

Data Mining I – D209

Task #1 Classification Analysis

Bret McHenry

Instructor: Dr. Festus Elleh

October 16, 2024

Table of Contents

[A1. Proposal of Question 3](#_Toc179969076)

[A2. Defined Goal 3](#_Toc179969077)

[B1. Explanation of Classification Method 3](#_Toc179969078)

[B2. Summary of Method Assumption 3](#_Toc179969079)

[B3. Packages or Libraries List 3](#_Toc179969080)

[C1. Data Preprocessing 4](#_Toc179969081)

[C2. Data Set Variables 4](#_Toc179969082)

[C3. Steps for Analysis 5](#_Toc179969083)

[Univariate Visualizations 8](#_Toc179969084)

[Bivariate Visualizations 10](#_Toc179969085)

[C4. Cleaned Data Set 14](#_Toc179969086)

[D1. Splitting the Data 15](#_Toc179969087)

[D2. Output and Intermediate Calculations 15](#_Toc179969088)

[D3. Code Execution 18](#_Toc179969089)

[E1. Accuracy and AUC 18](#_Toc179969090)

[E2. Results and Implications 19](#_Toc179969091)

[E3. Limitation 19](#_Toc179969092)

[E4. Course of Action 19](#_Toc179969093)

[F. Panopto Recording 20](#_Toc179969094)

[G. Sources for Third-Party Code 20](#_Toc179969095)

[H. Sources 20](#_Toc179969096)

# A1. Proposal of Question

Can subscribed services data and marketable demographic data be utilized to predict customer churn using the k-nearest neighbor’s method?

# A2. Defined Goal

The organization would like to focus their marketing efforts towards prospective customers that show the highest retention rates. The goal will be to produce a machine learning model that aids our marketing team in identifying the subscribed services and/or key demographics that guard against churn. The results could then be leveraged to package optimal services into offerings and/or target specific demographics in order to increase tenure.

# B1. Explanation of Classification Method

The K-NN classification method is designed to classify unseen data by a model built off of the “K” closest points, known as neighbors. The model predicts the label based off of the majority of the “K” closest points. The neighbor variable or “K” is determined by hyperparameter tuning, a technique that maximizes the model accuracy score which then needs to be evaluated against under or over fitting. The expected outcome is a model that improves the organization’s existing efforts to attract customers that have a better chance to be retained.

# B2. Summary of Method Assumption

One assumption of KNN is that distance defines the classification of labels. Meaning, data points that are not in close proximity to the classification are considered dissimilar. Due to this assumption, the model will utilize standardization by scaling all continuous numerical predictors.

# B3. Packages or Libraries List

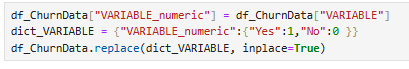
Python Libraries Utilized

|  |  |
| --- | --- |
| Package/Library | Usage |
| Pandas | Importing data into dataframe and data manipulations |
| numpy | Provide array calculations |
| missingno | Visual depiction of missing data. |
| matplotlib | Provide visualizations for Cleaning, EDA, Model Evaluation |
| scipy | Used to derive zscore via stats module |
| seaborn | Used to show boxplot visuals |
| sklearn.feature\_selection | Used to determine p\_value of candidate features (SelectKBest) |
| sklearn.preprocessing | Scale predictor variables |
| sklearn.model\_selection | Perform hold-out validation and hyperparameter tuning |
| sklearn.neighbors | Utilize KNN modeling |
| sklearn.metrics | Evaluate KNN model |
| warnings | Turn warnings off in IDE in order to print output |

# C1. Data Preprocessing

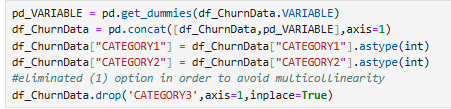
Because machine learning requires numerical data, one goal for preprocessing was to convert all categorical values prior to model evaluation. I chose to use ordinal encoding for the binary categorical variables and target variable. This was accomplished by performing the following steps:

1. Create additional numeric column in cleaned dataset
2. Create a dictionary variable holding binary values (Yes=1, No=0)
3. Replace the variable with the contents of the dictionary numeric variables



I utilized the technique of “One-Hot” encoding for the nominal categorical variables. The steps chosen to perform “One-Hot” encoding:

1. Create a dataframe with unique values of the categorical value
2. Attach the new dataframe to the clean dataframe (with concat function)
3. Convert new variables from Boolean to int (True=1, False-0)
4. Eliminate one variable in order to avoid multicollinearity. Note, this variable is represented by a “zero” condition in all of the other converted numerical variables.



# C2. Data Set Variables

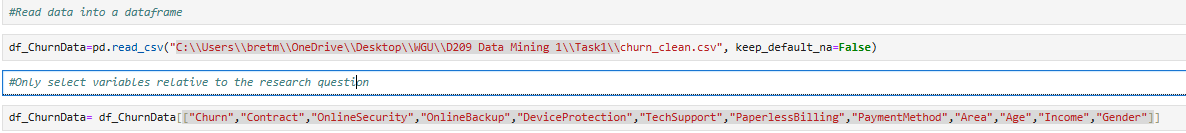
Initial set of variables selected to address research question:

|  |  |  |  |
| --- | --- | --- | --- |
| Variable Name | Data Type | Description | Example |
| Area | Categorical | Census data population category | Urban |
| Age | Numeric | Customer age | 68 |
| Income | Numeric | Customer annual income | 58634.51 |
| Gender | Categorical | Customer gender | Male |
| Contract | Categorical | Term of contract | One year |
| Churn | Categorical | Customer has stopped service within last month | Yes |
| OnlineSecurity | Categorical | Customer has opted for online security offering | Yes |
| OnlineBackup | Categorical | Customer has opted for online backup offering | Yes |
| DeviceProtection | Categorical | Customer has opted for device protection offering | Yes |
| TechSupport | Categorical | Customer has opted for technical support offering | Yes |
| PaperlessBilling | Categorical | Option to choose electronic statements | No |
| PaymentMethod | Categorical | Payment method chosen by customer | Electronic Check |

