D209 Data Mining I Performance Task 1

Sean Simmons

WDU Data Analytics

MSDA D209

February 2023

Part I: Research Question

A. Describe the purpose of this data mining report by doing the following:

1. Propose one question relevant to a real-world organizational situation that you will answer using one of the following classification methods:

- ***k*-nearest neighbor (KNN)**
- **Naive Bayes**

2. Define one goal of the data analysis. Ensure that your goal is reasonable within the scope of the scenario and is represented in the available data.

1. Can we use independent predictor variables to predict if a customer will Churn? Churn is defined as leaving the organization within the last month and is a categorical Yes or No variable. The K-nearest neighbor (KNN) classification method will be used to answer this question.
2. One goal of this analysis is to determine if a customer will churn using independent variables from the dataset and prepare to find which variables, or how many, are the most important in determining this. This goal is reasonable within our project because our classification method, KNN, can use the dataset to predict a categorical variable of interest for the organization. Churn is a binary categorical variable and we can use our other independent variables as predictors. Knowing what characteristics could predict if a future customer will Churn can aid the organization in their retention efforts.

Part II: Method Justification

B. Explain the reasons for your chosen classification method from part A1 by doing the following:

1. **Explain how the classification method you chose analyzes the selected data set. Include expected outcomes.**
2. **Summarize one assumption of the chosen classification method.**
3. **List the packages or libraries you have chosen for Python or R, and justify how *each* item on the list supports the analysis.**

1. KNN is a nonlinear supervised machine learning algorithm that splits datasets into training and test sets for classification and regression testing. For categorical target variables, seen here with Churn, KNN uses classification for prediction. More specifically, KNN will use churn and the other variables in the data set to perform a classification prediction on Churn based on the idea that you can use the distance between data points to predict the outcome of the variable. I chose to initially analyze the nearest seven data points for increased accuracy. The expected outcome is whether the next nearest data point will be truthful and to what reliability and accuracy that prediction can be made with (*K Nearest Neighbors with Python* 2019).
2. One assumption of KNN is that data points that are close in proximity to each other are very similar, also known as feature space and this space/distance can be measured mathematically and used to classify data points (Vishalmendekarhere 2021).
3. The packages and their justification are as follows (Also found annotated in the code file provided):

a. Pandas - To load my dataset into my coding environment and perform basic manipulations with the dataframe.
b. Numpy - To perform mathematical operations with the dataset.
c. Matplotlib, including subpackages - To perform variable analysis and provide visualizations for the models.
d. Sklearn, including subpackages - To perform KNN classification and modeling. This has subpackages that allow the data to be split into training and testing sets and then fit to the KNN model and tested for accuracy. Essentially, these packages performed the entirety of the data modeling and analysis.
e. Seaborn - To visualize the heatmap. To perform univariate and bivariate analysis.

Part III: Data Preparation

C. Perform data preparation for the chosen data set by doing the following:

1. Describe one data preprocessing goal relevant to the classification method from part A1.

2. Identify the initial data set variables that you will use to perform the analysis for the classification question from part A1, and classify *each* variable as continuous or categorical.

3. Explain *each* of the steps used to prepare the data for the analysis. Identify the code segment for *each* step.

4. Provide a copy of the cleaned data set.

1. One data preprocessing goal relevant to the KNN is to convert our categorical variables to numeric in order for them to be included in the analysis. The code for these can be seen in the file provided and changes all of the categorical values to original name_numeric with 0 and 1 for Yes/No and increasing counts for variables with more than 2 different values.
2. The initial data set variables we will use to perform the analysis of Churn are as follows: Our target, y, value is Churn. The other variables in the dataset are our feature variables, found below.

Variable Name	**Data Type**
Churn - Target	Categorical
Outage_Sec_perweek	Continuous
Contract	Continuous
Tenure	Continuous
MonthlyCharge	Continuous
Bandwidth_GB_year	Continuous
Email	Continuous
Yearly_equip_failure	Continuous
Contacts	Continuous
Children	Continuous
Age	Continuous
Income	Continuous
Gender	Categorical
DeviceProtection	Categorical
Phone	Categorical

Multiple	Categorical
OnlineSecurity	Categorical
OnlineBackup	Categorical
TechSupport	Categorical
StreamingTV	Categorical
StreamingMovies	Categorical
Techie	Categorical
Port_modem	Categorical
Tablet	Categorical
InternetService	Categorical
Paperless Billing	Categorical
Item 1	Categorical- Discrete Ordinal
Item 2	Categorical- Discrete Ordinal
Item 3	Categorical- Discrete Ordinal
Item 4	Categorical- Discrete Ordinal
Item 5	Categorical- Discrete Ordinal
Item 6	Categorical- Discrete Ordinal
Item 7	Categorical- Discrete Ordinal
Item 8	Categorical- Discrete Ordinal

```
Floats
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10000 entries, 0 to 9999
Data columns (total 7 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   Lat                 10000 non-null  float64
 1   Lng                 10000 non-null  float64
 2   Income              10000 non-null  float64
 3   Outage_sec_perweek  10000 non-null  float64
 4   Tenure              10000 non-null  float64
 5   MonthlyCharge       10000 non-null  float64
 6   Bandwidth_GB_Year   10000 non-null  float64
dtypes: float64(7)
memory usage: 547.0 KB
None
Integers
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10000 entries, 0 to 9999
Data columns (total 16 columns):
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   CaseOrder             10000 non-null  int64
 1   Zip                   10000 non-null  int64
 2   Population            10000 non-null  int64
 3   Children              10000 non-null  int64
 4   Age                   10000 non-null  int64
 5   Email                 10000 non-null  int64
 6   Contacts              10000 non-null  int64
 7   Yearly_equip_failure  10000 non-null  int64
 8   Item1                 10000 non-null  int64
 9   Item2                 10000 non-null  int64
 10  Item3                 10000 non-null  int64
 11  Item4                 10000 non-null  int64
 12  Item5                 10000 non-null  int64
 13  Item6                 10000 non-null  int64
 14  Item7                 10000 non-null  int64
 15  Item8                 10000 non-null  int64
dtypes: int64(16)
memory usage: 1.2 MB
None
```

```
Objects
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10000 entries, 0 to 9999
Data columns (total 27 columns):
 #   Column            Non-Null Count  Dtype 
---  ------            --------------  ----- 
 0   Customer_id       10000 non-null  object
 1   Interaction       10000 non-null  object
 2   UID               10000 non-null  object
 3   City              10000 non-null  object
 4   State             10000 non-null  object
 5   County            10000 non-null  object
 6   Area              10000 non-null  object
 7   TimeZone          10000 non-null  object
 8   Job               10000 non-null  object
 9   Marital           10000 non-null  object
 10  Gender            10000 non-null  object
 11  Churn             10000 non-null  object
 12  Techie            10000 non-null  object
```

PA_D209_C

```
 13  Contract          10000 non-null  object
 14  Port_modem        10000 non-null  object
 15  Tablet            10000 non-null  object
 16  InternetService   10000 non-null  object
 17  Phone             10000 non-null  object
 18  Multiple          10000 non-null  object
 19  OnlineSecurity    10000 non-null  object
 20  OnlineBackup      10000 non-null  object
 21  DeviceProtection  10000 non-null  object
 22  TechSupport       10000 non-null  object
 23  StreamingTV       10000 non-null  object
 24  StreamingMovies   10000 non-null  object
 25  PaperlessBilling  10000 non-null  object
 26  PaymentMethod     10000 non-null  object
dtypes: object(27)
memory usage: 2.1+ MB
None
```

```
Dataset Information
<bound method DataFrame.info of       CaseOrder Customer_id
Interaction  \
0             1     K409198  aa90260b-4141-4a24-8e36-b04ce1f4f77b
1             2     S120509  fb76459f-c047-4a9d-8af9-e0f7d4ac2524
2             3     K191035  344d114c-3736-4be5-98f7-c72c281e2d35
3             4      D90850  abfa2b40-2d43-4994-b15a-989b8c79e311
4             5     K662701  68a861fd-0d20-4e51-a587-8a90407ee574
...         ...         ...                                   ...
9995       9996     M324793  45deb5a2-ae04-4518-bf0b-c82db8dbe4a4
9996       9997     D861732  6e96b921-0c09-4993-bbda-a1ac6411061a
9997       9998     I243405  e8307ddf-9a01-4fff-bc59-4742e03fd24f
9998       9999     I641617  3775ccfc-0052-4107-81ae-9657f81ecdf3
9999      10000      T38070  9de5fb6e-bd33-4995-aec8-f01d0172a499

                                   UID          City State  \
0     e885b299883d4f9fb18e39c75155d990   Point Baker    AK
1     f2de8bef964785f41a2959829830fb8a  West Branch    MI
2     f1784cfa9f6d92ae816197eb175d3c71       Yamhill    OR
3     dc8a365077241bb5cd5ccd305136b05e       Del Mar    CA
4     aabb64a116e83fdc4befc1fbab1663f9     Needville    TX
...                                ...           ...   ...
9995  9499fb4de537af195d16d046b79fd20a   Mount Holly    VT
9996  c09a841117fa81b5c8e19afec2760104   Clarksville    TN
9997  9c41f212d1e04dca84445019bbc9b41c      Mobeetie    TX
9998  3e1f269b40c235a1038863ecf6b7a0df    Carrollton    GA
9999  0ea683a03a3cd544aefe8388aab16176  Clarkesville    GA

                      County    Zip       Lat        Lng  ...  MonthlyCharge  \
0     Prince of Wales-Hyder  99927  56.25100 -133.37571  ...     172.455519
1                    Ogemaw  48661  44.32893  -84.24080  ...     242.632554
2                   Yamhill  97148  45.35589 -123.24657  ...     159.947583
3                 San Diego  92014  32.96687 -117.24798  ...     119.956840
4                 Fort Bend  77461  29.38012  -95.80673  ...     149.948316
...                     ...    ...       ...        ...  ...            ...
9995                Rutland   5758  43.43391  -72.78734  ...     159.979400
9996             Montgomery  37042  36.56907  -87.41694  ...     207.481100
9997                Wheeler  79061  35.52039 -100.44180  ...     169.974100
9998                Carroll  30117  33.58016  -85.13241  ...     252.624000
9999              Habersham  30523  34.70783  -83.53648  ...     217.484000

     Bandwidth_GB_Year Item1 Item2  Item3  Item4  Item5 Item6 Item7 Item8
0           904.536110     5     5      5      3      4     4     3     4
```

