State Farm Project Model 2

July 29, 2019

1 State Farm Classification Project: Model 2

```
In [1]: import pandas as pd
        import numpy as np
        from sklearn.model_selection import cross_validate, train_test_split, GridSearchCV
        from sklearn.metrics import accuracy_score, confusion_matrix, classification_report
        %matplotlib inline
        import matplotlib.pyplot as plt
        from sklearn import metrics

In [3]: train = pd.read_csv("exercise_02_train.csv")

In [4]: train.head()

Out[4]:          x0         x1         x2        x3         x4         x5        x6  \
        0  0.198560  74.425320  67.627745 -3.095111  -6.822327  19.048071 -0.362378
        1 -29.662621  24.320711 -48.205182  1.430339  -6.552206   4.263074  6.551412
        2  15.493759 -66.160459  50.512903 -2.265792  14.428578   2.509323 -6.707536
        3 -19.837651  33.210943  53.405563  1.079462  11.364251  -1.064581  9.308857
        4  11.896655 -26.717872 -17.758176  1.692017  21.553537  -5.852097 -0.857435

                 x7         x8        x9 ...       x91       x92      x93       x94  \
        0 -10.699174 -22.699791 -1.561262 ...  0.800948  1.553846     asia -1.093926
        1   4.265483   1.245095  2.246814 ...  2.031707  7.544422     asia -3.659541
        2   3.820842 -11.100833 -1.459825 ... -0.992474  1.385799  america  1.299144
        3   9.266076  14.552959 -2.012755 ... -1.157845  6.036804     asia  0.521396
        4  -2.186940  18.075272 -1.404618 ... -3.045511 -1.719337     asia  1.526071

                x95        x96        x97        x98       x99  y
        0  16.202557  26.238591  -2.125570   9.644466  1.237667  0
        1  29.674259 -15.141647 -36.030599   5.820376  1.952183  1
        2  33.018090 -19.914894  26.212736   2.372690  0.558988  1
        3   9.664095 -27.197636  19.221130  13.382712  0.214462  0
        4 -25.608326  33.383803  -5.703269 -11.023730 -1.191319  0

        [5 rows x 101 columns]
```

In [5]:
```
def unique(list1):

    # insert the list to the set
    list_set = set(list1)
    # convert the set to the list
    unique_list = (list(list_set))
    for x in unique_list:
        print(x)

unique(train.dtypes)
```

```
int64
float64
object
```

In [6]:
```
predictCols = list(train)
predictCols.remove('y')
```

In [7]:
```
for col in predictCols:
    if train[col].dtype in [np.float64,np.int64]:
        train[col].fillna(train[col].mean(skipna = True), inplace=True)
```

In [8]:
```
# Ensure no remaining na's
numericCols = train.select_dtypes(include='number').columns
naVals = train[numericCols].isna().sum().sort_values()
naVals.sum()
```

Out[8]: 0

In [9]:
```
objectCols = train.select_dtypes(include='object').columns
print(objectCols)
```

```
Index(['x34', 'x35', 'x41', 'x45', 'x68', 'x93'], dtype='object')
```

In [10]:
```
train.x34.fillna(train.x34.mode()[0], inplace=True)
unique(train['x34'])
```

```
tesla
bmw
volkswagon
mercades
Honda
Toyota
ford
chevrolet
chrystler
nissan
```

In [11]:
```
train.x35.replace(['thurday', 'thur'], ['thursday','thursday'], inplace=True)
train.x35.replace(['wed'], ['wednesday'], inplace=True)
train.x35.replace(['fri'], ['friday'], inplace=True)
train.x35.fillna(train.x35.mode()[0], inplace=True)
unique(train['x35'])
```

```
monday
thursday
tuesday
wednesday
friday
```

In [12]:
```
# Convert currency column to float, remove nan's
train['x41'] = train['x41'].astype(str)
train['x41'] = train['x41'].map(lambda x: x.lstrip('$'))
train['x41'] = train['x41'].astype(np.float16)
train['x41'].fillna(0, inplace=True) # probably safer to replace nan's with 0, not me

print(train['x41'].isna().sum())
```

```
0
```