# C3. Steps for Analysis

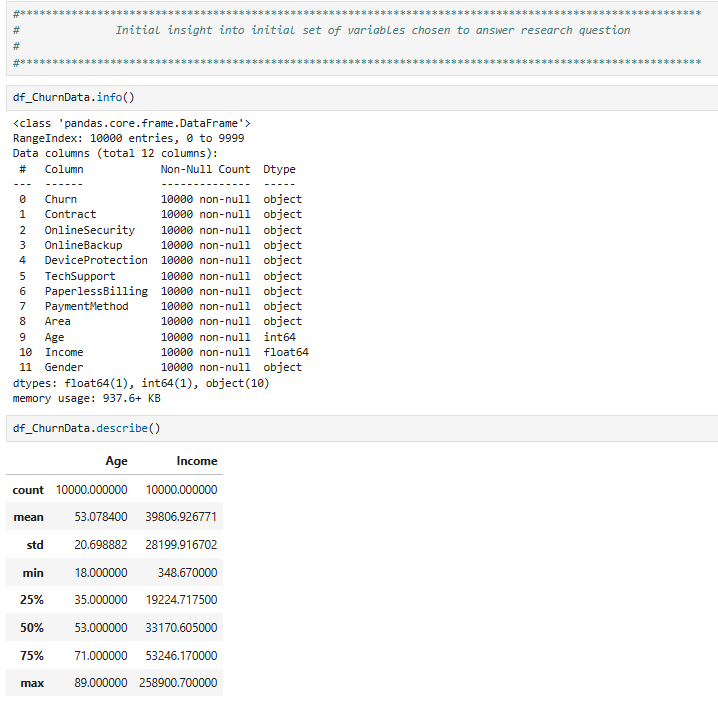
* Load Data

The data was loaded and the initial data set was selected based off of research the question.



* Preliminary Analysis

A preliminary analysis is performed in order to better understand the data structure and evaluate the data types.



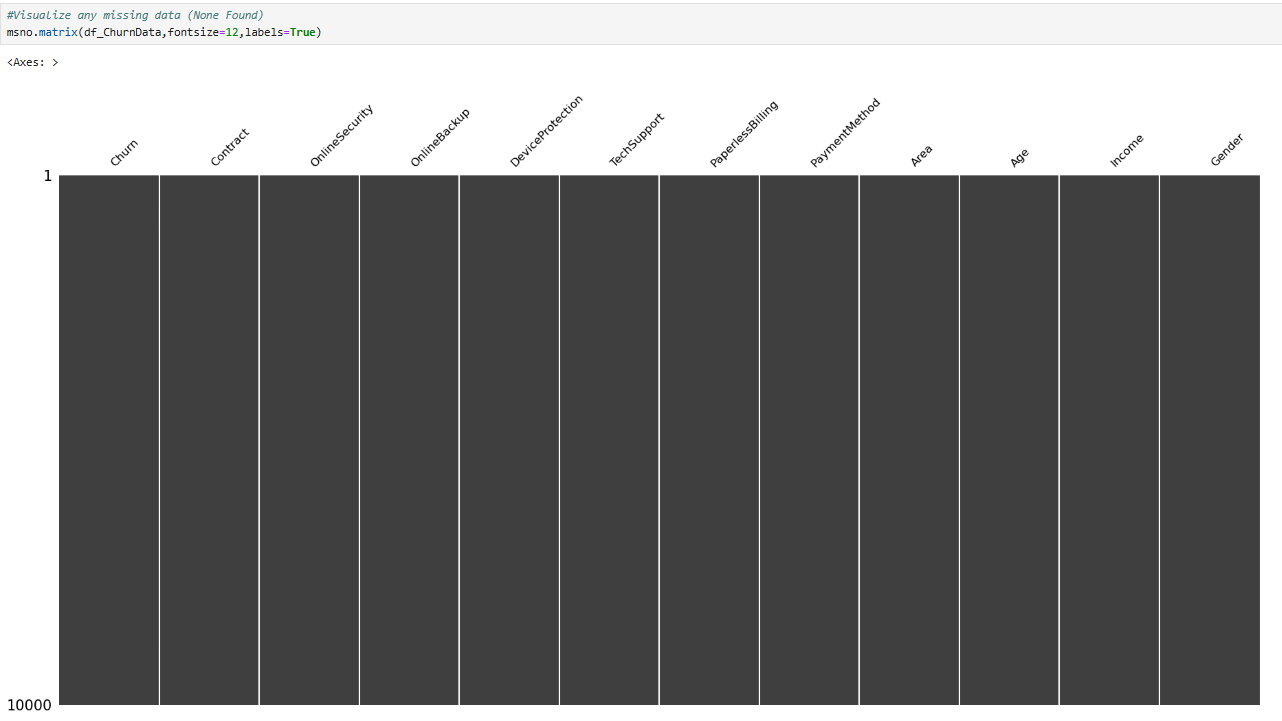
* Detect/Treat Duplicates, Missing Data, Data Validation and Outliers

Data cleaning is performed in order to both detect and treat duplicate data, missing data, errant data and outliers.