1	800.982766	3	4	3	3	4	3	4	4
2	2054.706961	4	4	2	4	4	3	3	3
3	2164.579412	4	4	4	2	5	4	3	3
4	271.493436	4	4	4	3	4	4	4	5
...	...	...	...	...	...	...	...	...	...
9995	6511.252601	3	2	3	3	4	3	2	3
9996	5695.951810	4	5	5	4	4	5	2	5
9997	4159.305799	4	4	4	4	4	4	4	5
9998	6468.456752	4	4	6	4	3	3	5	4
9999	5857.586167	2	2	3	3	3	3	4	1

Missing Values:

```
CaseOrder                0
Customer_id              0
Interaction              0
UID                      0
City                     0
State                    0
County                   0
Zip                      0
Lat                      0
Lng                      0
Population               0
Area                     0
TimeZone                 0
Job                      0
Children                 0
Age                      0
Income                   0
Marital                  0
Gender                   0
Churn                    0
Outage_sec_perweek       0
Email                    0
Contacts                 0
Yearly_equip_failure     0
Techie                   0
Contract                 0
Port_modem               0
Tablet                   0
InternetService          0
Phone                    0
Multiple                 0
OnlineSecurity           0
OnlineBackup             0
DeviceProtection         0
TechSupport              0
StreamingTV              0
StreamingMovies          0
PaperlessBilling         0
PaymentMethod            0
Tenure                   0
MonthlyCharge            0
Bandwidth_GB_Year        0
item1_responses          0
item2_fixes              0
item3_replacements       0
item4_reliability        0
item5_options            0
item6_respectfulness     0
item7_courteous          0
item8_listening          0
dtype: int64
```

Mean and then Median below:

```
CaseOrder                      5000.500000
Zip                           49153.319600
Lat                              38.757567
Lng                             -90.782536
Population                     9756.562400
Children                          2.087700
Age                              53.078400
Income                        39806.926771
Outage_sec_perweek               10.001848
Email                            12.016000
Contacts                          0.994200
Yearly_equip_failure              0.398000
Tenure                           34.526188
MonthlyCharge                   172.624816
Bandwidth_GB_Year              3392.341550
item1_responses                   3.490800
item2_fixes                       3.505100
item3_replacements                3.487000
item4_reliability                 3.497500
item5_options                     3.492900
item6_respectfulness              3.497300
item7_courteous                   3.509500
item8_listening                   3.495600
Churn_numeric                     0.735000
Area_numeric                      1.000000
Marital_numeric                   2.017500
Gender_numeric                    0.571800
Contract_numeric                  1.034000
PaymentMethod_numeric             1.700300
InternetService_numeric           0.772100
Techie_numeric                    0.832100
Port_modem_numeric                0.516600
Tablet_numeric                    0.700900
Phone_numeric                     0.093300
Multiple_numeric                  0.539200
OnlineSecurity_numeric            0.642400
OnlineBackup_numeric              0.549400
DeviceProtection_numeric          0.561400
TechSupport_numeric               0.625000
StreamingTV_numeric               0.507100
StreamingMovies_numeric           0.511000
PaperlessBilling_numeric          0.411800
dtype: float64
```

```
CaseOrder                        5000.500000
Zip                             48869.500000
Lat                                39.395800
Lng                               -87.918800
Population                       2910.500000
Children                            1.000000
Age                                53.000000
Income                          33170.605000
Outage_sec_perweek                 10.018560
Email                              12.000000
Contacts                            1.000000
Yearly_equip_failure                0.000000
Tenure                             35.430507
MonthlyCharge                     167.484700
Bandwidth_GB_Year                3279.536903
item1_responses                     3.000000
item2_fixes                         4.000000
```

```
                                        PA_[
item3_replacements                  3.000000
item4_reliability                   3.000000
item5_options                       3.000000
item6_respectfulness                3.000000
item7_courteous                     4.000000
item8_listening                     3.000000
Churn_numeric                       1.000000
Area_numeric                        1.000000
Marital_numeric                     2.000000
Gender_numeric                      1.000000
Contract_numeric                    1.000000
PaymentMethod_numeric               2.000000
InternetService_numeric             1.000000
Techie_numeric                      1.000000
Port_modem_numeric                  1.000000
Tablet_numeric                      1.000000
Phone_numeric                       0.000000
Multiple_numeric                    1.000000
OnlineSecurity_numeric              1.000000
OnlineBackup_numeric                1.000000
DeviceProtection_numeric            1.000000
TechSupport_numeric                 1.000000
StreamingTV_numeric                 1.000000
StreamingMovies_numeric             1.000000
PaperlessBilling_numeric            0.000000
dtype: float64
```

3. Steps of the data preparation and code are written and annotated in "D209_Task1_Code.ipynb", I have added the steps here just for summary (letter a for example will be in the code as a comment "#a").
    a. Import the data into my coding environment
    b. View the data type and summary information to prepare for the modeling
    c. Rename nondescript columns of items
    d. Check for missing values and mitigate if there are any
    e. Change categorical values into numeric counterparts (except for the discrete ordinal survey items columns)
    f. Check summary statistics, such as mean and median
    g. Drop unnecessary columns that will not be included in our model, as described earlier in the report (mainly demographic and unique customer id information).
    h. Extract data set to csv file
    i. Perform univariate and bivariate analysis of target/feature variables
    j. List features that will be used in the dataset (all remaining features)
    k. Set target (churn) variables and predictor (all other) variables. Here our data preparation ends and we begin our testing and modeling, which are also labeled for convenience.
    l. Split data set into training (70%) and test (30%) sets
    m. Fit data to KNN model and perform a prediction using comprehensive sklearn packages to fit the data, scale it, and test it.

n. Test for accuracy, classification matrix, and confusion matrix using sklearn packages

o. Perform final prediction and gather results from the model creation (AUC).

4. The copy of the cleaned data set “D209_Task1_clean.csv”

Part IV: Analysis

D. Perform the data analysis and report on the results by doing the following:

1. Split the data into training and test data sets and provide the file(s).

2. Describe the analysis technique you used to appropriately analyze the data. Include screenshots of the intermediate calculations you performed.

3. Provide the code used to perform the classification analysis from part D2.

1. The training set is found in “Task1_X_Train.csv”, “Task1_Y_Train”, “Task1_X_test.csv”, “Task1_Y_test.csv”and “Task1_Y_Predict.csv.” The data is split into 70% for training and 30% for testing.
2. The analysis technique I used to analyze the data is a combination of gathering the KNN score, accuracy, and classification/confusion matrix. We score the model and test for accuracy with our chosen seven n_neigbors. Next, we split the data set into our training and test sets, perform a classification report, and perform a new test and accuracy score. A confusion matrix is used to analyze our data after we scale it to improve the model and

we use these methods to test for the true and false positives/negatives to confirm our accuracy score (see below). Finally, we can test the data again and gather a new accuracy score. As the score improves, we can see the utility of this model and can create a heatmap of the percentage of true and false positives and negatives. The confusion matrix accuracy results support our model's viability, all of the mentioned results can be seen in the outputs below and described next in this report.

As we have analyzed our model, it is found that 31 is actually the best number of neighbors and the area under the curve (AUC) score must be computed to test the model. If the AUC is below .5, it indicates the prediction is worse than a random prediction. An AUC score above .5 indicates our prediction method is performing better than a random prediction. Our accuracy scores range from .726 and go up to .80, indicating a final 80.1% accuracy and our AUC score is .8062, which is above .50, indicating our model of prediction is better at predicting than a random guess (This is supported by the 5 fold analysis as well). The area under the curve score is visualized and the percentages are shown in a heatmap to visualize our true and false positives/negatives probability. The best parameters were found to be 31 with a best KNN score of 0.735.

a. Intermediate calculations output/screenshots:

```
Features for analysis include:
 ['Children', 'Age', 'Income', 'Outage_sec_perweek', 'Email', 'Contacts', 'Yearly_equ
ip_failure', 'Tenure', 'MonthlyCharge', 'Bandwidth_GB_Year', 'item1_responses', 'item
2_fixes', 'item3_replacements', 'item4_reliability', 'item5_options', 'item6_respectf
ulness', 'item7_courteous', 'item8_listening', 'Churn_numeric', 'Area_numeric', 'Mari
tal_numeric', 'Gender_numeric', 'Contract_numeric', 'PaymentMethod_numeric', 'Interne
tService_numeric', 'Techie_numeric', 'Port_modem_numeric', 'Tablet_numeric', 'Phone_n
umeric', 'Multiple_numeric', 'OnlineSecurity_numeric', 'OnlineBackup_numeric', 'Devic
eProtection_numeric', 'TechSupport_numeric', 'StreamingTV_numeric', 'StreamingMovies_
numeric']
```

```
KNeighborsClassifier(n_neighbors=7)
```

```
#m: Print initial accuracy score of KNN model
print('KNN Initial Accuracy: ', accuracy_score(y_test, y_pred))
```

```
KNN Initial Accuracy:  0.7263333333333334
```

```
#m: Compute classification metrics
print(classification_report(y_test, y_pred))
```

