Capstone 2:
Supervised Learning Capstone
Telco Customer Churn via Kaggle

May, 2020

Overview of Assignment

- Go out and find a dataset of interest.
  - Telco Customer Churn - IBM Watson analytics
  - Predict behavior to retain customers.
- Explore the data
- Model your outcome of interest.
  - Naive Bayes - Applies Bayes' theorem with strong feature independence assumptions
  - K-Nearest Neighbors (KNN) - Input consists of the K closest training samples in the feature space
  - Decision Tree - Data is continuously split according to a certain parameter
  - Random Forest - Combines results from Decision Trees at the end of the process.
  - Logistic Regression - Typically used when the dependent variable is binary
  - Support Vector Machine (SVM) - Uses classification algorithms for two-group classification problems.
  - Gradient Boosting - Combines results from Decision Trees along the way.

Features:

String

- CustomerID

Categorical

- Gender
- SeniorCitizen
- Partner
- Dependents
- PhoneService
- MultipleLines
- InternetService
- OnlineSecurity
- OnlineBackup
- DeviceProtection
- TechSupport
- StreamingTV
- StreamingMovies
- Contract
- PaperlessBilling
- PaymentMethod
- Churn

Continuous

- Tenure
- MonthlyCharges
- TotalCharges

Capstone objective:

Customer churn is when a customer ends their relationship with a business. Looking at the Telco data from the customers perspectives and services, I use various models to predict behavior and decisions to retain customers.

Exploration:

```
df.info()
#no null, but why is total charges an object?
```

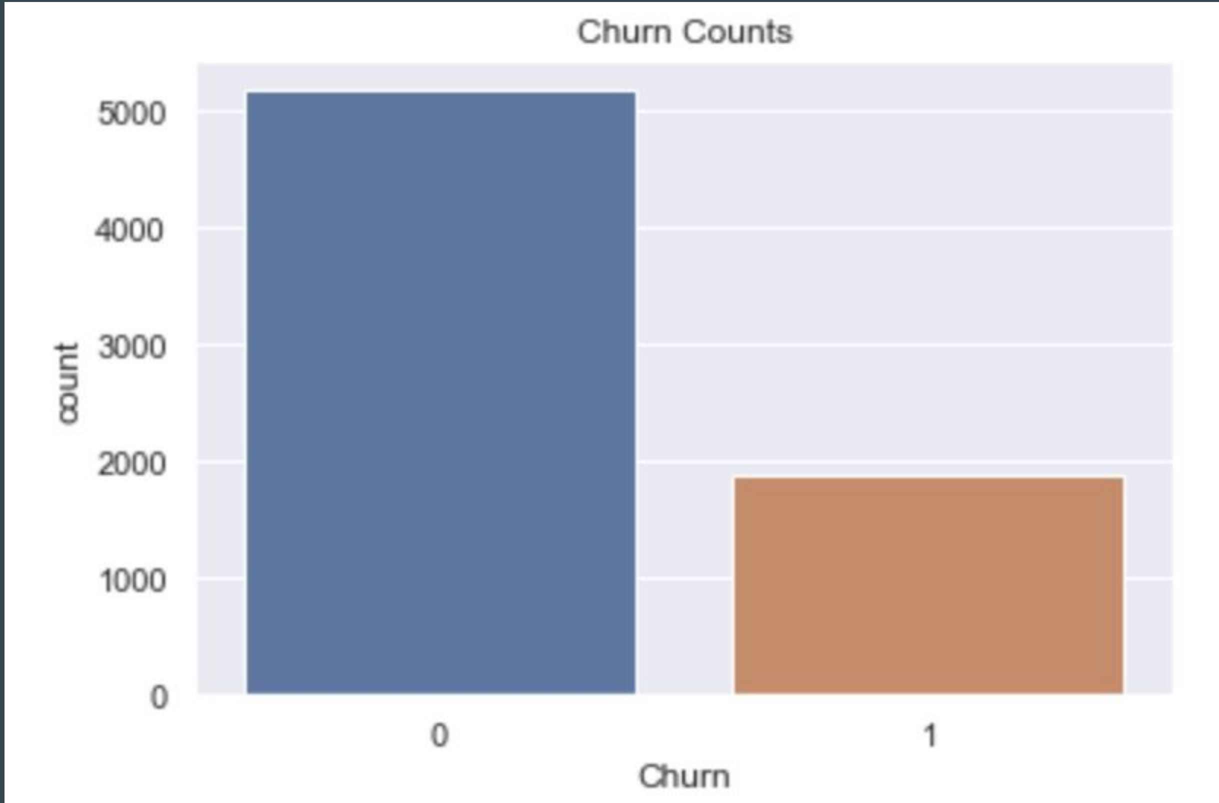
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7043 entries, 0 to 7042
Data columns (total 21 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   customerID        7043 non-null   object
 1   gender            7043 non-null   object
 2   SeniorCitizen     7043 non-null   int64
 3   Partner           7043 non-null   object
 4   Dependents        7043 non-null   object
 5   tenure            7043 non-null   int64
 6   PhoneService      7043 non-null   object
 7   MultipleLines     7043 non-null   object
 8   InternetService   7043 non-null   object
 9   OnlineSecurity    7043 non-null   object
 10  OnlineBackup      7043 non-null   object
 11  DeviceProtection  7043 non-null   object
 12  TechSupport       7043 non-null   object
 13  StreamingTV       7043 non-null   object
 14  StreamingMovies   7043 non-null   object
 15  Contract          7043 non-null   object
 16  PaperlessBilling  7043 non-null   object
 17  PaymentMethod     7043 non-null   object
 18  MonthlyCharges    7043 non-null   float64
 19  TotalCharges      7043 non-null   object
 20  Churn             7043 non-null   object
dtypes: float64(1), int64(2), object(18)
memory usage: 1.1+ MB
```

Unique Values for Features
-Convert to Binary