**Duplicates**

****

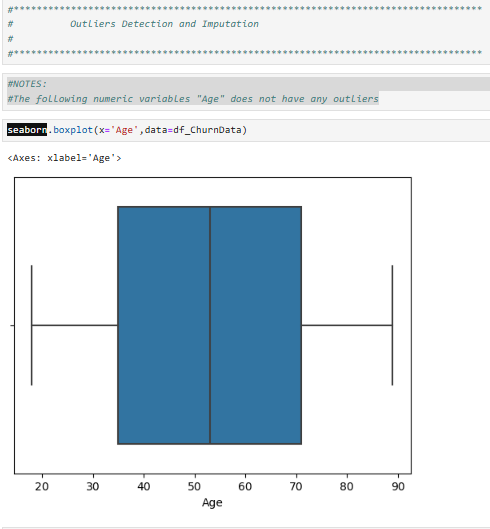
**Missing Data**

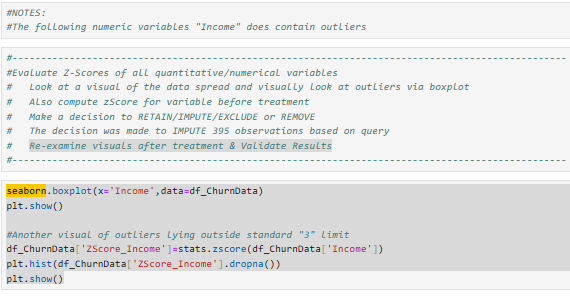
****

**Data Validation**

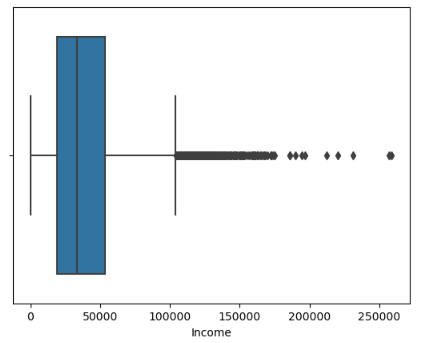
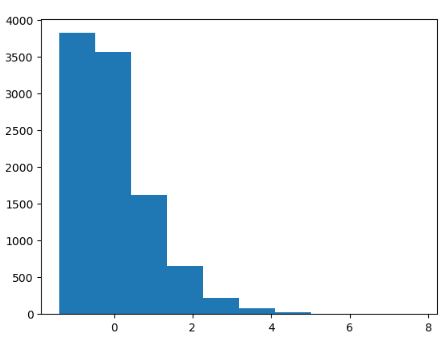
****

**Outliers**

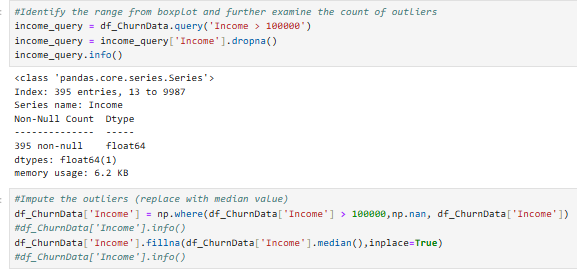
****

****

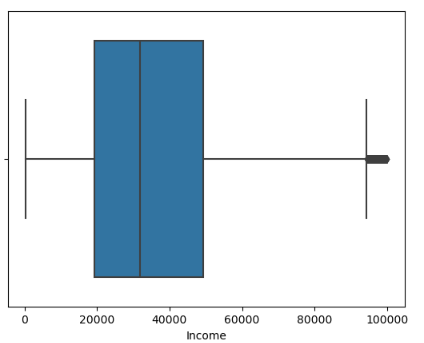
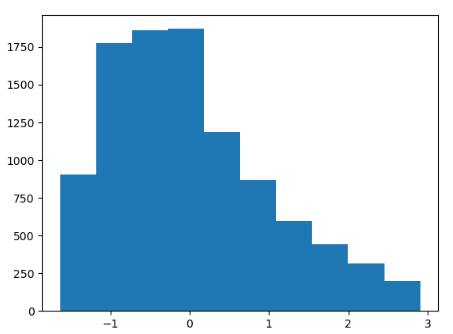
**BEFORE**

**ANALYSIS & TREATMENT**



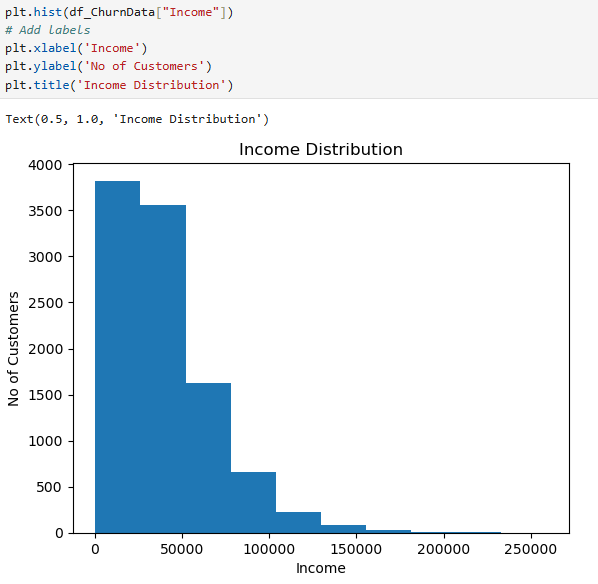
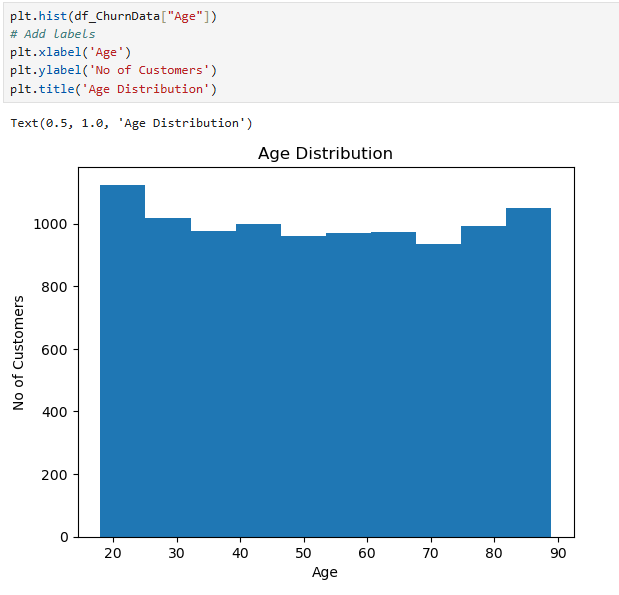
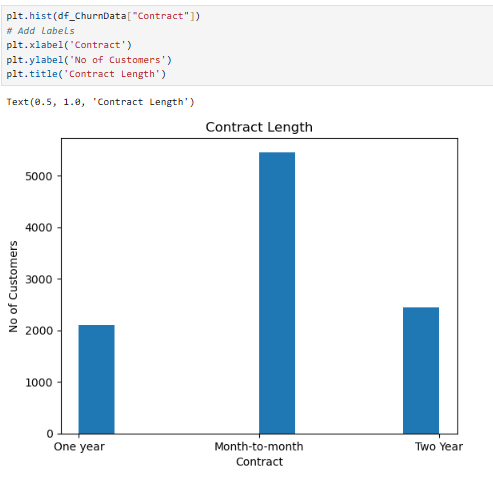
AFTER