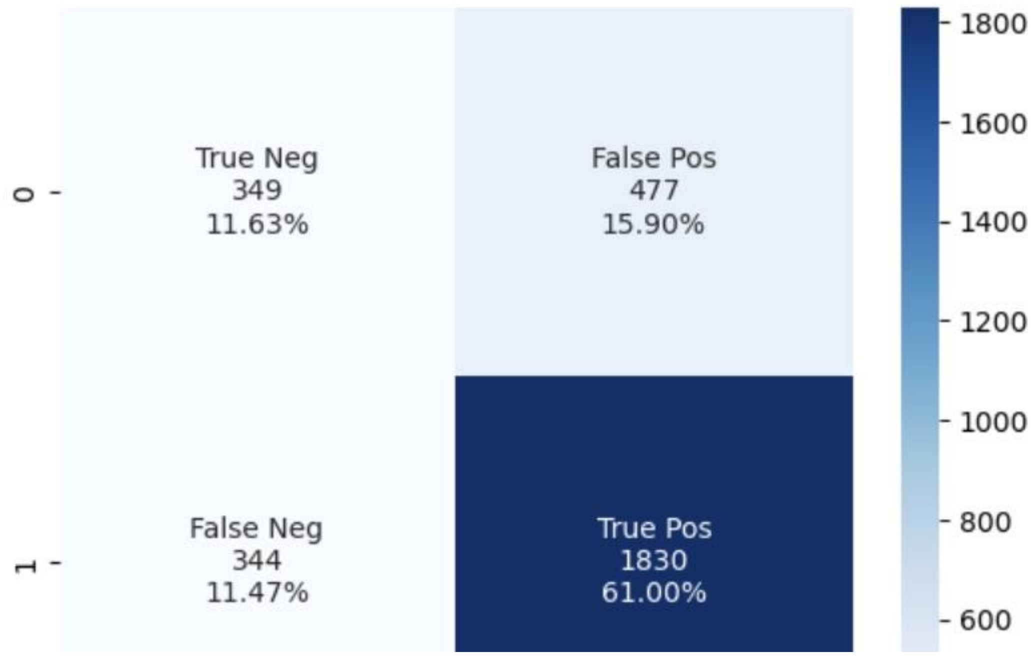
```
              precision    recall  f1-score   support

           0       0.50      0.42      0.46       826
           1       0.79      0.84      0.82      2174

    accuracy                           0.73      3000
   macro avg       0.65      0.63      0.64      3000
weighted avg       0.71      0.73      0.72      3000
```

```
#n: Print new accuracy score of scaled KNN model
print('KNN model New accuracy (scaled): {:0.3f}'.format(accuracy_score(y_test_scaled, y_pred_scaled)))
```

```
KNN model New accuracy (scaled): 0.801
```

```
#n: Compute classification metrics after scaling
print(classification_report(y_test_scaled, y_pred_scaled))

#Confusion_matrix & generate results
cf_matrix = confusion_matrix(y_test, y_pred)
print(cf_matrix)

# Visual confusion matrix
group_names = ['True Neg', 'False Pos', 'False Neg', 'True Pos']
group_counts = ["{0:0.0f}".format(value) for value in cf_matrix.flatten()]
group_percentages = ["{0:.2%}".format(value) for value in cf_matrix.flatten()/np.sum(cf_matrix)]
labels = [f"{v1}\n{v2}\n{v3}" for v1, v2, v3 in zip(group_names,group_counts,group_percentages)]
labels = np.asarray(labels).reshape(2,2)
sb.heatmap(cf_matrix, annot=labels, fmt='', cmap='Blues')
```

```
              precision    recall  f1-score   support

           0       0.66      0.60      0.63       558
           1       0.85      0.88      0.86      1442

    accuracy                           0.80      2000
   macro avg       0.75      0.74      0.75      2000
weighted avg       0.80      0.80      0.80      2000

[[ 349  477]
 [ 344 1830]]
<AxesSubplot:>
```




```python
#n: Set up parameters grid
param_grid = {'n_neighbors': np.arange(1, 50)}

# Re-initializing KNN for cross validation
knn = KNeighborsClassifier()
# Initializing GridSearch cross validation
knn_cv = GridSearchCV(knn , param_grid, cv=5)
# Fit model to
knn_cv.fit(X_train, y_train)

# Print best parameters
print('Best parameters for this KNN model: {}'.format(knn_cv.best_params_))
```

```
Best parameters for this KNN model: {'n_neighbors': 31}
```

```python
#n: Generate model best score
print('Best score for this KNN model: {:.3f}'.format(knn_cv.best_score_))
```

```
Best score for this KNN model: 0.735
```

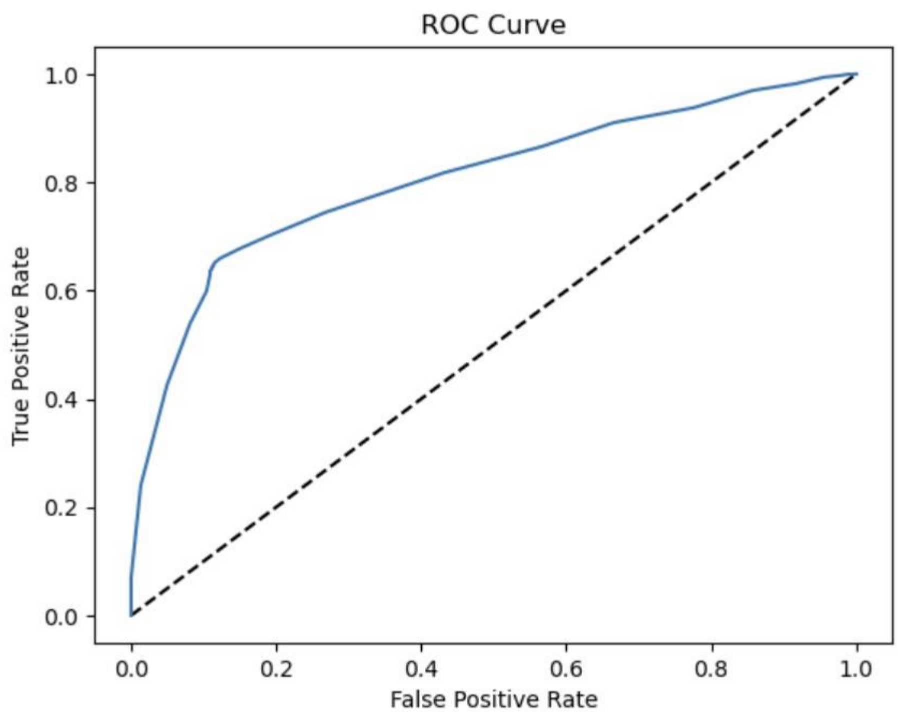
```
#n: Compute and print AUC score
print("The Area under curve (AUC) on validation dataset is: {:.4f}".format(roc_auc_score(y_test, y_pred_prob)))
```

```
The Area under curve (AUC) on validation dataset is: 0.8062
```

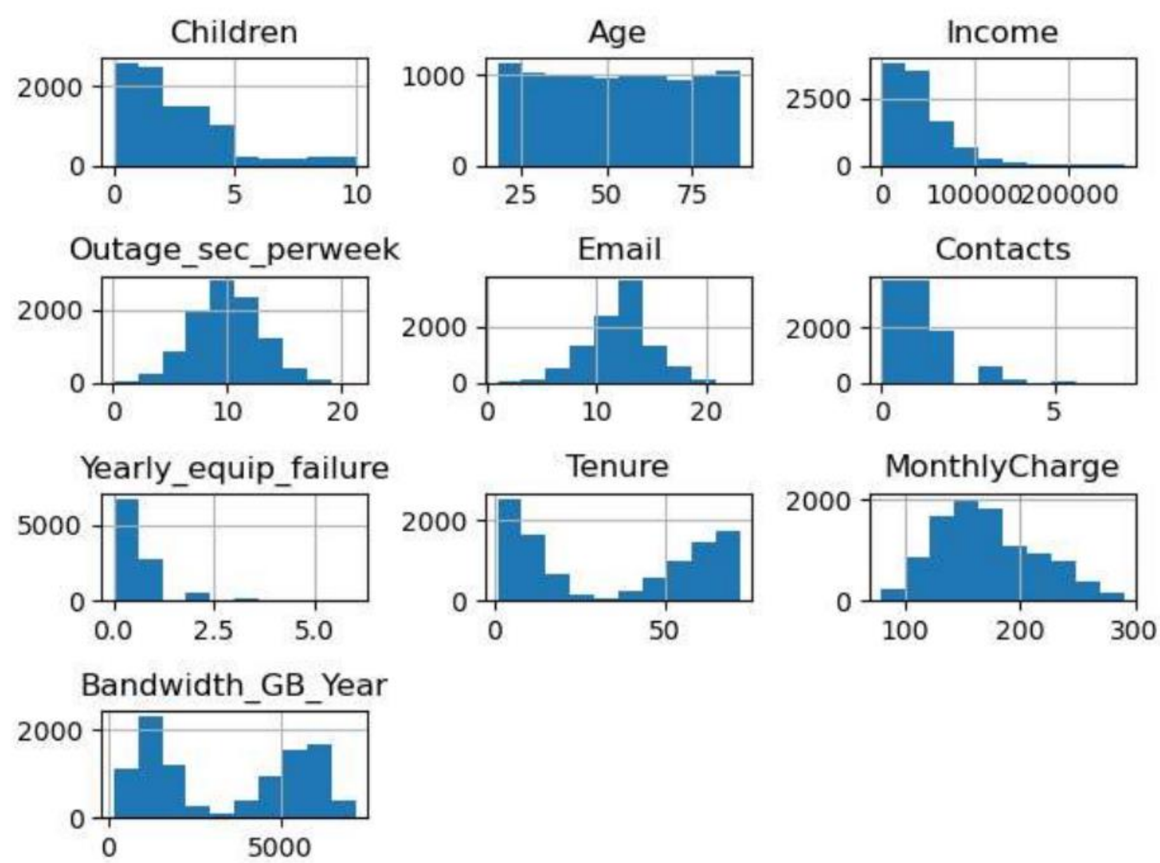
```
#n:Print list of AUC scores
print("AUC scores computed using 5-fold cross-validation: {}".format(cv_auc))
```

```
AUC scores computed using 5-fold cross-validation: [0.68222821 0.17760236 0.96643691 0.98773457 0.58834745]
```