```
customerID          [7590-VHVEG, 5575-GNVDE, 3668-QPYBK, 7795-CFOC...
gender                                                 [Female, Male]
SeniorCitizen                                                  [0, 1]
Partner                                                     [Yes, No]
Dependents                                                  [No, Yes]
tenure              [1, 34, 2, 45, 8, 22, 10, 28, 62, 13, 16, 58, ...
PhoneService                                                [No, Yes]
MultipleLines                             [No phone service, No, Yes]
InternetService                                [DSL, Fiber optic, No]
OnlineSecurity                         [No, Yes, No internet service]
OnlineBackup                           [Yes, No, No internet service]
DeviceProtection                       [No, Yes, No internet service]
TechSupport                            [No, Yes, No internet service]
StreamingTV                            [No, Yes, No internet service]
StreamingMovies                        [No, Yes, No internet service]
Contract                         [Month-to-month, One year, Two year]
PaperlessBilling                                            [Yes, No]
PaymentMethod       [Electronic check, Mailed check, Bank transfer...
MonthlyCharges      [29.85, 56.95, 53.85, 42.3, 70.7, 99.65, 89.1,...
TotalCharges        [29.85, 1889.5, 108.15, 1840.75, 151.65, 820.5...
Churn                                                       [No, Yes]
dtype: object
```

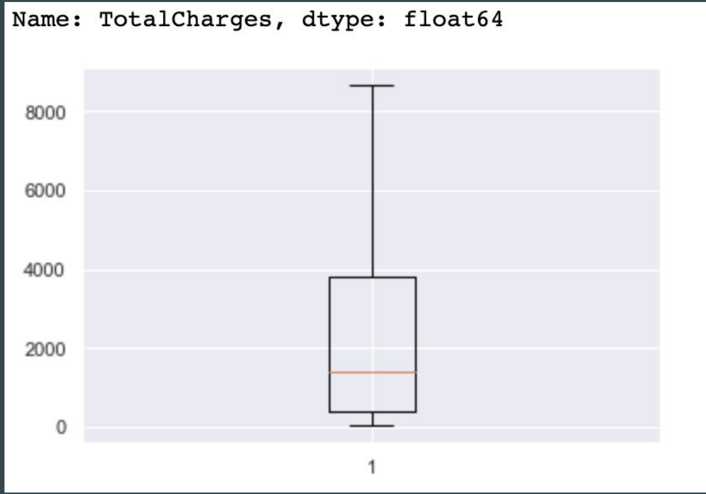
Split Churn Samples
Distribution Plot

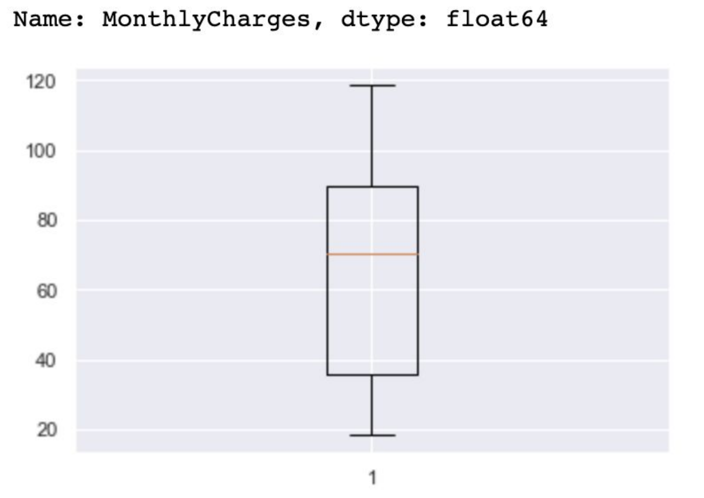
Churn Counts
No Churn Customers: 5,174
Churn Customers: 1,869

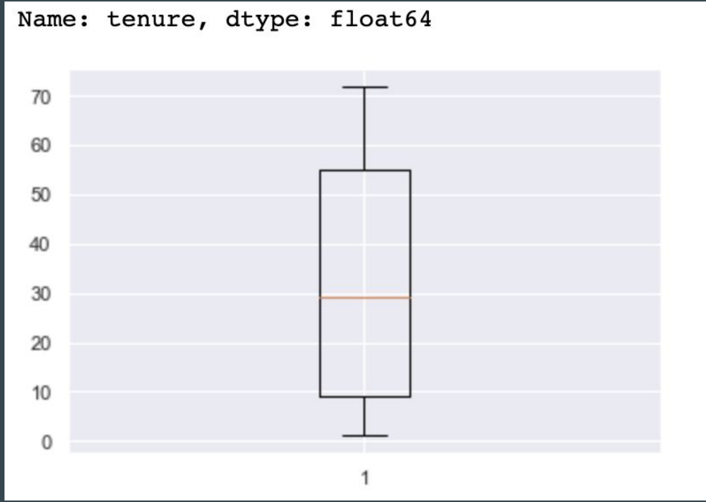