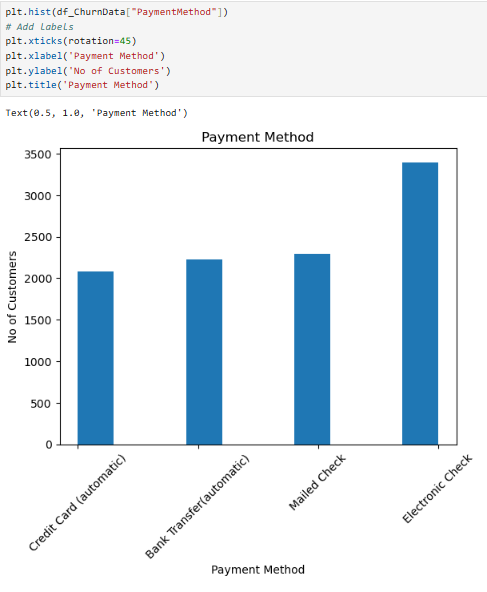
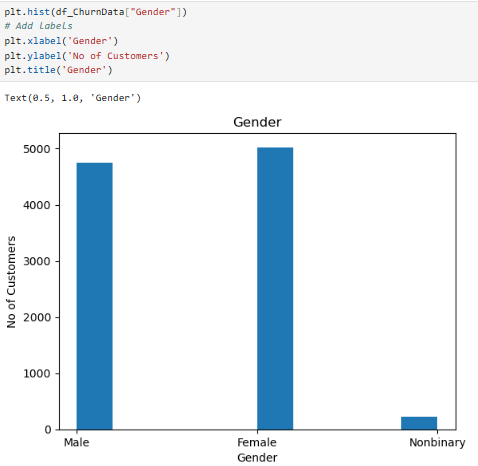
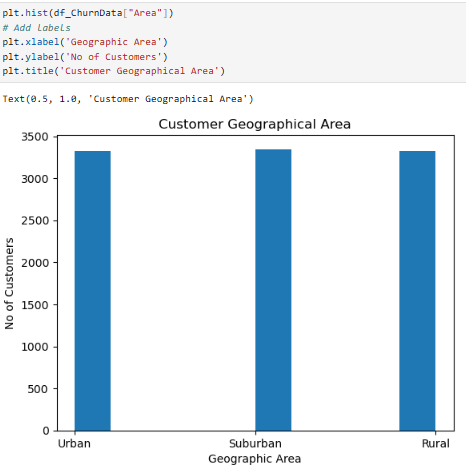
 

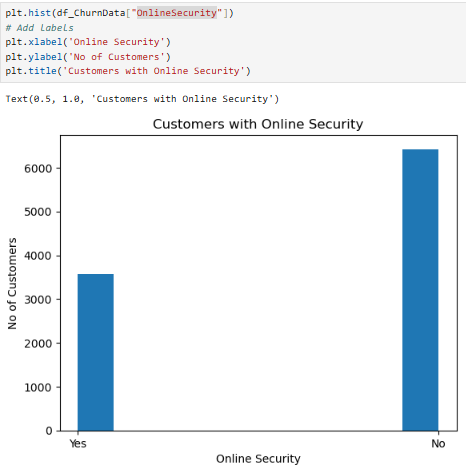
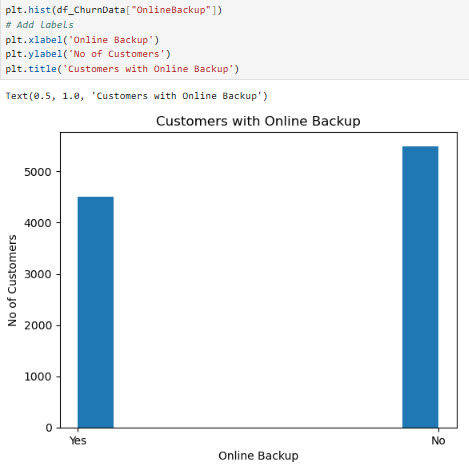
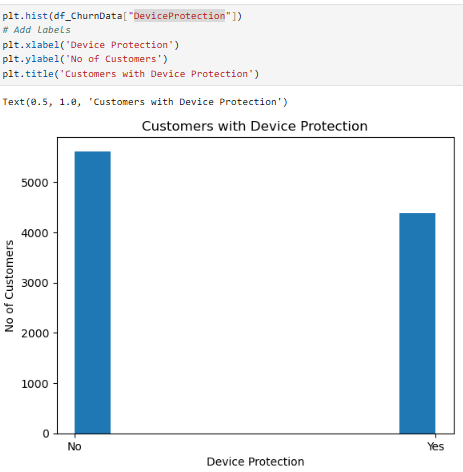
* EDA