In [13]:
```
# Convert percentage column to float, remove nan's
train['x45'] = train['x45'].astype(str)
train['x45'] = train['x45'].map(lambda x: x.rstrip('%'))
train['x45'] = train['x45'].astype(np.float16)
train['x45'].fillna(train['x45'].mean(skipna = True), inplace=True) # since very few

print(train['x41'].isna().sum())
```

```
0
```

In [14]:
```
# Month Column
train.x68.replace(['Dev'], ['Dec'], inplace=True) # because I'm OCD
train.x68.replace(['sept.'], ['Sep'], inplace=True)
train.x68.replace(['January'], ['Jan'], inplace=True)
train.x68.replace(['July'], ['Jul'], inplace=True)
train.x68.fillna(train.x68.mode()[0], inplace=True)

unique(train['x68'])
```

```
Jun
Apr
Aug
Feb
Jul
```

```
Oct
Dec
Jan
Mar
Sep
Nov
May
```

```
In [15]: # Region
         train.x93.replace(['euorpe'], ['europe'], inplace=True)
         train = train[pd.isna(train['x93']) == False]
         print(train['x93'].isna().sum())
         # Region seems significant, and there's only 7 NA's, so remove rows with this as NA
```

```
0
```

```
In [16]: # Check if target has na's
         print(train['y'].isna().sum())
```

```
0
```

```
In [17]: train = pd.get_dummies(train)
```

```
In [18]: # Ensure we converted all non-numeric columns to numeric
         train.select_dtypes(include='object').columns
```

```
Out[18]: Index([], dtype='object')
```

```
In [19]: train.describe()
```

```
Out[19]:                  x0            x1            x2            x3            x4  \
         count  39993.000000  39993.000000  39993.000000  39993.000000  39993.000000
         mean       3.447752     -7.788416      1.704644     -0.072832      0.121980
         std       16.245334     37.012224     38.382930      1.503022     16.289301
         min      -60.113902   -157.341119   -163.339956     -6.276969    -61.632319
         25%       -7.595295    -32.731869    -24.141082     -1.087780    -10.896141
         50%        3.446322     -7.987507      1.959477     -0.062721      0.105307
         75%       14.266326     16.848201     27.511371      0.940330     11.076726
         max       75.311659    153.469221    154.051060      5.837559     65.949709

                         x5            x6            x7            x8            x9  \
         count  39993.000000  39993.000000  39993.000000  39993.000000  39993.000000
         mean      -0.607009      0.035852     -0.052430     -2.911144     -0.024524
         std       15.583132      9.040667      6.952184     13.148182      2.939696
         min      -62.808995    -35.060656    -26.736717    -53.735586    -11.497395
         25%      -11.181510     -6.089227     -4.746572    -11.722590     -2.003827
```

```
50%      -0.576660   0.044975  -0.037833  -2.940961  -0.054184
75%       9.954957   6.100325   4.636585   5.857648   1.954809
max      63.424046  45.053946  34.267792  66.936936  11.271939

           ...          x68_Jul       x68_Jun       x68_Mar       x68_May  \
count      ...     39993.000000  39993.000000  39993.000000  39993.000000
mean       ...         0.277199      0.231516      0.010777      0.119221
std        ...         0.447621      0.421806      0.103252      0.324052
min        ...         0.000000      0.000000      0.000000      0.000000
25%        ...         0.000000      0.000000      0.000000      0.000000
50%        ...         0.000000      0.000000      0.000000      0.000000
75%        ...         1.000000      0.000000      0.000000      0.000000
max        ...         1.000000      1.000000      1.000000      1.000000

            x68_Nov       x68_Oct       x68_Sep   x93_america      x93_asia  \
count  39993.000000  39993.000000  39993.000000  39993.000000  39993.000000
mean       0.003776      0.022604      0.087140      0.078289      0.885555
std        0.061331      0.148639      0.282044      0.268629      0.318355
min        0.000000      0.000000      0.000000      0.000000      0.000000
25%        0.000000      0.000000      0.000000      0.000000      1.000000
50%        0.000000      0.000000      0.000000      0.000000      1.000000
75%        0.000000      0.000000      0.000000      0.000000      1.000000
max        1.000000      1.000000      1.000000      1.000000      1.000000

         x93_europe
count  39993.000000
mean       0.036156
std        0.186681
min        0.000000
25%        0.000000
50%        0.000000
75%        0.000000
max        1.000000

[8 rows x 127 columns]
```

