```
In [280]:
           # Import libraries
           import pandas as pd
           import seaborn as sns
           import matplotlib.pyplot as plt
           import statsmodels.api as sm
           import numpy as np
           from sklearn.metrics import accuracy_score, confusion_matrix
           # Import dataset
           df = pd.read_csv("C:/Users/hkeim/OneDrive/Documents/School/D208/churn clean.csv")
In [281]:
           # Create dummies for bianry objects
           df=pd.get_dummies(df, columns=['Churn', 'Techie', 'Port_modem', 'Tablet', 'Phone', 'Multiple'
                                                           'OnlineSecurity', 'OnlineBackup', 'DeviceProte
In [282]:
           # Drop 'No' dummies
           df=df.drop(['Churn_No', 'Techie_No', 'Port_modem_No', 'Tablet_No', 'Phone No', 'Multiple No',
                       'StreamingTV_No', 'StreamingMovies_No', 'PaperlessBilling_No'], axis=1)
In [283]:
           #drop unnecessary variables
           df=df.drop(['Customer_id', 'Interaction', 'UID', 'City', 'County', 'Job', 'Item1', 'Item2', '
                            'Item5', 'Item6', 'Item7', 'Item8'], axis=1)
In [284]:
           # Use Pearson Correlation to choose initial model variables
           plt.figure(figsize=(12,10))
           cor = df.corr()
           sns.heatmap(cor, annot=True, cmap=plt.cm.Reds)
           plt.show()
```

```
1.00
                                       .04<mark>-0.9</mark>) 0.45, 00.70 080.02900.20 07/90 07/910 08 2008 0 07250 820 0740 1-0.0010 074 6 95, 80 08 30 174 0 2, 40 1580 910 075 07
                                                                                                                                                                                                 0.75
              Population
                 Children
                                                                                                                                                                                                  0.50
                   Income
Outage sec perweek
                 Contacts
                                                                                                                                                                                                 0.25
                                                 QEE 6 00 A SOOT B O 25 B O 25 B O 25 D O 25 E O 0 61 O . O - D 2 O O 0 7 D 1 A 2 O O 0 60 0 8 SOO 5 O O 3 O O 0 7 SO O D 2 O O 0 5 G O 0 2 O O 0 5 G O 0 2 O O 0 4 5 D 1
  Yearly_equip_failure
                                                        B 6 0 5 D D 70 002 Q 0 20 90 Q 4 0 20 80 1 <mark>2 1 0</mark> . 0 0 303 9 <mark>9 0 . 4 9</mark> 0 . 00 10 0 83 D 00 5040 3 CR 00 1 0 0 50 2 1 . 0 2 8 C
                                              D40-D20-004-80-9-8-1-0.00-8-0-2-0-0-20-04-30-07/20-0-331-0-0-00-307-0-0-7/20-0-10-3-20-0-4-8-2-60-1-60-1-20-4-80-601-00-
        MonthlyCharge
                                        DASCO 1 70 CB 40 CB 59 255 CO 150 CB 70 44 20 O 150 CB 30 1 <mark>2</mark>0.950 CO 6 1 <mark>0 .44</mark> CO 95 CCB 70 O 0 DO 1 B C 630 21 C 42 CO 1750 CO 1254 C
                                                                                                                                                                                                 - 0 00
                                                       9860490560590016020086010.4\frac{9}{0}.3\frac{70}{0}.4\frac{4}{0}10.067098200980261-9.014051.056019.230.29.00
                                                                                                                 82014D.000990-20100-3093.D920
      Port_modem_Yes
                                                   DD 01 201 01 40 00 05 60 8 00 1 10 15 00 15 00 1 10 20 1 10 20 1 10 00 94 0 0 02 50 00 00 00 10 0 10 0 10 00 10
                                                                                                                                                                                                 -0.25
               Tablet Yes
                                                                                   CROSS 40 007/50-20 30 0 50 0-10 30 2 60 003 0.0 20 10 2 5 1 0 0 003 4 0 0 1 20 0 3 12 40 1
               Phone_Yes
            Multiple Yes
   OnlineSecurity Yes
                                                                                                                                                                                                  -0.50
    OnlineBackup Yes
DeviceProtection Yes
       TechSupport Yes
      StreamingTV_Yes
                                                                                                                                                                                                 -0.75
StreamingMovies Yes
  PaperlessBilling Yes
                                                                                                                 Techie_Yes.
                                     Zip
Lat
                                                                                        éarly_equip_failure
                                                                                                   MonthlyCharge
                                                                                                                                                 OnlineBackup_Yes
                                                                                                                                                                            PaperlessBilling_Yes
                                                                         Outage_sec_perweek
                                                                                                        Bandwidth_GB_Year
                                                                                                              chum_Yes
                                                                                                                        Port modem Yes
                                                                                                                                       Multiple_Yes
                                                                                                                                             OnlineSecurity_Yes
                                                                                                                                                       DeviceProtection_Yes
                                                                                                                                                                 StreamingTV_Yes
```

```
In [285]: # Create dataframe with initial variables
df=df.filter(items=['CaseOrder', 'Tenure', 'MonthlyCharge', 'Bandwidth_GB_Year', 'Churn_Yes',
```