```
0    5174
1    1869
Name: Churn, dtype: int64
```

Check Continuous Variables for Outliers via Boxplot





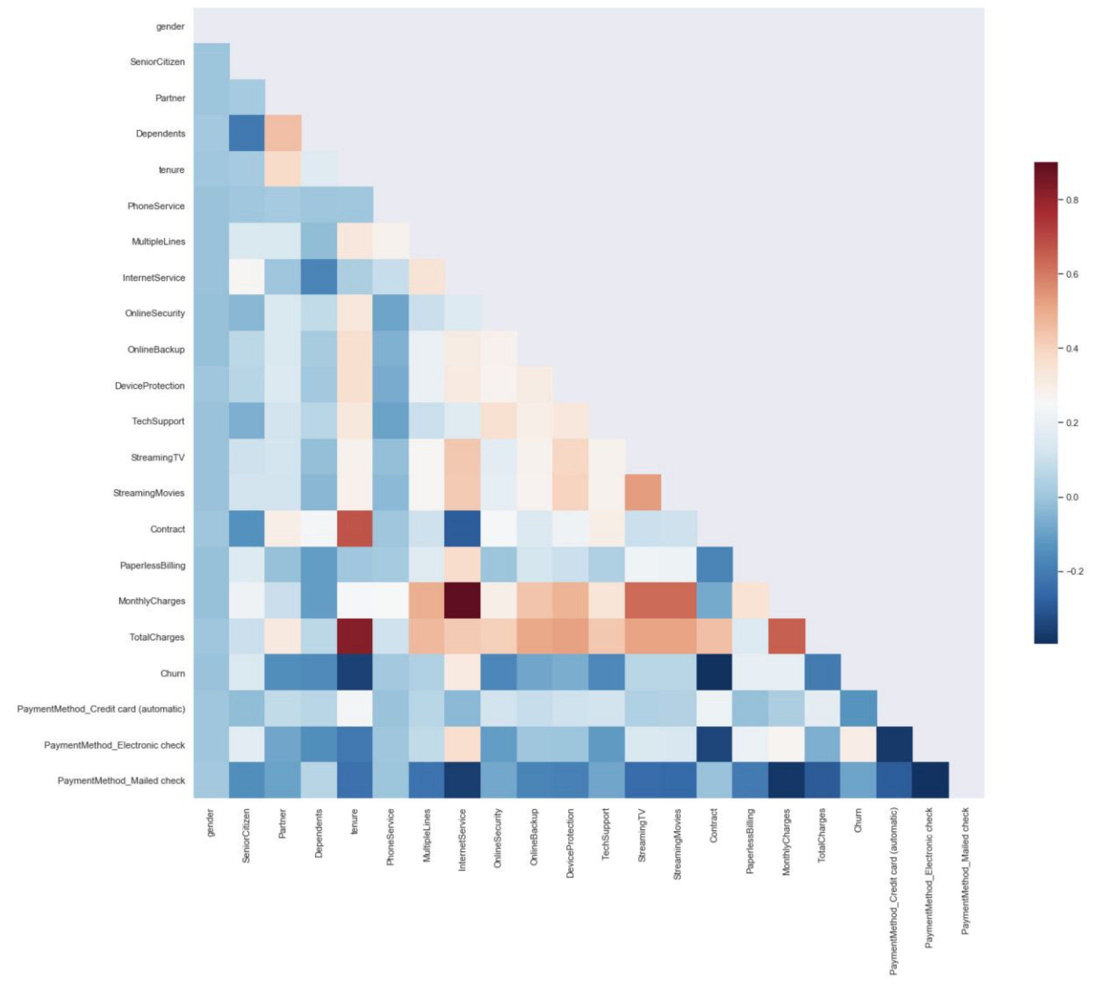


Multicollinearity Heat Map of Features