1.1 Now check class imbalance

```
In [20]: x = train.y.value_counts()
         print(x)

         print("% oF Training Set with Positives: " + "{0:.0%}".format(x[1] / (x[0] + x[1])))
         print("% oF Training Set with Negatives: " + "{0:.0%}".format(x[0] / (x[0] + x[1])))
```

```
0    31851
1     8142
Name: y, dtype: int64
% oF Training Set with Positives: 20%
```

```
% oF Training Set with Negatives: 80%
```

1.1.1 This class imbalance is not too bad, so we don't need to do resampling...

1.2 Now scale data to normalize

In [21]:
```python
from sklearn.preprocessing import StandardScaler

predictCols = list(train)
predictCols.remove('y')

train2 = train.copy()
for col in predictCols:
    scaler = StandardScaler()
    if train2[col].dtype != 'float64': # convert data type to avoid warning
        train2[col] = np.float64(train2[col])
    scaler.fit(train2[[col]])
    train2.loc[:,col] = scaler.transform(train2[[col]])
```

In [ ]:

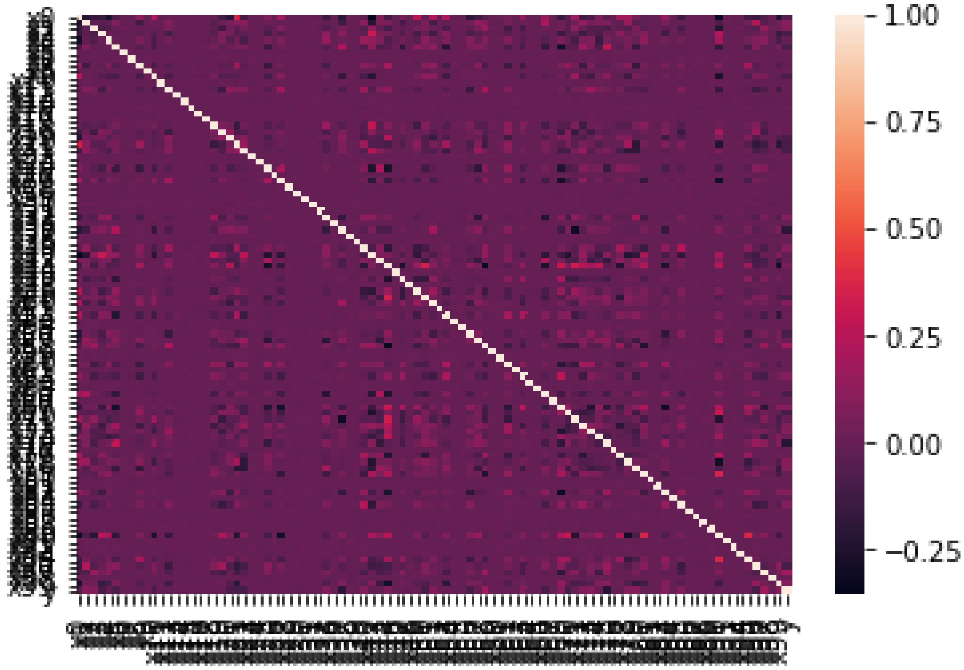
1.3 Now check correlation

In [22]:
```python
import seaborn as sns
%matplotlib inline

# calculate the correlation matrix
corr = train[numericCols].corr()

# plot the heatmap
sns.heatmap(corr,
        xticklabels=corr.columns,
        yticklabels=corr.columns)
```