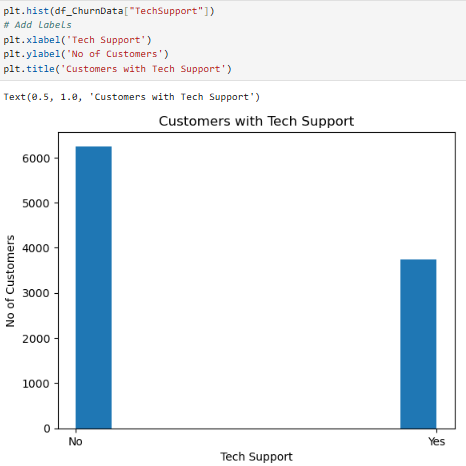
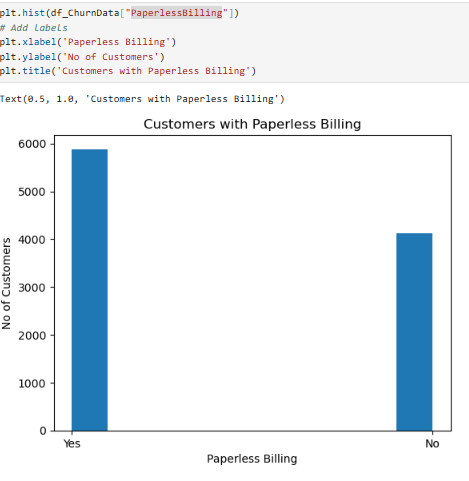
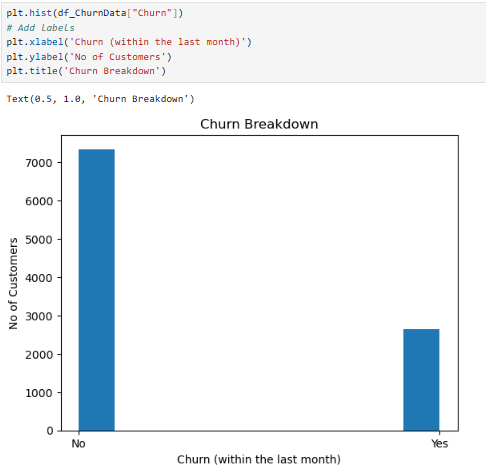
Exploratory analysis was performed on each of the remaining variables in order to get a better understanding of the data being considered for the model.

### Univariate Visualizations

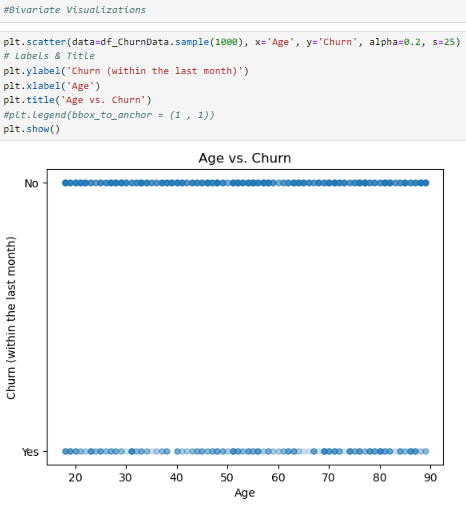
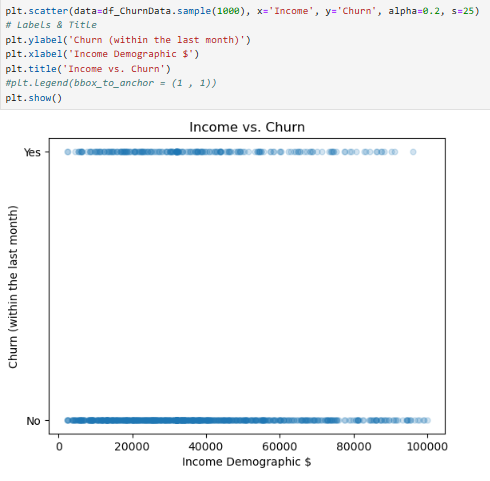
 

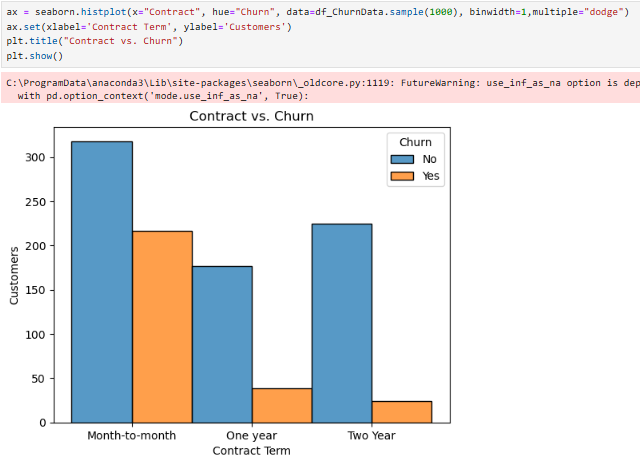
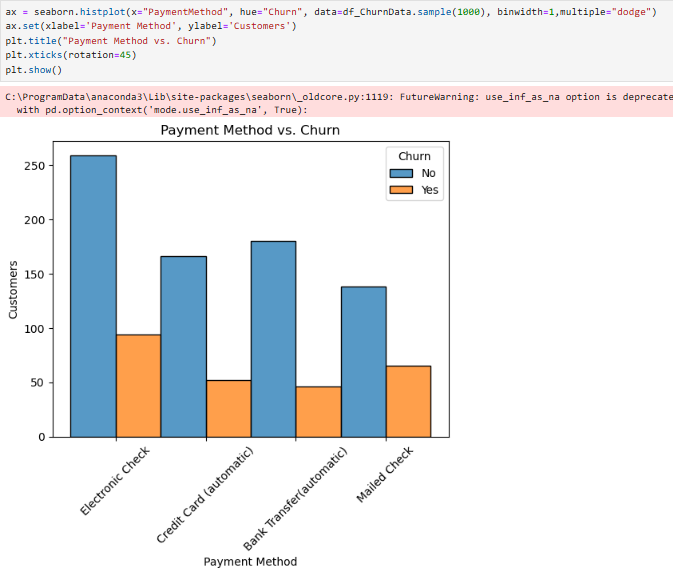
  