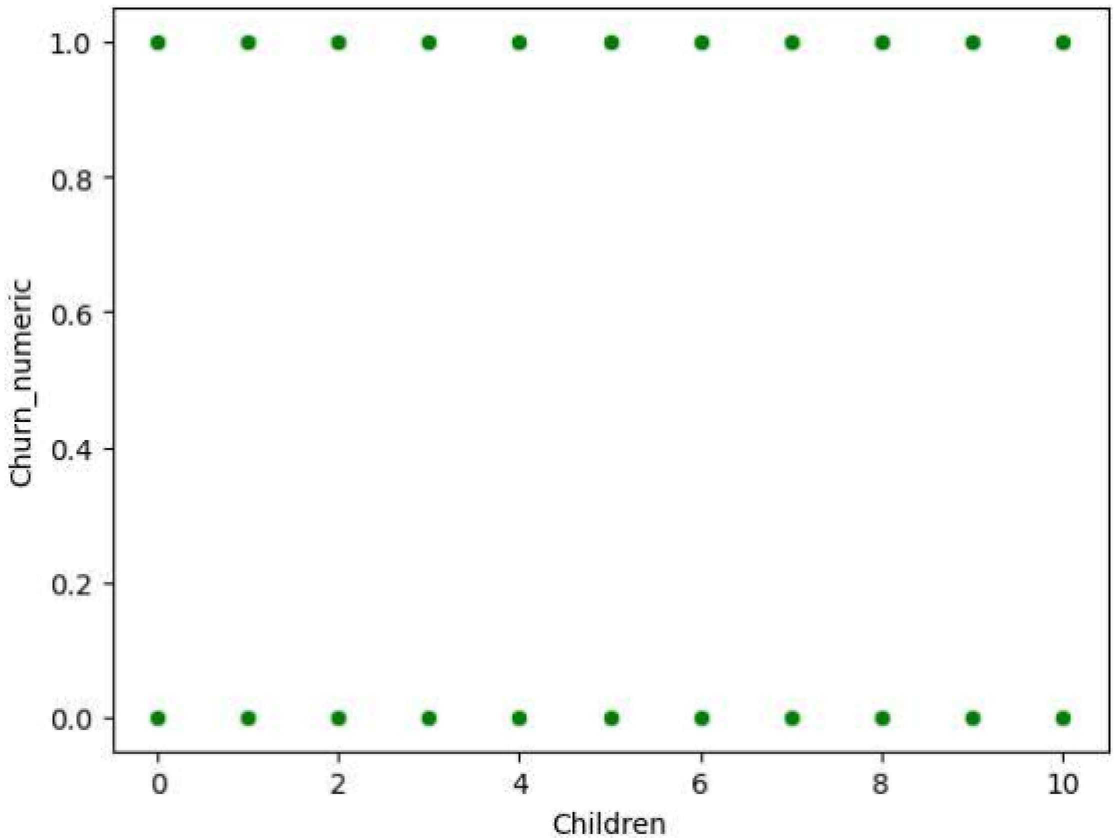
```
fpr, tpr, thresholds = roc_curve(y_test, y_pred_prob)
mpl.plot([0, 1], [0, 1], 'k--')
mpl.plot(fpr, tpr)
mpl.xlabel('False Positive Rate')
mpl.ylabel('True Positive Rate')
mpl.title('ROC Curve')
mpl.show()
```