```python
# Find index of feature columns with correlation greater than 0.55
to_drop = [column for column in upper.columns if any(upper[column] > 0.55)
```

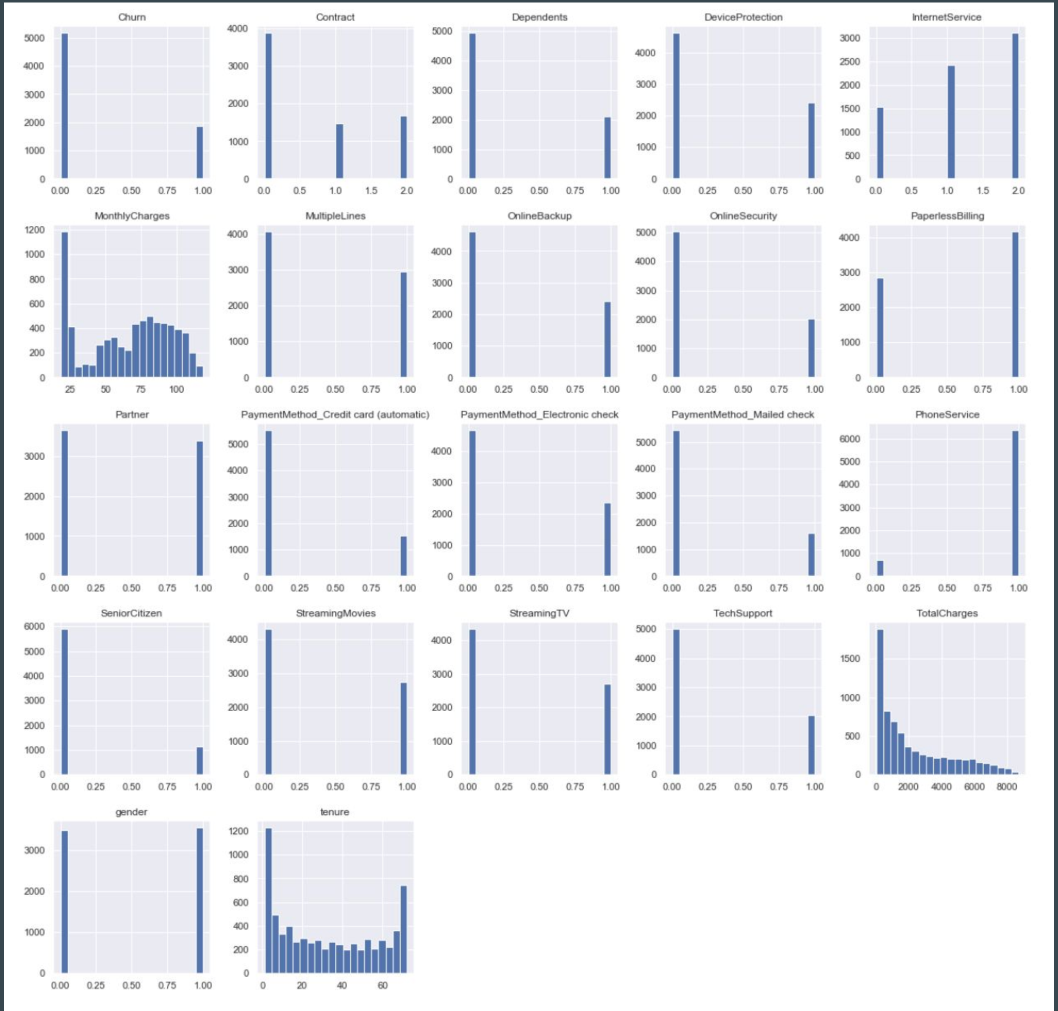
```python
to_drop
```