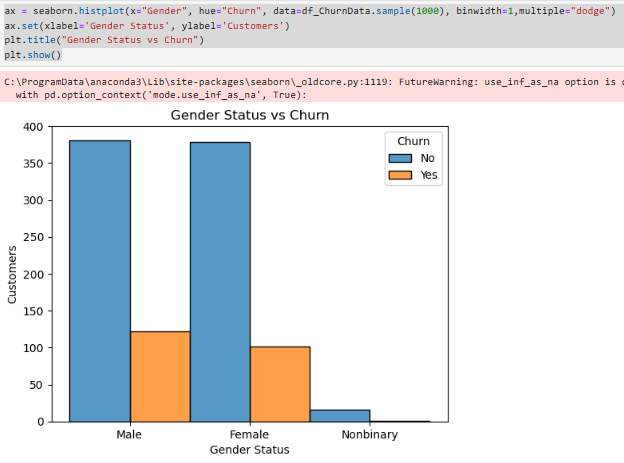
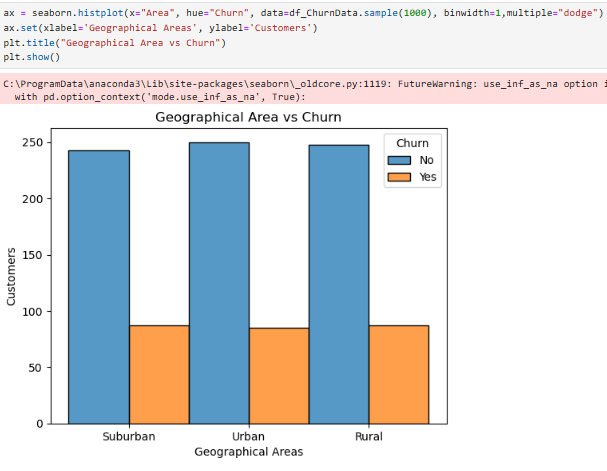
  

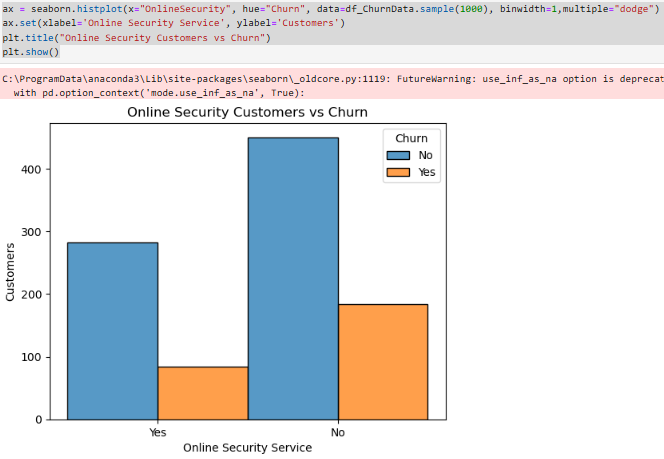
  

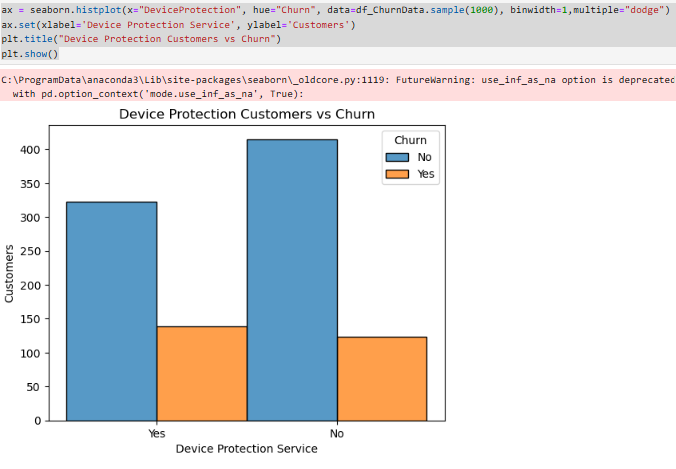
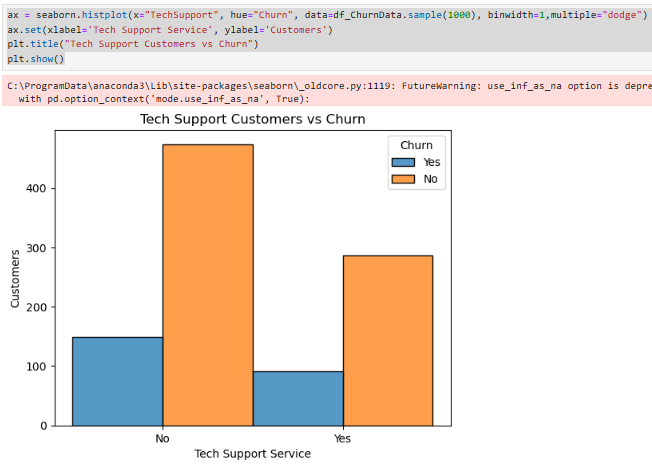
### Bivariate Visualizations

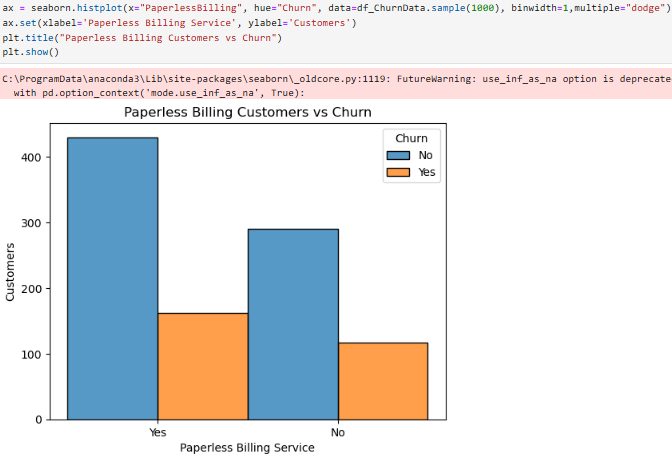
 



* Data Wrangling (Categorical Re-expression)

Machine learning work off of numerical data, so all categorical variables need to be encoded.