Out[22]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x2194c03b3c8>
```




```
In [23]: # Attempt to find columns with high-correlation, and remove if necessary
         for col in numericCols:
             #print("\n\n" + col)
             q = corr[col].sort_values(ascending = False)
             q = q.drop(col)
             q = q[abs(q) >= 0.5]
             perfectNames = q.index.values
             if len(q) > 0:
                 print(col)
```

1.4 Clearly, no columns in the data set are highly correlated, so no need to remove.

```
In [24]: y = train2['y']
         train2 = train2.drop(['y'], axis=1)
```

```
In [ ]:
```

2 Now can use Support Vector Machine model

```
In [25]: from sklearn import svm
```

```
In [92]: # Train test split
         X_train, X_test, y_train, y_test = train_test_split(train2, y, test_size=.2, random_s
```

2.1 First try linear kernel, although we don't expect this problem to be linear

In [93]:
```
svclassifier = svm.SVC(kernel='linear', C = 1.0, max_iter = 10000)
svclassifier.fit(X_train, y_train)
```

```
C:\Python 3.7\lib\site-packages\sklearn\svm\base.py:244: ConvergenceWarning: Solver terminated
  % self.max_iter, ConvergenceWarning)
```

Out[93]:
```
SVC(C=1.0, cache_size=200, class_weight=None, coef0=0.0,
  decision_function_shape='ovr', degree=3, gamma='auto_deprecated',
  kernel='linear', max_iter=10000, probability=False, random_state=None,
  shrinking=True, tol=0.001, verbose=False)
```

In [94]:
```
pred_train = svclassifier.predict(X_train)
pred_test = svclassifier.predict(X_test)
```

In [95]:
```
# evaluate predictions
train_accuracy = accuracy_score(y_train, pred_train)
print("Train Accuracy: %.2f%%" % (train_accuracy * 100.0))

test_accuracy = accuracy_score(y_test, pred_test.round())
print("Test Accuracy: %.2f%%" % (test_accuracy * 100.0))
```

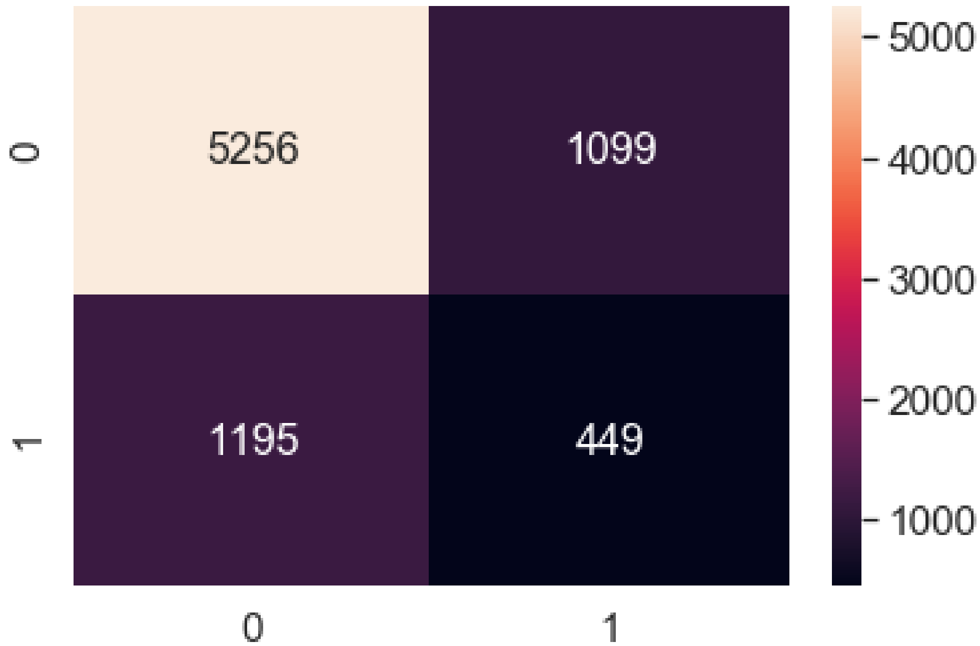
```
Train Accuracy: 71.74%
Test Accuracy: 71.32%
```

In [96]:
```
## Accuracy is not very good, but at least we're not over-fitting
```

In [97]:
```
cm = confusion_matrix(y_test, pred_test)