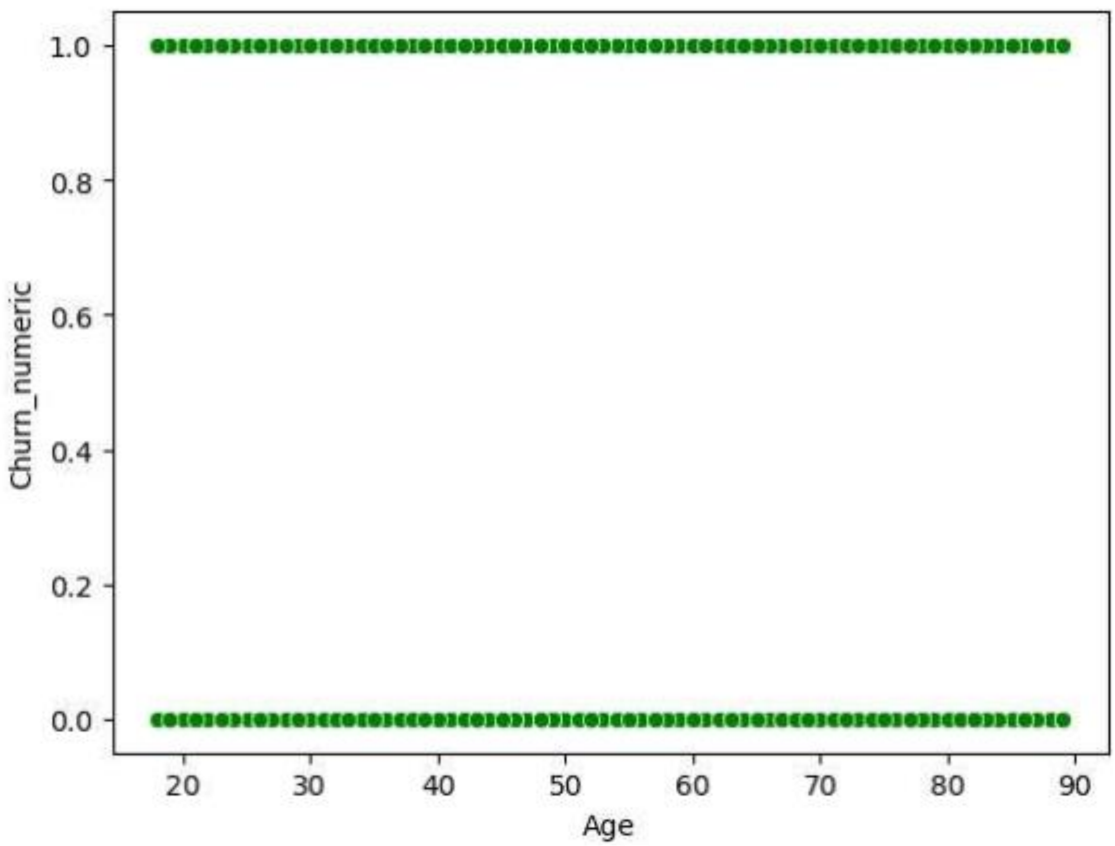


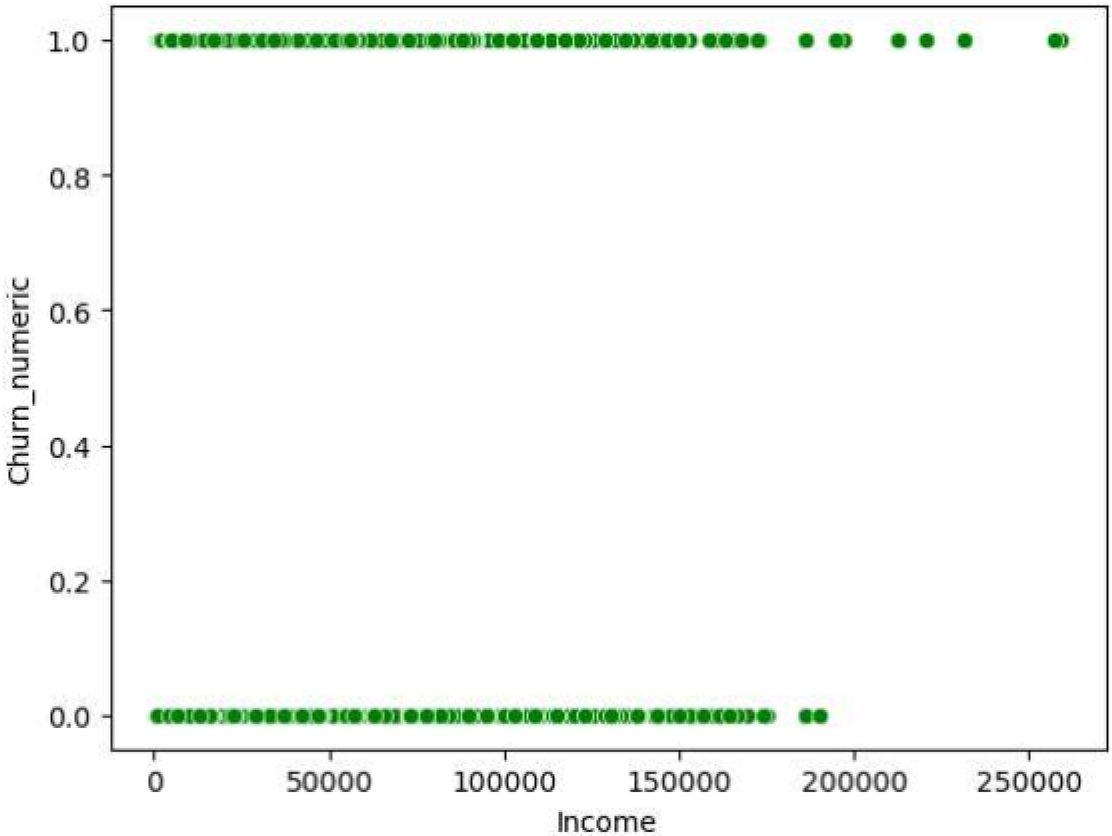
Univariate analysis

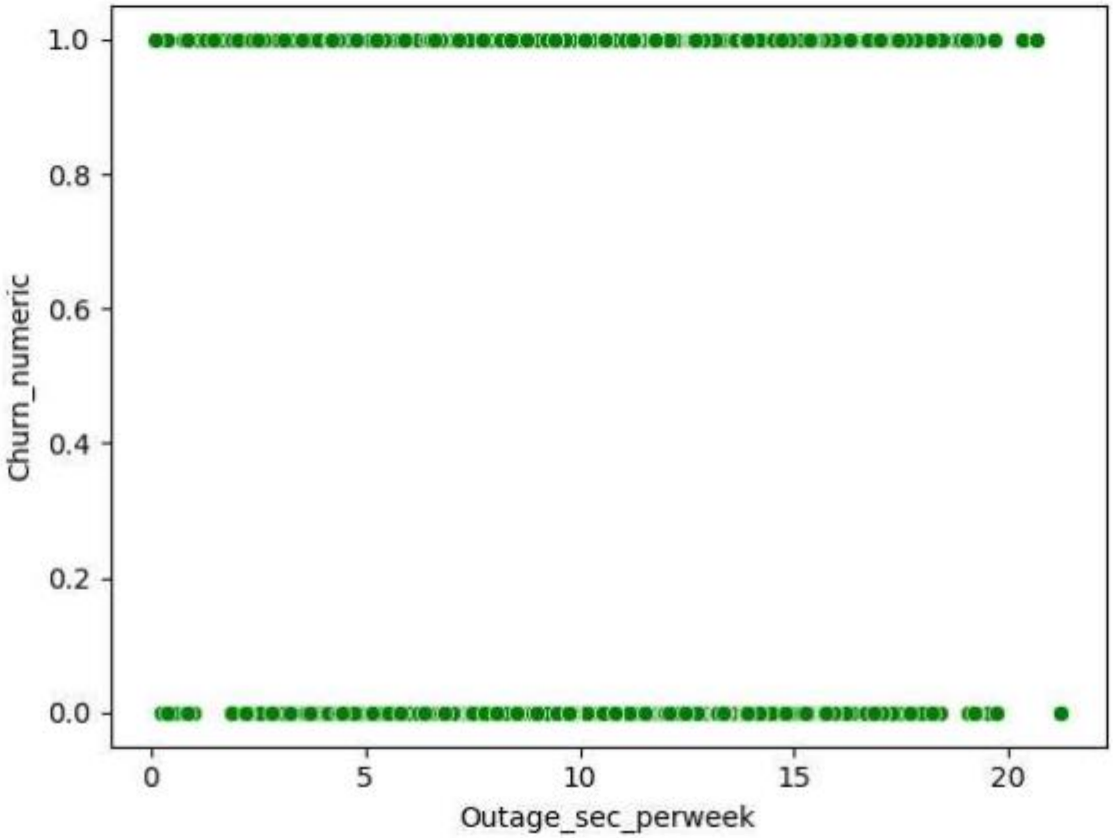


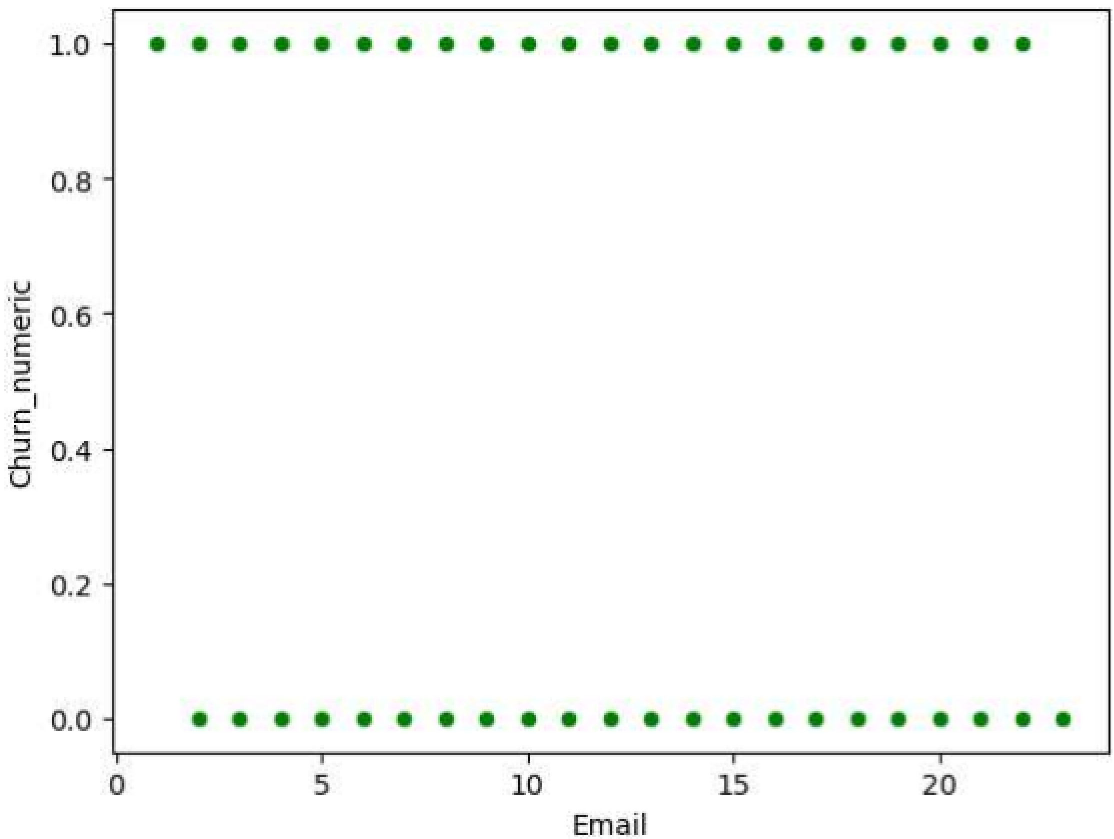
Bivariate Analysis:

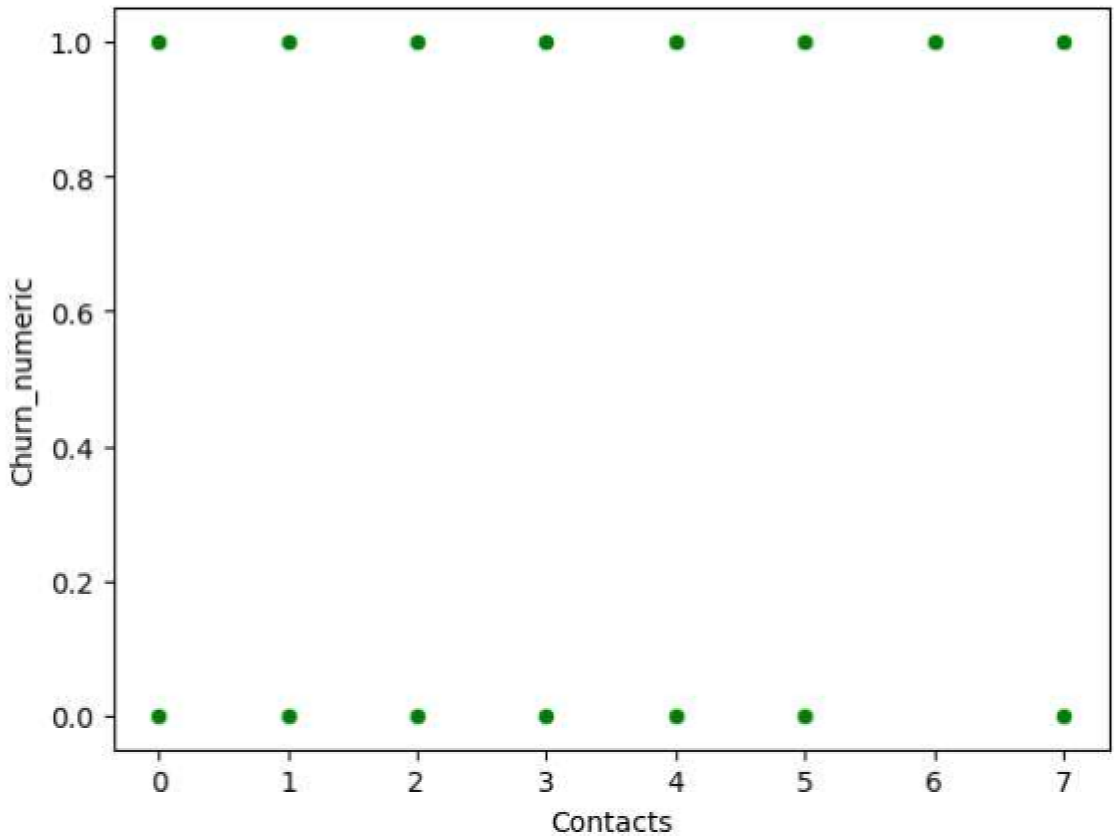


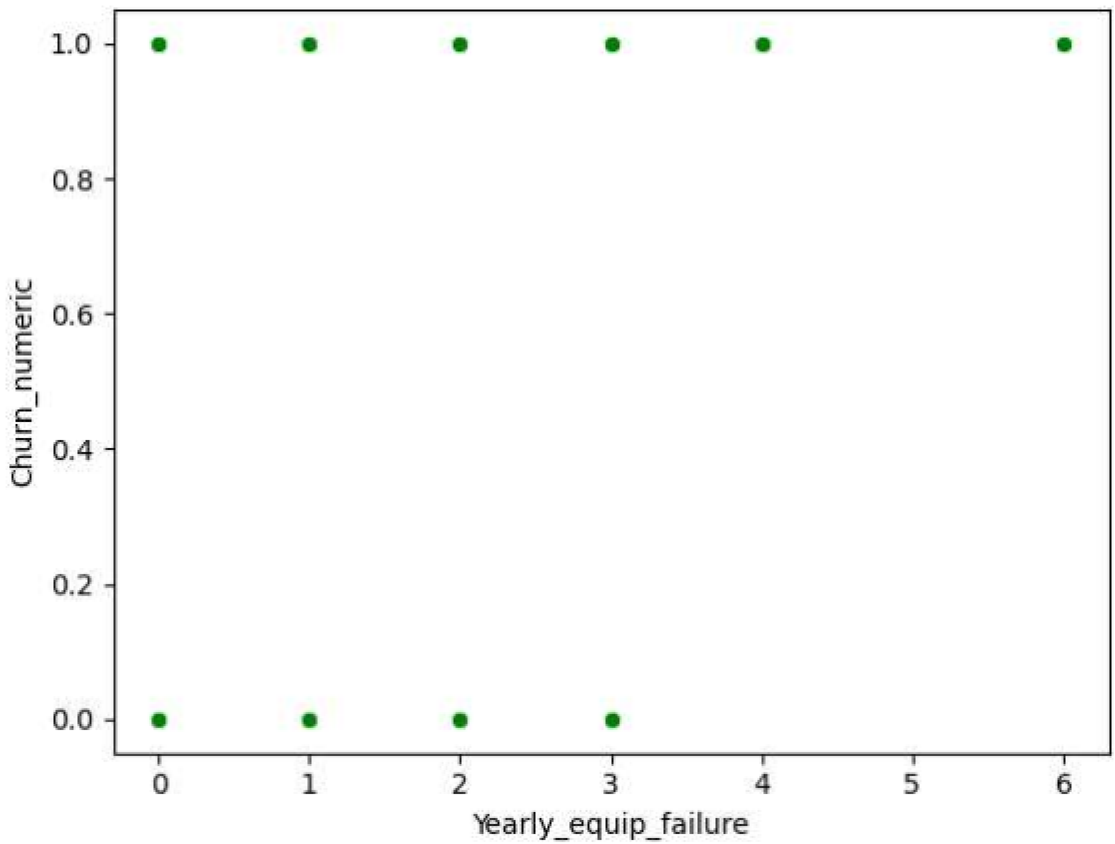

1.0
0.8
0.6
0.4
0.2
0.0
Churn_numeric
20
30
40
50
60
70
80
90
Age

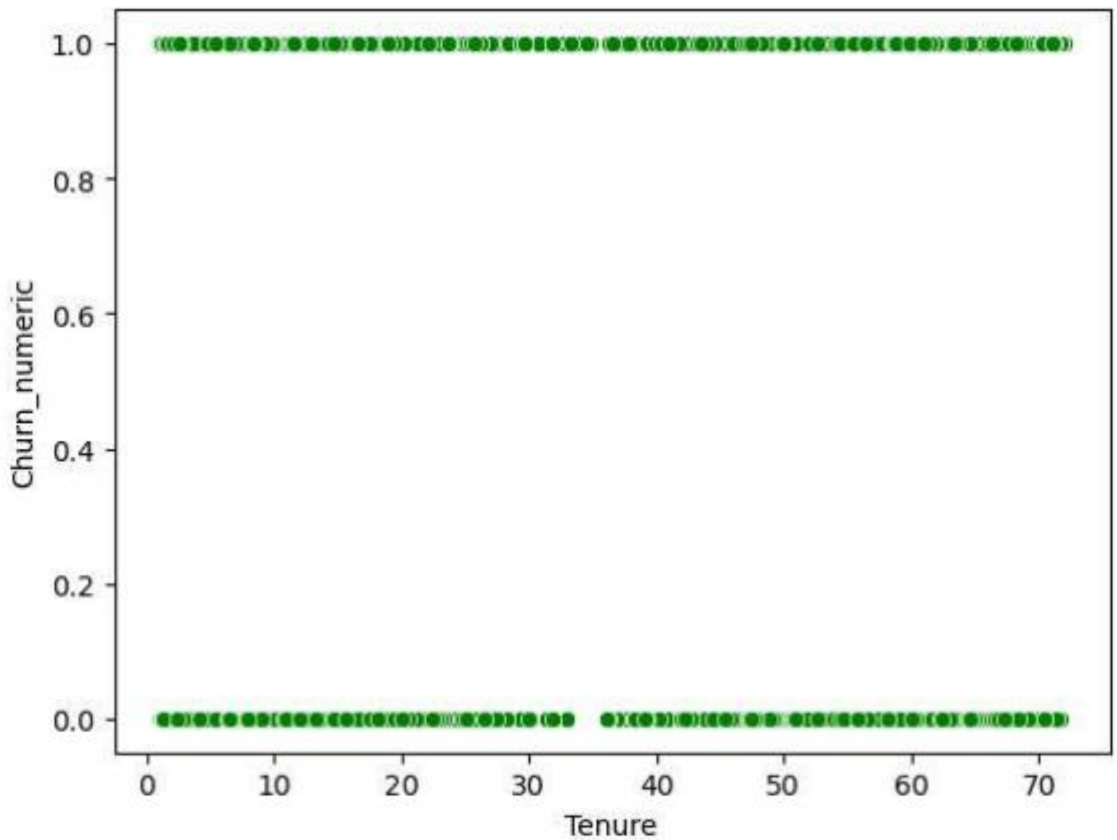

1.0
0.8
0.6
0.4
0.2
0.0
Churn_numeric
0
50000
100000
150000
200000
250000
Income

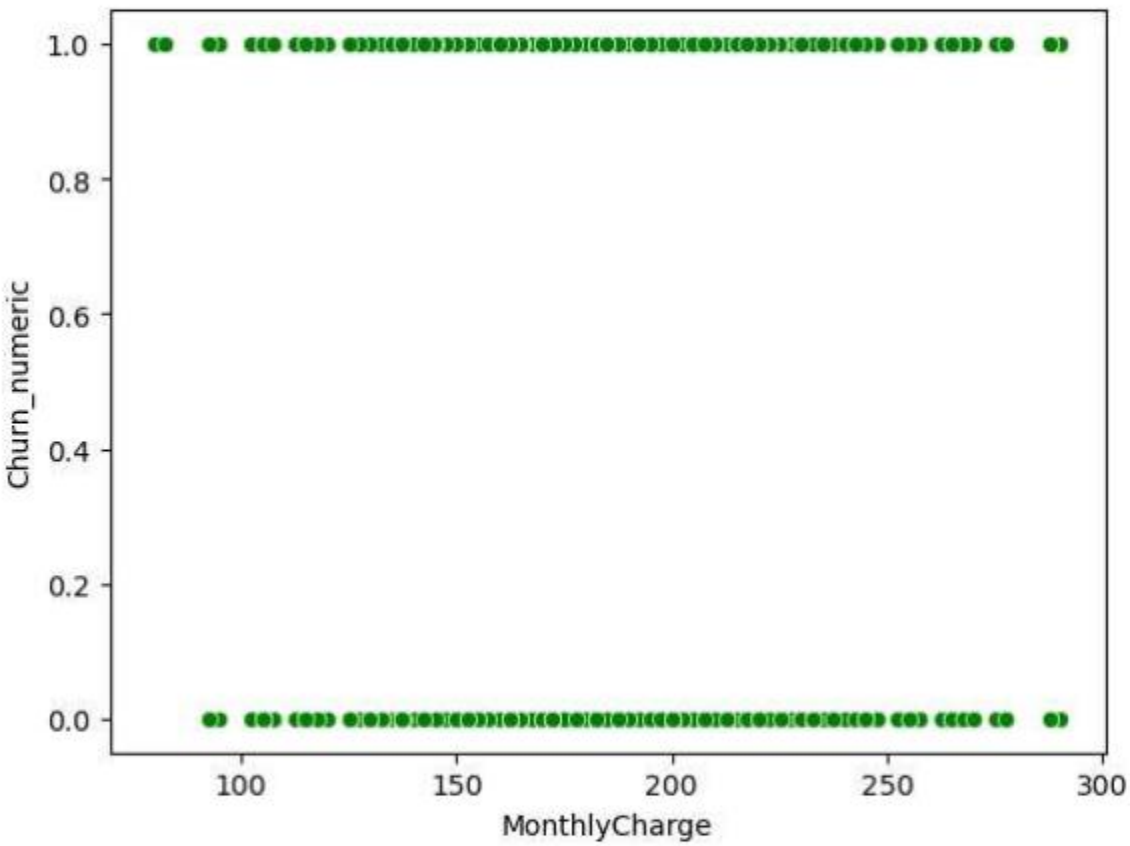

1.0
0.8
0.6
0.4
0.2
0.0
Churn_numeric
0
5
10
15
20
Outage_sec_perweek


1.0
0.8
0.6
0.4
0.2
0.0
Churn_numeric
0
5
10
15
20
Email