Ordinal Encoding



One-Hot Encoding



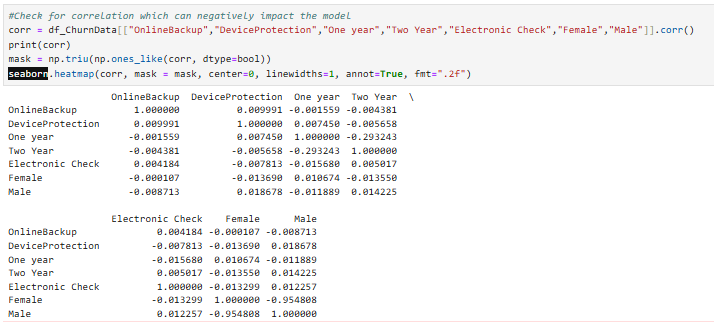
* Feature Selection

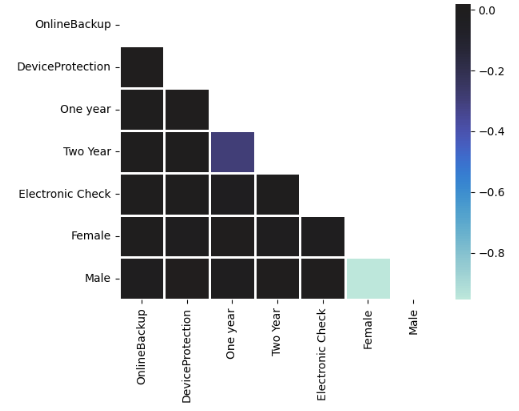
Feature selection for this model was chosen by using the SelectKBest method from Scikit-learn library. This algorithm provides an easy way to assess which features depict the greatest impact on our target variable. For this model, all variables showing a p-value less than 0.05 were chosen.

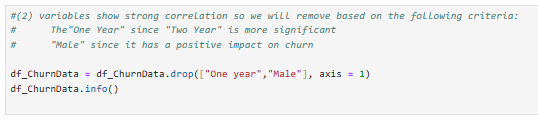


* Correlation Check

Machine learning models can be negatively impacted by strongly correlated predictors. Therefore, variables identified with strong correlation were removed. The variables chosen to be removed (“One Year” and “Male”) showed a lower p\_value than their counterpart.







* Columns Removed

In order to finalize the cleaned data set, unnecessary variables were removed and variables were renamed for clarity.



# C4. Cleaned Data Set

**See prepared data attached**

“Bret McHenry D209\_SectionC4\_CleanExtract.csv”

# D1. Splitting the Data



**See prepared data attached**

“Bret McHenry D209\_SectionD1\_X\_train.csv”

“Bret McHenry D209\_SectionD1\_X\_test.csv”

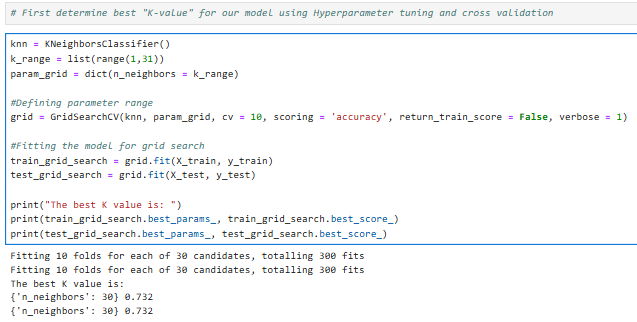
“Bret McHenry D209\_SectionD1\_y\_train.csv”

“Bret McHenry D209\_SectionD1\_y\_test.csv”

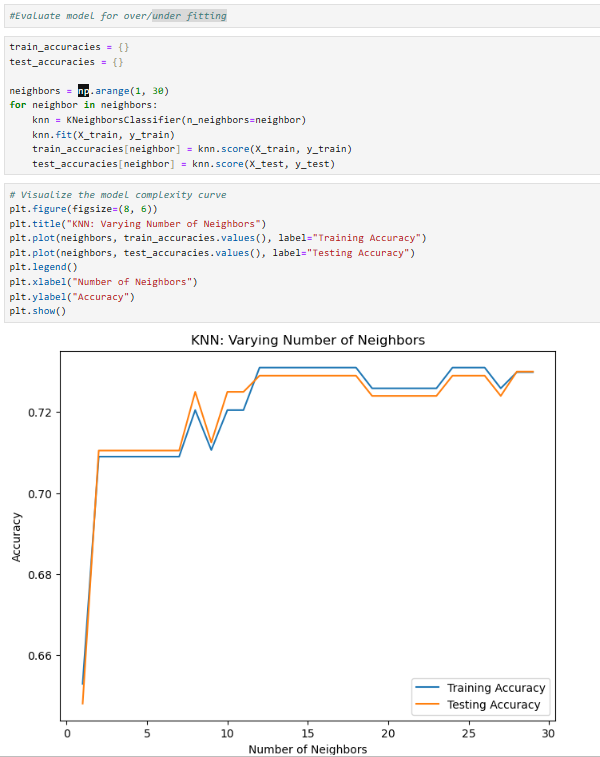
# D2. Output and Intermediate Calculations

After splitting the cleaned data into training and test sets in order to perform supervised learning, I was prepared to scale the data due to the sensitivity of the KNN model regarding distance. However, since the feature selection process contained no continuous variables, it was not necessary to standardize the data. The next step was to develop an optimal model based off of the features selected. The primary parameter in the KNN model is “k” representing the number of nearest neighbors to consider in order to perform classification of the target variable. I chose to utilize hyperparameter tuning along with cross validation in order to be confident that the model would provide the best possible accuracy. The algorithm determined that the best “k” value was 28, delivering a 73.2% accuracy score against both the training and test data. However, I then needed to run a model complexity curve in order to guard against under/over fitting. The visual provided below depicted that the optimal value for “k” neighbors should be 12 without compromising the model’s accuracy. The training model was then fitted with the derived “k” value of 12 and run against the test data set. The confusion matrix showed an accuracy of nearly 73% with an AUC score of 56.7%.