sns.set(font_scale=1.4)#for label size
sns.heatmap(cm, annot=True,fmt='g',annot_kws={"size": 16})# font size
```

Out[97]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x19431df2908>
```



2.1.1 Slightly more false negatives than false postives...probably due to the class imbalance 0's to 1's in the training set heavily weighted toward 0's (80%)

2.2 Now try the Radial Basis Function kernel

In [98]:
```
svclassifier = svm.SVC(kernel='rbf', C = 1.0, max_iter = 10000,gamma = 'auto')
svclassifier.fit(X_train, y_train)

pred_train = svclassifier.predict(X_train)
pred_test = svclassifier.predict(X_test)

test_accuracy = accuracy_score(y_test, pred_test)
print("Test Accuracy: %.2f%%" % (test_accuracy * 100.0))
```

```
C:\Python 3.7\lib\site-packages\sklearn\svm\base.py:244: ConvergenceWarning: Solver terminated
  % self.max_iter, ConvergenceWarning)


Test Accuracy: 98.35%
```

2.2.1 Massive improvement by switching to rbf kernel. Increased max_iter beyond 10,000 didn't have much effect on accuracy, but was much slower

2.3 Now use a grid search with 5-fold cross-validation to find the hyperparameters C and gamma

In [39]:
```
# Tune the two main hyperparameters
from sklearn.model_selection import GridSearchCV

def svc_param_selection(X, y, nfolds):
    Cs = [0.001, 0.01, 0.1, 1, 10]
    gammas = [0.001, 0.01, 0.1, 1]
    param_grid = {'C': Cs, 'gamma' : gammas}
    grid_search = GridSearchCV(svm.SVC(kernel='rbf',max_iter = 1000,gamma = 'auto'), 
    grid_search.fit(X, y)
    grid_search.best_params_
    return grid_search.best_params_

svc_param_selection(X_train, y_train, 5)
```

```
C:\Python 3.7\lib\site-packages\sklearn\svm\base.py:244: ConvergenceWarning: Solver terminated
  % self.max_iter, ConvergenceWarning)
C:\Python 3.7\lib\site-packages\sklearn\svm\base.py:244: ConvergenceWarning: Solver terminated
  % self.max_iter, ConvergenceWarning)
C:\Python 3.7\lib\site-packages\sklearn\svm\base.py:244: ConvergenceWarning: Solver terminated
  % self.max_iter, ConvergenceWarning)
C:\Python 3.7\lib\site-packages\sklearn\svm\base.py:244: ConvergenceWarning: Solver terminated
  % self.max_iter, ConvergenceWarning)
C:\Python 3.7\lib\site-packages\sklearn\svm\base.py:244: ConvergenceWarning: Solver terminated
  % self.max_iter, ConvergenceWarning)
C:\Python 3.7\lib\site-packages\sklearn\svm\base.py:244: ConvergenceWarning: Solver terminated
  % self.max_iter, ConvergenceWarning)
C:\Python 3.7\lib\site-packages\sklearn\svm\base.py:244: ConvergenceWarning: Solver terminated
  % self.max_iter, ConvergenceWarning)
C:\Python 3.7\lib\site-packages\sklearn\svm\base.py:244: ConvergenceWarning: Solver terminated
  % self.max_iter, ConvergenceWarning)
C:\Python 3.7\lib\site-packages\sklearn\svm\base.py:244: ConvergenceWarning: Solver terminated
  % self.max_iter, ConvergenceWarning)
C:\Python 3.7\lib\site-packages\sklearn\svm\base.py:244: ConvergenceWarning: Solver terminated
  % self.max_iter, ConvergenceWarning)