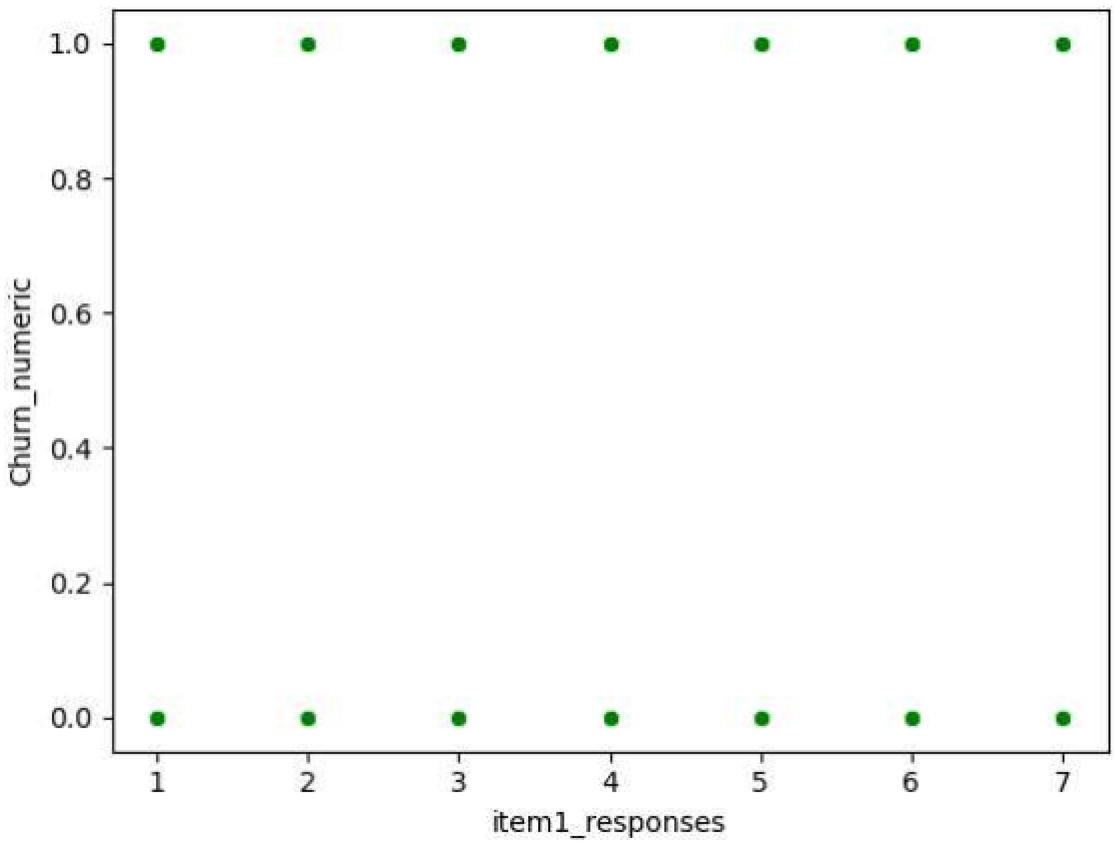

1.0
0.8
0.6
0.4
0.2
0.0
Churn_numeric
0
1
2
3
4
5
6
7
Contacts

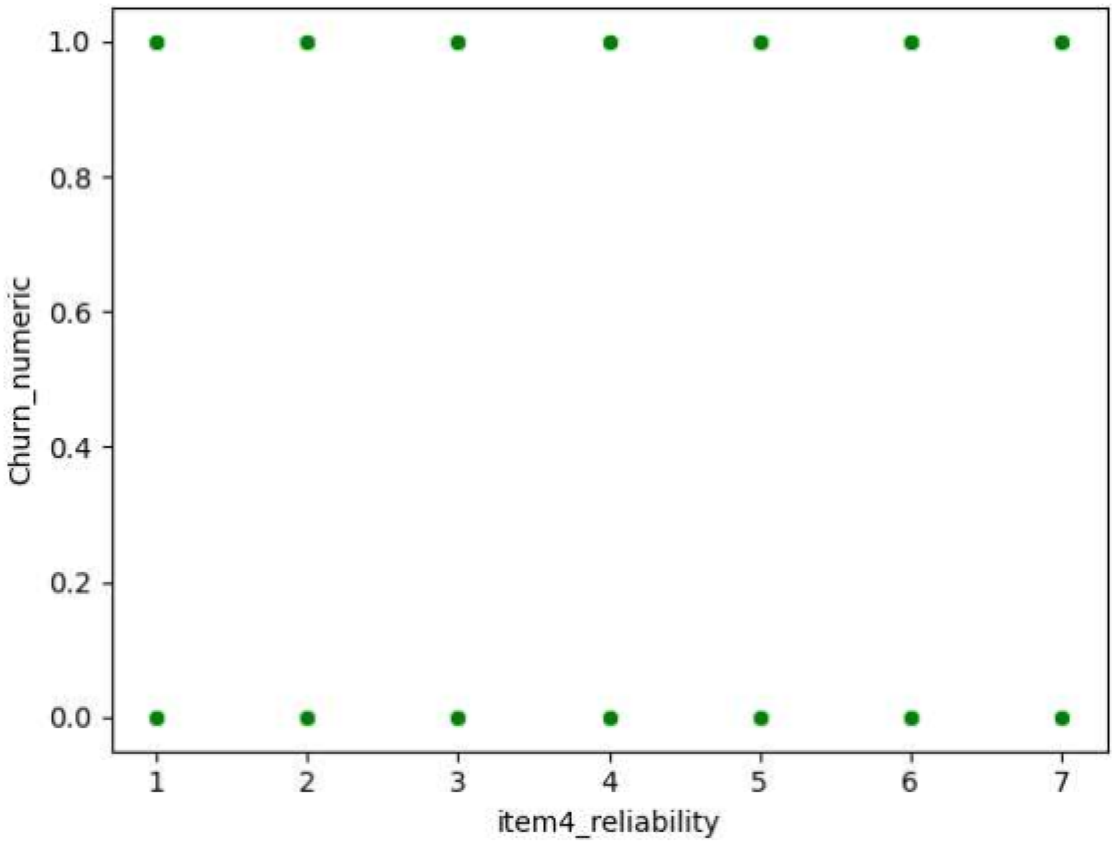

1.0
0.8
0.6
0.4
0.2
0.0
Churn_numeric
0
1
2
3
4
5
6
Yearly_equip_failure


1.0
0.8
0.6
0.4
0.2
0.0
Churn_numeric
0
10
20
30
40
50
60
70
Tenure

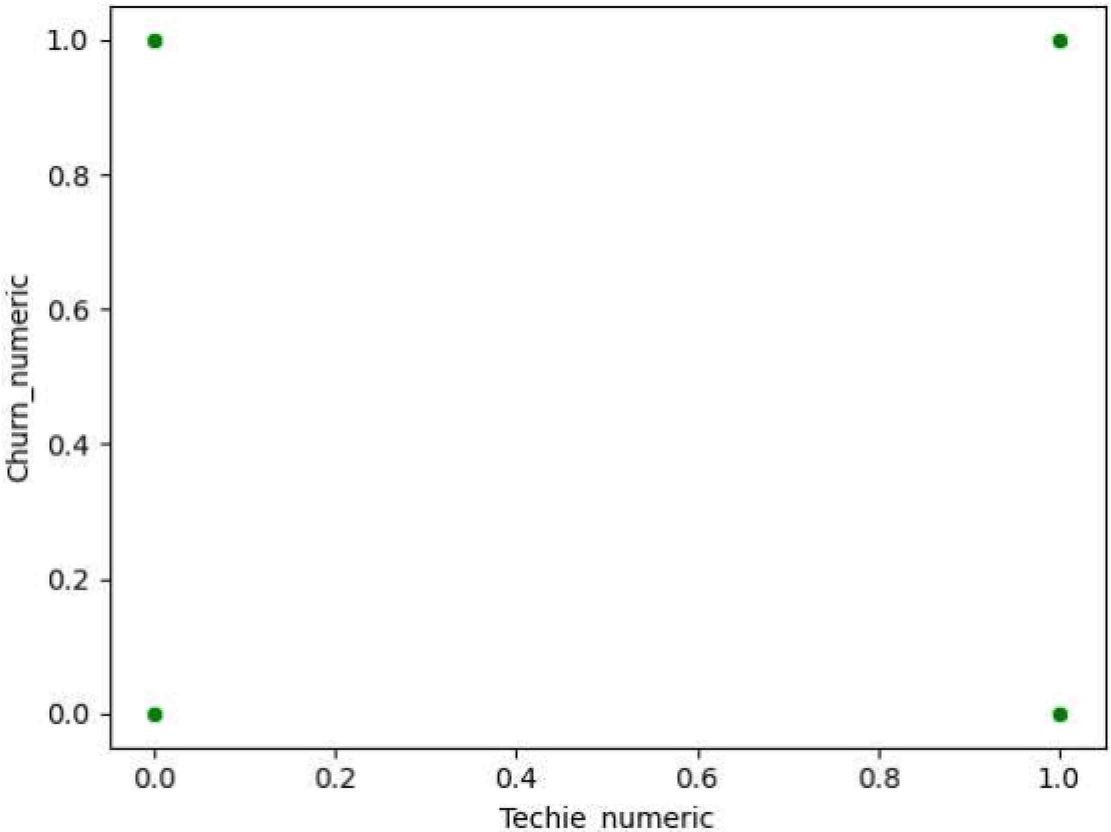


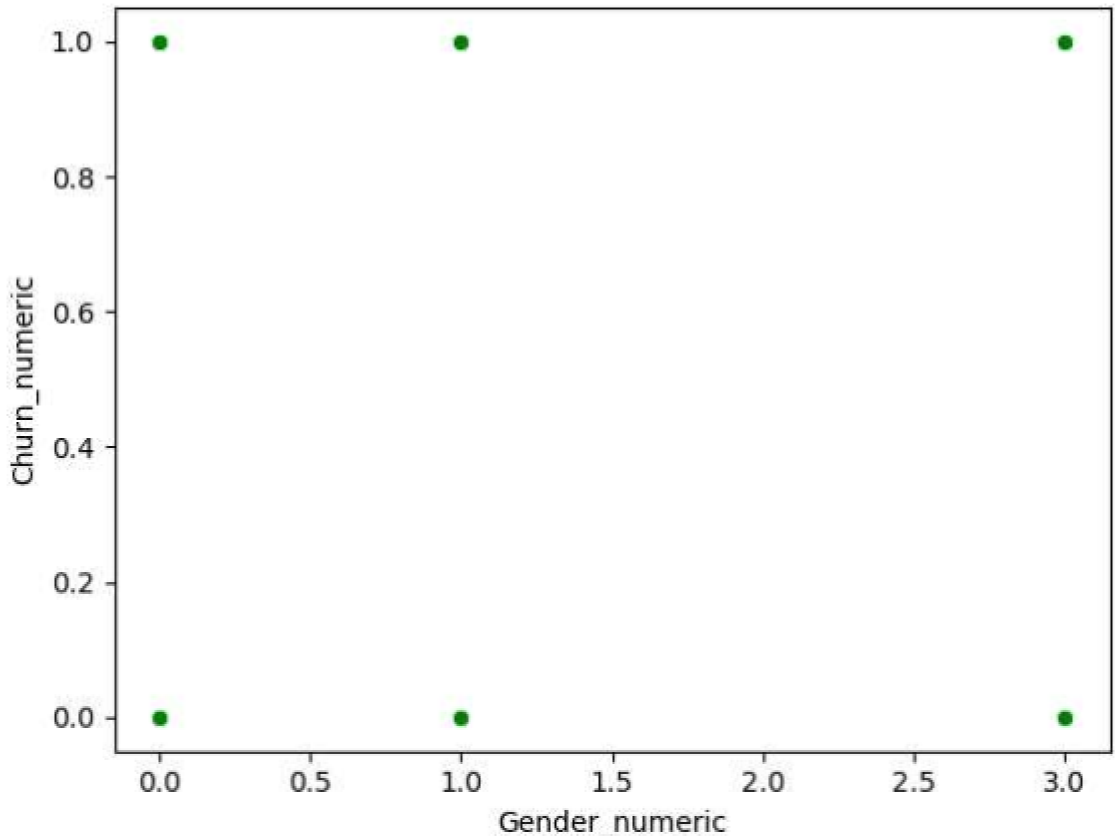
1.0
0.8
0.6
0.4
0.2
0.0
Churn_numeric
100
150
200
250
300
MonthlyCharge