In [286]:

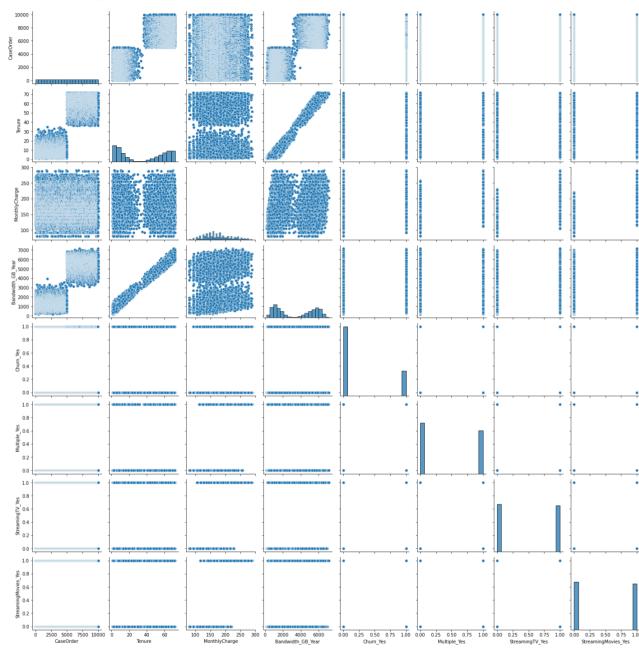
Summary statistics
print(df.describe())

count mean std min 25% 50%	CaseOrder 10000.00000 5000.50000 2886.89568 1.00000 2500.75000 5000.50000	Tenure 10000.000000 34.526188 26.443063 1.000259 7.917694 35.430507	MonthlyCharge 10000.000000 172.624816 42.943094 79.978860 139.979239 167.484700	Bandwidth_GB_Year \ 10000.000000 3392.341550 2185.294852 155.506715 1236.470827 3279.536903
75%	7500.25000	61.479795	200.734725	5586.141370
max	10000.00000	71.999280	290.160419	7158.981530
	Churn_Yes	Multiple_Yes	StreamingTV_Ye	s StreamingMovies_Yes
count	10000.000000	10000.000000	10000.00000	0 10000.000000
mean	0.265000	0.460800	0.49290	0.489000
std	0.441355	0.498486	0.49997	5 0.499904
min	0.000000	0.000000	0.00000	0.000000
25%	0.000000	0.000000	0.00000	0.000000
50%	0.000000	0.000000	0.00000	0.000000

75% 1.000000 1.000000 1.000000 1.000000 max 1.000000 1.000000 1.000000 1.000000

In [287]: # Univariate and bivariate visualizations
sns.pairplot(df)

Out[287]: <seaborn.axisgrid.PairGrid at 0x2a7451fc6d0>



```
# Defining the independent variables
X=df[['CaseOrder', 'Tenure', 'MonthlyCharge', 'Bandwidth_GB_Year', 'Multiple_Yes', 'Streaming'
```

In [289]: # Get initial model intercept
X = sm.add_constant(X)

