Contract and Gender')
```



In DA2c we can see that there is a visible differnce between the length of tenure and the churn rate. This relation is imporatant to the firm and shoudl be used later to identify how the company can maximize the tenure of its customers.

DA2d shows us that commiting to a month to month contract has a very high churn rate compared to not having a one. By not having one means that the customer has a different contract. Possibly one that has a longer commitment.

DA2e builds upon the relsult of DA2d. Having a month to month contract has very high churn. However, an individual that has a 2 year contract is much more likely to stay, possibly even after thier two year contract is up.

Predictive Analytics: Classification Models

Preparing Data for use in Classification Methods.

In [53]:
```
#dropping any possible null values as they will mess up the Learning Algorithms
cls_df = cls_df.dropna()
#setting the variable we want predicted to be Churn
y = cls_df['Churn']
#setting all other variables besides Churn to be indepedent variables.
X= cls_df.drop(columns=['Churn'])

#Splitting the dataset into 70 percent training data and 30 percent testing
X_train,X_test,y_train,y_test = train_test_split(X,y, test_size= .3, random_state = 105)
```

Decision Tree

Resources: https://stackoverflow.com/questions/54812230/sklearn-min-impurity-decrease-explanation

In [54]:
```
x = 0
warnings.filterwarnings('ignore')
# using i as indicator of the depth to see at which depth the highest precision can be achieved
for i in range(1,100):
    mTree = DecisionTreeClassifier(criterion='entropy', random_state=45, max_depth=i)
    mTree.fit(X_train,y_train)
    y_hat = mTree.predict(X_test)
    accuracy = accuracy_score(y_test,y_hat)
    if accuracy >x:
        x = round(accuracy,2)
        depth = i
        precision = round(precision_score(y_test,y_hat),2)
print(f"With depth of {depth} our accuracy is:{x}")
print(f"With depth of {depth} our precision is: {precision}")
```

```
With depth of 5 our accuracy is:0.78
With depth of 5 our precision is: 0.61
```

In [55]:
```
#Setting the sct or Small Class Tree for visualizeation and use in a confusion Matrix
sct = DecisionTreeClassifier(max_depth = 5,min_impurity_decrease = .01)
#fitting the data into the sct
sct.fit(X_train,y_train)
```

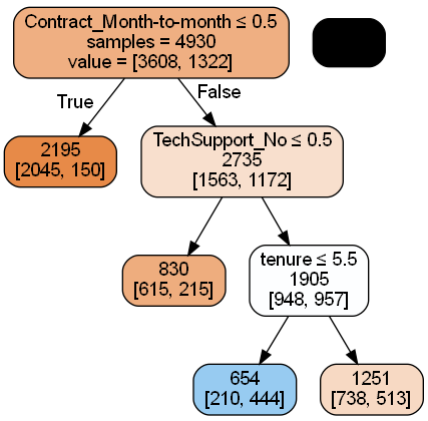
Out[55]:
```
DecisionTreeClassifier(max_depth=5, min_impurity_decrease=0.01)
```

In [56]:
```
print('Confusion Matrix for Train data')
# allows us to see the quanity of predictions missed by the Tree model for the train data set
classificationSummary(y_train,sct.predict(X_train))

print('\n\nConfusion Matrix for Test data')
# Allows us to see the True and false positives/negatives. A visual representation in a way of how our model is preforming.
classificationSummary(y_test,sct.predict(X_test))
```

```
Confusion Matrix for Train data
Confusion Matrix (Accuracy 0.7793)

       Prediction
Actual    0    1
     0 3398  210
     1  878  444


Confusion Matrix for Test data
Confusion Matrix (Accuracy 0.7733)

       Prediction
Actual    0    1
     0 1475   91
     1  388  159
```

In [49]:
```
#to visualize the tree and the criteria in which it breaks it down.
plotDecisionTree(sct, feature_names = X_train.columns)
```

Out[49]:



The decision tree functions by looking at certain criteria and learning in what ways it can break down the data recieved. In breaking it down it classifies or predicts the most likely output. In this case we are the prediction is whether or not the customer will Churn. By training the model it is able to accurately predict the churn of customers 78 percent of the time. The precision of the model is lower at a 61 percent. This just means that the accuracy is slightly distributed and not clustered amongst the data that is accurate.

The model is then put into a confusion matrix to help see where the model is failing to identify the most. From the confusion matrix we can see that we are more likely to find False Negatives. This means that we are incorrectly classifying some customers as not churning when in reality they are. This is inconvient since we want to focus on the individuals that are likely to churn opposed to those who do not need insentives to stay.

Finally, we have the decision tree visualized and broken into characteristics for us. True refers to those who are not churning, False are those who do churn. The tree has determined that the number one factor in determining if someone will churn or not is based on if they have a Month to Month contract. If the customer does not have a Month to month contract they are selected as not churning. This break down continues and seperates customers who churn and do not.

Nearest Neighbors

In [47]:
```
nn = 10
for i in range(7, nn+1):
    knn = KNeighborsClassifier(n_neighbors = i)
    knn.fit(X_train, y_train)
    y_hat= knn.predict(X_test)
    print(f' k = {i} Accuracy: {round(accuracy_score(y_test,y_hat),2)}')
    print(f' k = {i} Precision {round(precision_score(y_test,y_hat),2)}')
    print()
```

```
 k = 7 Accuracy: 0.77
 k = 7 Precision 0.56

 k = 8 Accuracy: 0.78
 k = 8 Precision 0.61

 k = 9 Accuracy: 0.78
 k = 9 Precision 0.58

 k = 10 Accuracy: 0.78
 k = 10 Precision 0.61
```

Nearest Neighbors looks at surrounding data points to dertermine what a certain point will be. In this case, if the data points surround a point we are trying to predict are more densly populated with Churn then it will assume that point is a Churn point as well. In this dataset we can see that the model is most accurate and precise when it incorporates 8 data points. This represents that when trying to identify a new point the model accurately predicts wether or not a cusotomer will churn 78 percent of the time. This model holds the same accuarcy and precision percentages as the Decison Tree model.

Findings

After cleaning the data and breaking it down into values that can be manipulated and analyed we can make several observations about churn at Telco. The first is that gender plays has not real effect on the churn of customers. This is later solidified in the bar plots that show no difference when it comes to men and women. The second observation is that tenure, contract type and not having tech support have effects on the a customer churning or not. The most obvious correlation being that of churn and whether or not the customers have a month to month contract opposed to two or one year contracts. This is a sign for the company to tell its sales employees to push for new incoming customers to get a one or two year contract opposed to the month to month. Doing so may reduce churn in the long run. Having tech support included in your contract aslo seems to have an effect on whether or not the customer will churn. The final node in teh decision tree is affected by tenure. However, inspecting further we can see that tenure value does a poor job at identifying churn, atleast in this form of the decision tree model. I made the decision to run along side this model a Nearest Neighbors to see how the predicition compares. At a level of 8, meaning the 8 surrounding data points, the nearest neighbor model performs equally as well as the Decision Tree. Moving forward the company should continue to make use of the nearest Neighbors to monitor if the decision tree becomes more insightful or not in accuracy and precision. Overall, the company can now predict churn based on the factors presented and with this information have avenues to invest in to reduce churn rate, increasing profitability.

ps

have a merry christmas, thank you for all your teachings. I really enjoyed your classes this semester. Enjoy the holidays.