C:\Python 3.7\lib\site-packages\sklearn\svm\base.py:244: ConvergenceWarning: Solver terminated
  % self.max_iter, ConvergenceWarning)
C:\Python 3.7\lib\site-packages\sklearn\svm\base.py:244: ConvergenceWarning: Solver terminated
  % self.max_iter, ConvergenceWarning)
C:\Python 3.7\lib\site-packages\sklearn\svm\base.py:244: ConvergenceWarning: Solver terminated
  % self.max_iter, ConvergenceWarning)
C:\Python 3.7\lib\site-packages\sklearn\svm\base.py:244: ConvergenceWarning: Solver terminated
  % self.max_iter, ConvergenceWarning)
C:\Python 3.7\lib\site-packages\sklearn\svm\base.py:244: ConvergenceWarning: Solver terminated
```

```
  % self.max_iter, ConvergenceWarning)
C:\Python 3.7\lib\site-packages\sklearn\svm\base.py:244: ConvergenceWarning: Solver terminated
  % self.max_iter, ConvergenceWarning)
C:\Python 3.7\lib\site-packages\sklearn\svm\base.py:244: ConvergenceWarning: Solver terminated
  % self.max_iter, ConvergenceWarning)
C:\Python 3.7\lib\site-packages\sklearn\svm\base.py:244: ConvergenceWarning: Solver terminated
  % self.max_iter, ConvergenceWarning)
C:\Python 3.7\lib\site-packages\sklearn\svm\base.py:244: ConvergenceWarning: Solver terminated
  % self.max_iter, ConvergenceWarning)
C:\Python 3.7\lib\site-packages\sklearn\svm\base.py:244: ConvergenceWarning: Solver terminated
  % self.max_iter, ConvergenceWarning)
C:\Python 3.7\lib\site-packages\sklearn\svm\base.py:244: ConvergenceWarning: Solver terminated
  % self.max_iter, ConvergenceWarning)
C:\Python 3.7\lib\site-packages\sklearn\svm\base.py:244: ConvergenceWarning: Solver terminated
  % self.max_iter, ConvergenceWarning)
C:\Python 3.7\lib\site-packages\sklearn\svm\base.py:244: ConvergenceWarning: Solver terminated
  % self.max_iter, ConvergenceWarning)
C:\Python 3.7\lib\site-packages\sklearn\svm\base.py:244: ConvergenceWarning: Solver terminated
  % self.max_iter, ConvergenceWarning)
C:\Python 3.7\lib\site-packages\sklearn\svm\base.py:244: ConvergenceWarning: Solver terminated
  % self.max_iter, ConvergenceWarning)
C:\Python 3.7\lib\site-packages\sklearn\svm\base.py:244: ConvergenceWarning: Solver terminated
  % self.max_iter, ConvergenceWarning)
C:\Python 3.7\lib\site-packages\sklearn\svm\base.py:244: ConvergenceWarning: Solver terminated
  % self.max_iter, ConvergenceWarning)
C:\Python 3.7\lib\site-packages\sklearn\svm\base.py:244: ConvergenceWarning: Solver terminated
  % self.max_iter, ConvergenceWarning)
C:\Python 3.7\lib\site-packages\sklearn\svm\base.py:244: ConvergenceWarning: Solver terminated
  % self.max_iter, ConvergenceWarning)
C:\Python 3.7\lib\site-packages\sklearn\svm\base.py:244: ConvergenceWarning: Solver terminated
  % self.max_iter, ConvergenceWarning)
C:\Python 3.7\lib\site-packages\sklearn\svm\base.py:244: ConvergenceWarning: Solver terminated
  % self.max_iter, ConvergenceWarning)
C:\Python 3.7\lib\site-packages\sklearn\svm\base.py:244: ConvergenceWarning: Solver terminated
  % self.max_iter, ConvergenceWarning)
C:\Python 3.7\lib\site-packages\sklearn\svm\base.py:244: ConvergenceWarning: Solver terminated