In [290]: # Defining the dependent Variable
y=df['Churn_Yes']

```
In [291]:
               # Initial Logistic regression model
               model = sm.Logit(endog=y, exog=X).fit()
               print(model.summary())
              Optimization terminated successfully.
                           Current function value: inf
                           Iterations 8
                                                   Logit Regression Results
              ______
                                                  Churn_Yes No. Observations:
                                                                                                                10000
              Dep. Variable:
              Model:
                                                       Logit Df Residuals:
                                                                                                                   9992
                                                            MLE Df Model:
              Method:
                                                                                                                       7
                                         Sat, 09 Oct 2021 Pseudo R-squ.:
              Date:
                                                                                                                     inf
                                         21:46:55 Log-Likelihood:
              Time:
                                                                                                                    -inf
                                                           True LL-Null:
              converged:
                                                                                                                 0.0000
              Covariance Type: nonrobust LLR p-value:
                                                                                                               1.000
              ______
                                                coef std err z P > |z| [0.025 0.975]

        const
        -3.8150
        0.193
        -19.784
        0.000
        -4.193
        -3.437

        CaseOrder
        6.086e-06
        2.01e-05
        0.302
        0.762
        -3.34e-05
        4.55e-05

        Tenure
        -0.2347
        0.012
        -19.684
        0.000
        -0.258
        -0.211

        MonthlyCharge
        0.0104
        0.002
        6.730
        0.000
        0.007
        0.013

        Bandwidth_GB_Year
        0.0019
        0.000
        13.694
        0.000
        0.002
        0.002

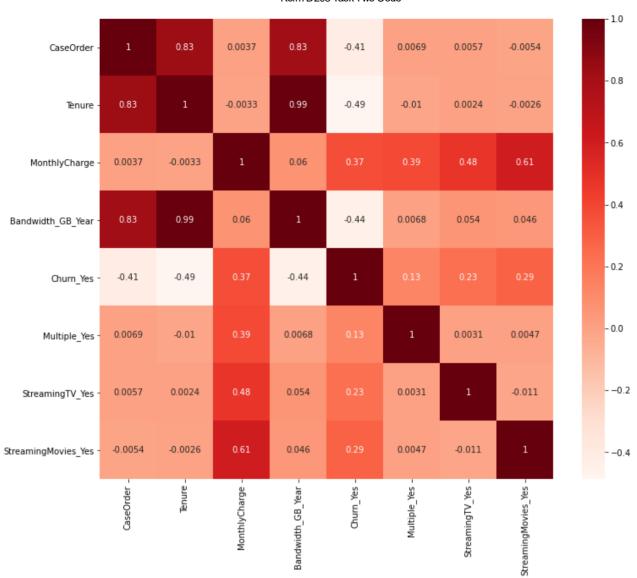
        Multiple_Yes
        0.6336
        0.082
        7.766
        0.000
        0.474
        0.793

        StreamingTV_Yes
        1.1356
        0.099
        11.498
        0.000
        0.942
        1.329

        StreamingMovies_Yes
        1.4812
        0.109
        13.642
        0.000
        1.268
        1.694

              ______
In [292]:
               # Initial model odds ratios
               np.exp(model.params)
Out[292]: const
                                              0.022037
              CaseOrder
                                             1.000006
              Tenure
                                             0.790815
              MonthlyCharge
                                             1.010412
              Bandwidth_GB_Year
Multiple_Yes
StreamingTV_Yes
                                             1.001879
                                           1.884349
                                            3.113014
              StreamingMovies_Yes 4.398402
              dtype: float64
In [293]:
               # Initial model predictions
               pred = model.predict(exog=X)
               pred.head()
Out[293]: 0
                  0.390642
                  0.960092
              1
                  0.528901
              2
                  0.200313
                    0.267200
              dtype: float64
In [294]:
               # Initial model rounded predictions
               round(pred)
Out[294]: 0
                         0.0
              1
                         1.0
              2
                         1.0
              3
                         0.0
              4
                         0.0
```

```
9995
                  0.0
          9996
                  0.0
          9997
                  0.0
          9998
                  0.0
          9999
                  0.0
          Length: 10000, dtype: float64
In [295]:
           # Initial model confusion matrix
           confusion_matrix(y_true=list(y), y_pred=list(round(pred)))
Out[295]: array([[6799, 551],
                 [ 944, 1706]], dtype=int64)
In [296]:
           # Initial model accuracy of the fitted model
           accuracy score(y true=list(y), y pred=list(round(pred)))
Out[296]: 0.8505
In [297]:
           # Use Pearson Correlation to choose reduced model variables
           plt.figure(figsize=(12,10))
           cor = df.corr()
           sns.heatmap(cor, annot=True, cmap=plt.cm.Reds)
           plt.show()
```