```
['Contract', 'MonthlyCharges', 'TotalCharges']
```



Histogram of Features

Large count of low values for:
Monthly Charges
Total Charges
Tenure



Histogram of Features

Low Values Show:
Phone Service = 1
& No Other Services

```
number = df[df['TotalCharges'] <= 24]
number
#Makes sense. Tenure is 1 and they only have phone service.
#tell story about Data.
```

nts	tenure	PhoneService	MultipleLines	InternetService	OnlineSecurity	OnlineBackup
0	1	1	0	0	0	0
0	1	1	0	0	0	0
0	1	1	0	0	0	0
1	1	1	0	0	0	0
0	1	1	0	0	0	0
1	1	1	0	0	0	0
1	1	1	0	0	0	0

```
number.Churn.value_counts()
#30% of phone service only came for one month?
```

```
0    104
1     58
Name: Churn, dtype: int64
```

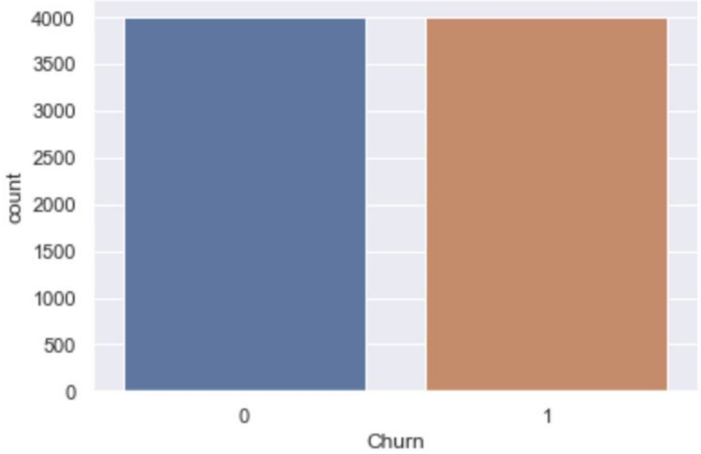
Treat for Class Imbalance

```
#upsample/downsample
#lost part of data from downsample. Results were not satisfactory.
from sklearn.utils import resample
no_churn = resample(no_churn, n_samples = 4000, random_state=1)
churn = resample(churn, n_samples = 4000, random_state=1)
max_sizes = no_churn['Churn'].value_counts().max()
max_sizes
```