  % self.max_iter, ConvergenceWarning)
C:\Python 3.7\lib\site-packages\sklearn\svm\base.py:244: ConvergenceWarning: Solver terminated
  % self.max_iter, ConvergenceWarning)
C:\Python 3.7\lib\site-packages\sklearn\svm\base.py:244: ConvergenceWarning: Solver terminated
  % self.max_iter, ConvergenceWarning)
C:\Python 3.7\lib\site-packages\sklearn\svm\base.py:244: ConvergenceWarning: Solver terminated
  % self.max_iter, ConvergenceWarning)
C:\Python 3.7\lib\site-packages\sklearn\svm\base.py:244: ConvergenceWarning: Solver terminated
  % self.max_iter, ConvergenceWarning)
C:\Python 3.7\lib\site-packages\sklearn\svm\base.py:244: ConvergenceWarning: Solver terminated
```

```
  % self.max_iter, ConvergenceWarning)
C:\Python 3.7\lib\site-packages\sklearn\svm\base.py:244: ConvergenceWarning: Solver terminated
  % self.max_iter, ConvergenceWarning)
C:\Python 3.7\lib\site-packages\sklearn\svm\base.py:244: ConvergenceWarning: Solver terminated
  % self.max_iter, ConvergenceWarning)
C:\Python 3.7\lib\site-packages\sklearn\svm\base.py:244: ConvergenceWarning: Solver terminated
  % self.max_iter, ConvergenceWarning)
C:\Python 3.7\lib\site-packages\sklearn\svm\base.py:244: ConvergenceWarning: Solver terminated
  % self.max_iter, ConvergenceWarning)
C:\Python 3.7\lib\site-packages\sklearn\svm\base.py:244: ConvergenceWarning: Solver terminated
  % self.max_iter, ConvergenceWarning)
C:\Python 3.7\lib\site-packages\sklearn\svm\base.py:244: ConvergenceWarning: Solver terminated
  % self.max_iter, ConvergenceWarning)
C:\Python 3.7\lib\site-packages\sklearn\svm\base.py:244: ConvergenceWarning: Solver terminated
  % self.max_iter, ConvergenceWarning)
C:\Python 3.7\lib\site-packages\sklearn\svm\base.py:244: ConvergenceWarning: Solver terminated
  % self.max_iter, ConvergenceWarning)
C:\Python 3.7\lib\site-packages\sklearn\svm\base.py:244: ConvergenceWarning: Solver terminated
  % self.max_iter, ConvergenceWarning)
C:\Python 3.7\lib\site-packages\sklearn\svm\base.py:244: ConvergenceWarning: Solver terminated
  % self.max_iter, ConvergenceWarning)
C:\Python 3.7\lib\site-packages\sklearn\svm\base.py:244: ConvergenceWarning: Solver terminated
  % self.max_iter, ConvergenceWarning)
C:\Python 3.7\lib\site-packages\sklearn\svm\base.py:244: ConvergenceWarning: Solver terminated
  % self.max_iter, ConvergenceWarning)
C:\Python 3.7\lib\site-packages\sklearn\svm\base.py:244: ConvergenceWarning: Solver terminated
  % self.max_iter, ConvergenceWarning)
C:\Python 3.7\lib\site-packages\sklearn\svm\base.py:244: ConvergenceWarning: Solver terminated
  % self.max_iter, ConvergenceWarning)
C:\Python 3.7\lib\site-packages\sklearn\svm\base.py:244: ConvergenceWarning: Solver terminated
  % self.max_iter, ConvergenceWarning)
C:\Python 3.7\lib\site-packages\sklearn\svm\base.py:244: ConvergenceWarning: Solver terminated
  % self.max_iter, ConvergenceWarning)
C:\Python 3.7\lib\site-packages\sklearn\svm\base.py:244: ConvergenceWarning: Solver terminated