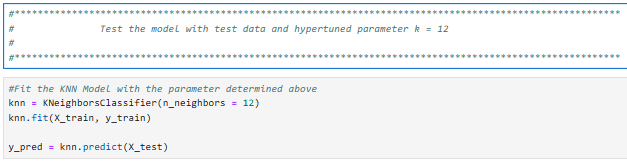
Hyperparameter Tuning

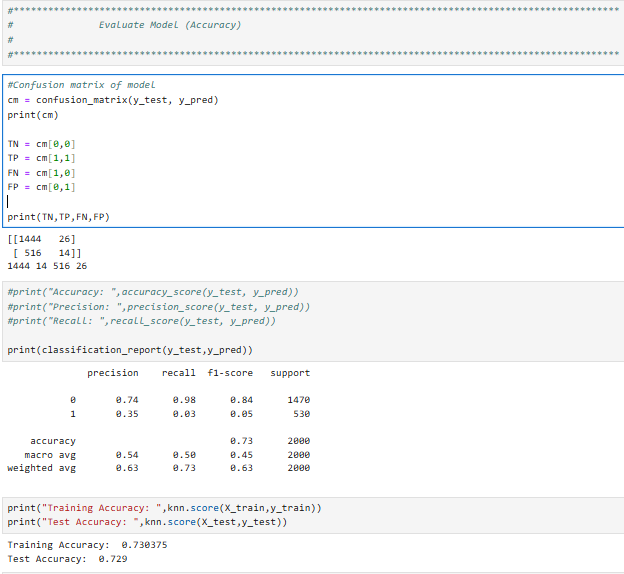


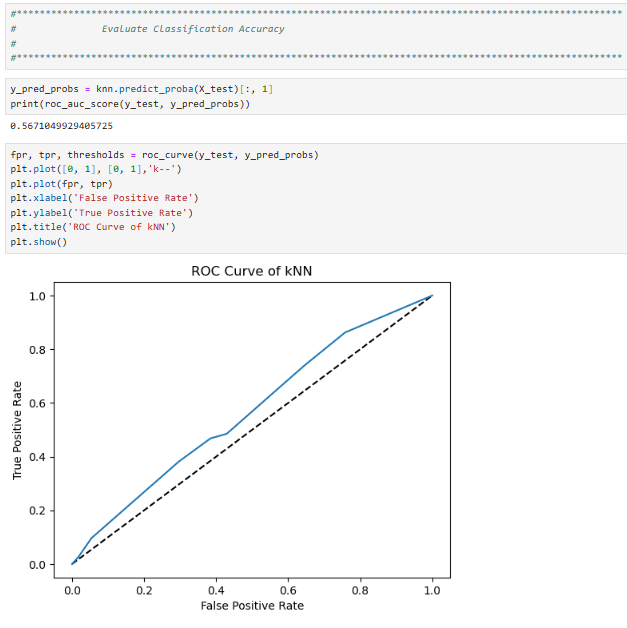
Model Complexity Curve Analysis



Results







# D3. Code Execution

**\*NOTE: Code attached as “Bret McHenry D209\_SectionD3.ipynb”.**

# E1. Accuracy and AUC

The accuracy of a model is determined by the number of correct predictions, both true and false divided by the total observations. The model in this exercise performed an accuracy of nearly 73%. The model was fit with training data (seen) and measured against test data (unseen) in order to provide the metric. The precision of the model focuses on the percentage of true positive values divided by true and false positive combined. In the model created, the precision was 35% and the recall rate (TP / (TP + FN)) was 2.6%.

The AUC is a metric to rate the performance of the classification model. It is a summary measure of how well the model performs against unseen data with 1 indicating a perfect model, 0 predicting all values incorrectly and .5 indicating complete randomness, meaning every classification prediction has a 50% chance of being correct. AUC stands for “Area Under Curve”, representing the 2-dimensional area under an ROC curve. An AUC score between .5 and 1 indicates that, as the metric moves closer to 1, the model has a higher chance to classify labels correctly. The AUC score from our model in this exercise is 0.5671049929405725.

# E2. Results and Implications

The accuracy of the model created depicts the challenge of the organization to target customers that “stick” with their organization and mitigate the associated costs. As noted above, the model demonstrated a 73% accuracy metric when run against the test data set. However, since accuracy is defined as the number of correct predictions divided by the number of samples, the model provides challenging results. Although the ability to predict true positives, known as precision, is only 35%, the more concerning metric is that there were 530 customers in the test data set that experienced churn and only 14 were classified correctly shown as the recall rate (less than 3%). In contrast, over 98% of the negative conditions were classified correctly with a 74% precision rate. Moreover, the research question is based off of the ability to target customers that are less likely to churn which would equate to the negative condition or churn=0. Therefore, I do believe the model provides valuable insight, but would need more analysis to provide more clarity. I believe the AUC score is consistent with the assessment of the accuracy metric. An AUC score of 56.7% is on the positive side of the ROC curve, but not much better than complete randomness or a score of 50%. However, the ROC curve is looking at things from the ability to discern effectiveness of the model via a true positive rate versus a false positive rate. Again, in this case, the true negative rate versus the false negative rate might be more meaningful since it would depict the model’s ability to predict customers that would not be as susceptible to churning and better support the research question.

# E3. Limitation

The biggest limitation I perceived in the creation of this model is understanding the importance/impact of each predictor on the model. During feature selection, I utilized the SelectKBest module from the sklearn package. I decided to utilize predictors based off of significance as declared by their corresponding p-value (< 0.05). The method was great, providing a ranking of features. However, I simply utilized the result of the method, unable to detect if the predictor had a positive or negative impact on the target variable. Therefore, I question if the significance of the chosen features could have offset each other and not been the optimal mix to produce a better model. As noted above, I would assert that further research is needed in order to more confidently address the research question.

# E4. Course of Action

Based on the model’s results, I would recommend further research in order to fine tune the model. I believe the churn data could be flipped so that 1 (a positive result) could equate to customers not churning and 0 (a negative result) could represent a customer experiencing churn. I believe the model evaluation metrics might provide more favorable results and provide confidence in the output especially since the ROC curve targets the effectiveness of the positive condition by graphing the “TPR” (true positive rate) versus the “FPR” (false positive rate). Also, I believe that more insight into the feature selection process could increase the model’s effectiveness.

# F. Panopto Recording

Below is a link to my code review.

# <https://wgu.hosted.panopto.com/Panopto/Pages/Viewer.aspx?id=5fe7eb74-83ad-4e2c-b389-b20b00fb65cb>

# G. Sources for Third-Party Code

N/A

# H. Sources

N/A

**References**

D209 Data Mining 1 Task 1 Cohort.pptx (2024)

Assumption of KNN (Slide #23)

D209 Data Mining 1 Task 1 Cohort.pptx (2024)

SelectKBest Code in Python (Slide #32)

D209 Data Mining 1 Task 1 Cohort.pptx (2024)

Building the kNN Model in Python (hyperparameter tuning) (Slides #47, #48)