In [298]: # Pairwise correlation analysis
X.corr()

Out[298]:		const	CaseOrder	Tenure	MonthlyCharge	Bandwidth_GB_Year	Multiple_Yes	Streamin
	const	NaN	NaN	NaN	NaN	NaN	NaN	
	CaseOrder	NaN	1.000000	0.832550	0.003677	0.825561	0.006915	
	Tenure	NaN	0.832550	1.000000	-0.003337	0.991495	-0.010422	
	MonthlyCharge	NaN	0.003677	-0.003337	1.000000	0.060406	0.385979	
	Bandwidth_GB_Year	NaN	0.825561	0.991495	0.060406	1.000000	0.006823	
	Multiple_Yes	NaN	0.006915	-0.010422	0.385979	0.006823	1.000000	
	StreamingTV_Yes	NaN	0.005690	0.002440	0.482312	0.054314	0.003097	
	StreamingMovies_Yes	NaN	-0.005353	-0.002574	0.608115	0.045600	0.004691	-

```
# Defining the reduced independent variables
Xr=df[['MonthlyCharge', 'Bandwidth_GB_Year', 'Multiple_Yes', 'StreamingTV_Yes']]
```

```
In [300]:
           # Get reduced model intercept
           Xr = sm.add constant(Xr)
In [301]:
           # Reduced Logistic regression model
           modelr = sm.Logit(endog=y, exog=Xr).fit()
           print(modelr.summary())
           Optimization terminated successfully.
                    Current function value: inf
                    Iterations 7
                                      Logit Regression Results
           _____
           Dep. Variable:
                                       Churn Yes No. Observations:
                                                                                     10000
           Model:
                                            Logit Df Residuals:
                                                                                       9995
           Method:
                                              MLE Df Model:
                                                                                          4
           Date:
                               Sat, 09 Oct 2021 Pseudo R-squ.:
                                                                                        inf
           Time:
                                       21:48:25 Log-Likelihood:
                                                                                        -inf
                                             True LL-Null:
                                                                                      0.0000
           converged:
           Covariance Type: nonrobust LLR p-value:
                                                                                      1.000
           ______
                                 coef std err z P > |z| [0.025 0.975]
           ______

      const
      -4.7256
      0.149
      -31.783
      0.000
      -5.017
      -4.434

      MonthlyCharge
      0.0311
      0.001
      30.962
      0.000
      0.029
      0.033

      Bandwidth_GB_Year
      -0.0008
      2e-05
      -41.530
      0.000
      -0.001
      -0.001

      Multiple_Yes
      0.0184
      0.065
      0.284
      0.777
      -0.109
      0.145

      StreamingTV_Yes
      0.5945
      0.069
      8.635
      0.000
      0.460
      0.729

           ______
In [302]:
           # Reduced model odds ratios
           np.exp(modelr.params)
                                0.008866
Out[302]: const
                             1.031545
           MonthlyCharge
           Bandwidth_GB_Year
                                0.999171
           Multiple Yes
                                1.018540
           StreamingTV Yes
                                1.812107
           dtype: float64
In [303]:
           # Reduced model predictions
           pred = modelr.predict(exog=Xr)
           pred.head()
Out[303]: 0
               0.470017
               0.940376
           1
               0.190846
           2
               0.099609
           3
                0.574606
           dtype: float64
In [304]:
           # Reduced model rounded predictions
           round(predr)
                   0.0
Out[304]: 0
           1
                   1.0
           2
                   0.0
           3
                   0.0
                   1.0
                  . . .
           9995
                   0.0
```

```
9996
                  0.0
          9997
                  0.0
          9998
                  0.0
          9999
                  0.0
          Length: 10000, dtype: float64
In [305]:
           # Reduced model confusion matrix
           confusion_matrix(y_true=list(y), y_pred=list(round(predr)))
Out[305]: array([[6744, 606],
                 [1080, 1570]], dtype=int64)
In [306]:
           # Reduced model accuracy of the fitted model
           accuracy_score(y_true=list(y), y_pred=list(round(predr)))
Out[306]: 0.8314
In [307]:
           # Export regression data set
           df.to_csv("C:/Users/hkeim/OneDrive/Documents/School/D208/Keim D208 Task Two Prepared Data.csv
  In [ ]:
```