```
churned = pd.concat([churn, no_churn])
```

```
sns.countplot('Churn', data=churned)
churned.Churn.value_counts()
```

```
1    4000
0    4000
Name: Churn, dtype: int64
```

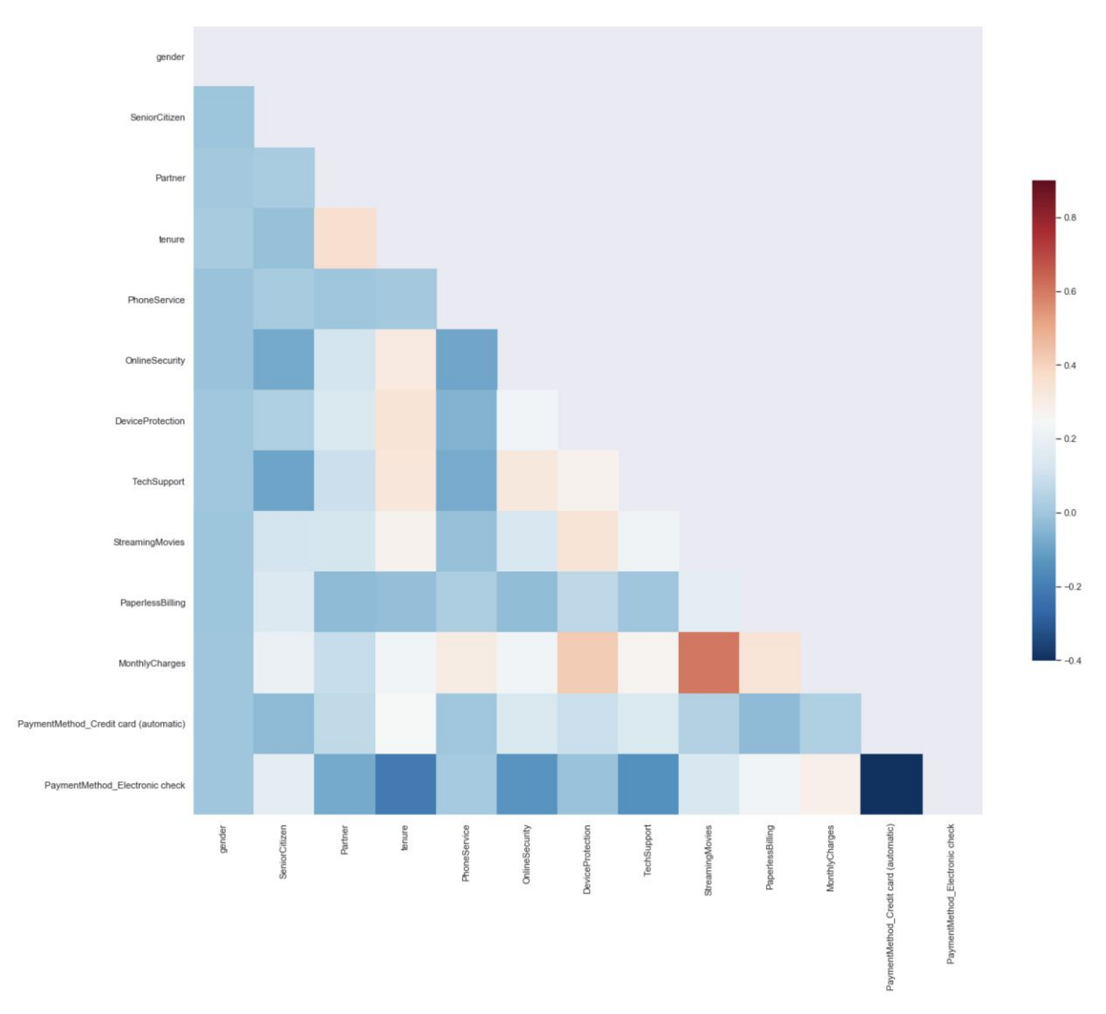


Drop Features With High Multicollinearity

```
# Split data into classes and training groups.
X = churned.drop(['customerID', 'TotalCharges', 'Churn', 'InternetService',
'PaymentMethod_Mailed check', 'PaymentMethod_Mailed check',
'Contract', 'Dependents','MultipleLines', 'OnlineBackup', 'StreamingTV'],
y = churned.Churn
X_train, X_test, y_train, y_test = train_test_split(X, y, random_state=1)
len(X_train)
```

```
6000
```

Multicollinearity Heat Map of Features After Highly Correlated Features Were Dropped



Models Accuracy Summary

```
Naive Bayes Unweighted Accuracy: 0.74 (+/- 0.02)
KNN Unweighted Accuracy: 0.83 (+/- 0.04)
Decision Tree Unweighted Accuracy: 0.91 (+/- 0.02)
Random ForestUnweighted Accuracy: 0.92 (+/- 0.02)
Logistic Regression Unweighted Accuracy: 0.76 (+/- 0.03)
SVM Unweighted Accuracy: 0.68 (+/- 0.14)
Gradient Boosting Unweighted Accuracy: 0.78 (+/- 0.03)
```

Apply Normalization to SVM

```
#stability increased here w/ Normalization (had impact on regression)
sv = LinearSVC()
sv.fit(X_train, y_train)
sv_predict = sv.predict(X_test)
#sv_predicts = sv.predict_proba(X_test)
sv_score = cross_val_score(sv, X, y, cv=10)
print('Cross_Val_Score: {}'.format(sv_score))
print("Unweighted Accuracy: %0.2f (+/- %0.2f)" % (sv_score.mean(), sv_score
```

```
Cross_Val_Score: [0.77     0.75     0.76625 0.74     0.74625 0.77     0.7875
0.75875 0.78625
 0.75125]
Unweighted Accuracy: 0.76 (+/- 0.03)
```

Continue Exploration:

- Gather age group data.
- Gather data from competing companies.
- Gather behavioral data.
- Are there various networks?
- Are there reliability issues?

Questions?