1.0
0.8
0.6
0.4
0.2
0.0
Churn_numeric
1
2
3
4
5
6
7
item1_responses

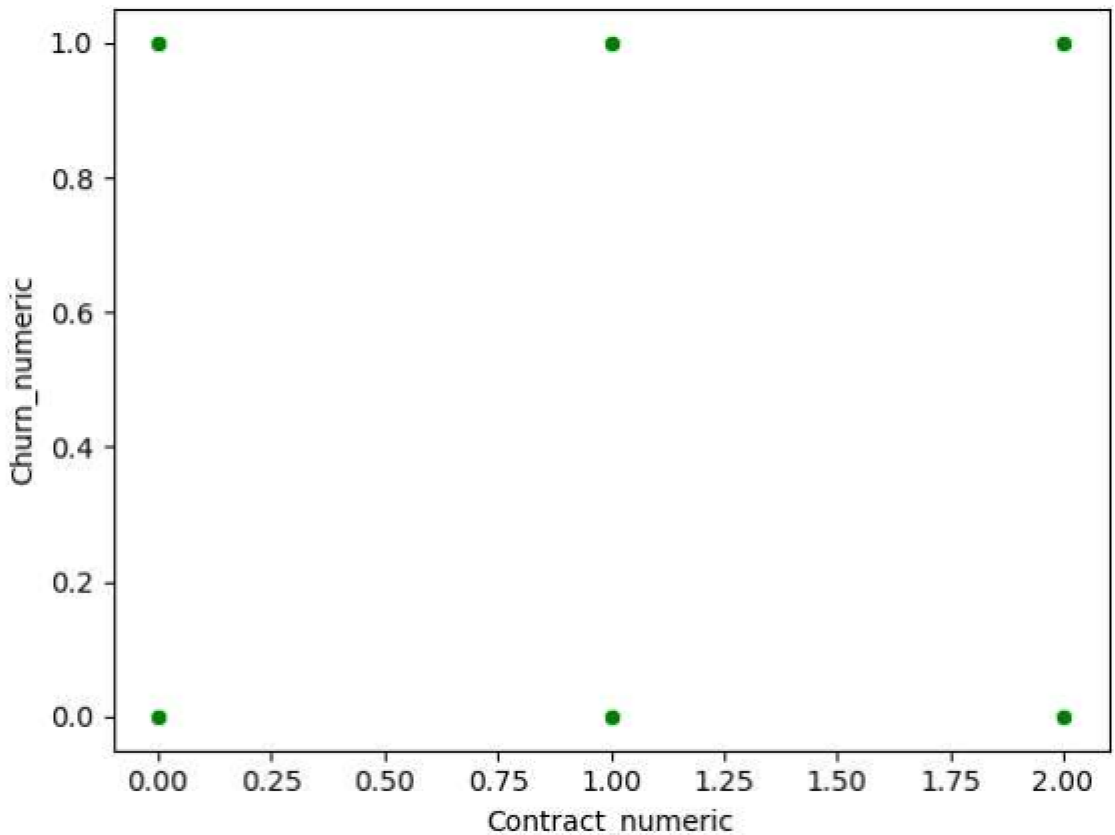


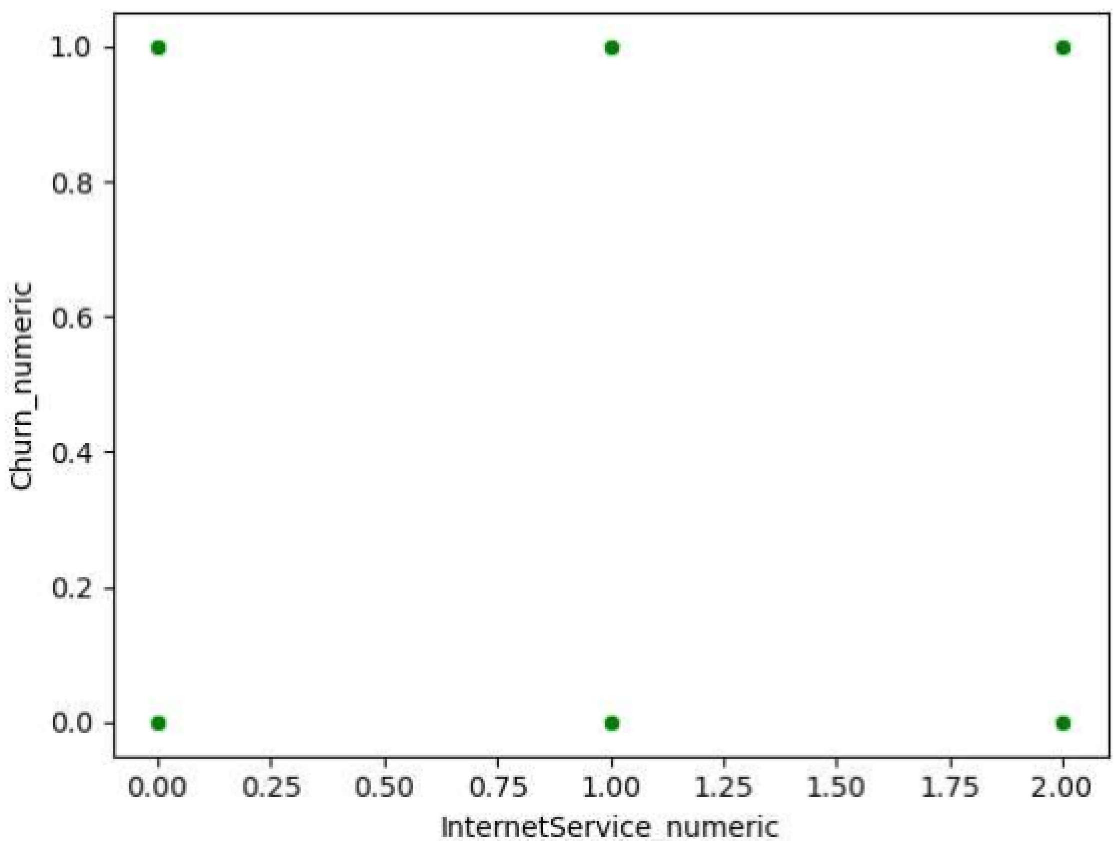
1.0
0.8
0.6
0.4
0.2
0.0
Churn_numeric
1
2
3
4
5
6
7
item4_reliability


1.0
0.8
0.6
0.4
0.2
0.0
Churn_numeric
0.0
0.2
0.4
0.6
0.8
1.0
Techie_numeric



Churn_numeric
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.5
1.0
1.5
2.0
2.5
3.0
Gender_numeric


1.0
0.8
0.6
0.4
0.2
0.0
Churn_numeric
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
2.00
Contract_numeric


1.0
0.8
0.6
0.4
0.2
0.0
Churn_numeric
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
2.00
InternetService_numeric

3. The code used to perform the classification analysis from part D2 is found in “D209_Task1_Code.ipynb”

Part V: Data Summary and Implications

E. Summarize your data analysis by doing the following:

1. Explain the accuracy and the area under the curve (AUC) of your classification model.

2. Discuss the results and implications of your classification analysis.

3. Discuss one limitation of your data analysis.

4. Recommend a course of action for the real-world organizational situation from part A1 based on your results and implications discussed in part E2.

1. The model had an initial accuracy of .726, or 72.6%. While this could be considered good or bad accuracy, that decision can be left up to the organization. After scaling our data and performing another round of modeling, we find an accuracy score of .801, or 81%, which is an improvement in the accuracy of this model and therefore, the utility of the model. The AUC shows whether a prediction is better than a random guess, which any value above .5 indicating it is, and below .5 would mean the prediction model is not better than random guessing. The area under the curve (AUC) of our classification model

is .8065 (supported by 5-fold validation), which indicates our model is a better prediction than random guessing.

2. The results of classification analysis show that using KNN classification can predict, with 80.1% accuracy, whether a customer can churn at a better ability than random guessing with 31 variables as the ideal number of neighbors. The implications are that we can use this model, further fine tuning it in another project or presenting it to stakeholders/project managers, to give an option to predict churn. Churn is a valuable indicator for the company of if a customer has left them in the last month, so predicting it can influence the retention efforts of the organization.
3. One limitation of my data analysis is that the accuracy was brought up to only 80.1%. This is not certainty or 100% accuracy, which is of course the ideal results an organization would want, so this classification method leaves room for error (19.9%). The data can be refined through the acquisition phase or the number of datapoints can be increased to try and improve this model or a new, more accurate model will need to be tested.
4. Based on the results and implications, I would recommend this organization to use this model as one of their tools to predict whether a customer is likely to churn, improving the retention efforts and giving the organization options of what to do to decrease the likelihood of a customer churning. One example of this is to further explore the 31 nearest neighbors identified in the model and see what can be done to alter them to increase retention (decrease churn probability).

Part VI: Demonstration

F. Provide a Panopto video recording that includes a demonstration of the functionality of the code used for the analysis and a summary of the programming environment.

Panopto Video Link:

https://wgu.hosted.panopto.com/Panopto/Pages/Viewer.aspx?id=ef03237b-39a8-4e85-a559-afa3016a5828

G. Record the web sources used to acquire data or segments of third-party code to support the analysis. Ensure the web sources are reliable.

Sources for code regarding the modeling:

Python, R. (n.d.). *The k-Nearest Neighbors (kNN) Algorithm in Python – Real Python*. Realpython.com. https://realpython.com/knn-python/

H. Acknowledge sources, using in-text citations and references, for content that is quoted, paraphrased, or summarized.

References

K Nearest Neighbors with Python | ML. (2019, June 7). GeeksforGeeks. https://www.geeksforgeeks.org/k-nearest-neighbors-with-python-ml/

Vishalmendekarhere. (2021, January 22). *It's All About Assumptions, Pros & Cons*. The Startup. https://medium.com/swlh/its-all-about-assumptions-pros-cons-497783cfed2d

I. Demonstrate professional communication in the content and presentation of your submission.

This aspect of the rubric is evaluated through the entirety of this report and I hope professionalism has shown continuously.