  % self.max_iter, ConvergenceWarning)
C:\Python 3.7\lib\site-packages\sklearn\svm\base.py:244: ConvergenceWarning: Solver terminated
  % self.max_iter, ConvergenceWarning)
C:\Python 3.7\lib\site-packages\sklearn\svm\base.py:244: ConvergenceWarning: Solver terminated
  % self.max_iter, ConvergenceWarning)
C:\Python 3.7\lib\site-packages\sklearn\svm\base.py:244: ConvergenceWarning: Solver terminated
  % self.max_iter, ConvergenceWarning)
C:\Python 3.7\lib\site-packages\sklearn\svm\base.py:244: ConvergenceWarning: Solver terminated
  % self.max_iter, ConvergenceWarning)
C:\Python 3.7\lib\site-packages\sklearn\svm\base.py:244: ConvergenceWarning: Solver terminated
  % self.max_iter, ConvergenceWarning)
C:\Python 3.7\lib\site-packages\sklearn\svm\base.py:244: ConvergenceWarning: Solver terminated
```

```
  % self.max_iter, ConvergenceWarning)
C:\Python 3.7\lib\site-packages\sklearn\svm\base.py:244: ConvergenceWarning: Solver terminated
  % self.max_iter, ConvergenceWarning)
C:\Python 3.7\lib\site-packages\sklearn\svm\base.py:244: ConvergenceWarning: Solver terminated
  % self.max_iter, ConvergenceWarning)
C:\Python 3.7\lib\site-packages\sklearn\svm\base.py:244: ConvergenceWarning: Solver terminated
  % self.max_iter, ConvergenceWarning)
C:\Python 3.7\lib\site-packages\sklearn\svm\base.py:244: ConvergenceWarning: Solver terminated
  % self.max_iter, ConvergenceWarning)
C:\Python 3.7\lib\site-packages\sklearn\svm\base.py:244: ConvergenceWarning: Solver terminated
  % self.max_iter, ConvergenceWarning)
C:\Python 3.7\lib\site-packages\sklearn\svm\base.py:244: ConvergenceWarning: Solver terminated
  % self.max_iter, ConvergenceWarning)
C:\Python 3.7\lib\site-packages\sklearn\svm\base.py:244: ConvergenceWarning: Solver terminated
  % self.max_iter, ConvergenceWarning)
C:\Python 3.7\lib\site-packages\sklearn\svm\base.py:244: ConvergenceWarning: Solver terminated
  % self.max_iter, ConvergenceWarning)
C:\Python 3.7\lib\site-packages\sklearn\svm\base.py:244: ConvergenceWarning: Solver terminated
  % self.max_iter, ConvergenceWarning)
C:\Python 3.7\lib\site-packages\sklearn\svm\base.py:244: ConvergenceWarning: Solver terminated
  % self.max_iter, ConvergenceWarning)
C:\Python 3.7\lib\site-packages\sklearn\svm\base.py:244: ConvergenceWarning: Solver terminated
  % self.max_iter, ConvergenceWarning)
C:\Python 3.7\lib\site-packages\sklearn\svm\base.py:244: ConvergenceWarning: Solver terminated
  % self.max_iter, ConvergenceWarning)
C:\Python 3.7\lib\site-packages\sklearn\svm\base.py:244: ConvergenceWarning: Solver terminated
  % self.max_iter, ConvergenceWarning)
C:\Python 3.7\lib\site-packages\sklearn\svm\base.py:244: ConvergenceWarning: Solver terminated
  % self.max_iter, ConvergenceWarning)
C:\Python 3.7\lib\site-packages\sklearn\svm\base.py:244: ConvergenceWarning: Solver terminated
  % self.max_iter, ConvergenceWarning)
C:\Python 3.7\lib\site-packages\sklearn\svm\base.py:244: ConvergenceWarning: Solver terminated
  % self.max_iter, ConvergenceWarning)
C:\Python 3.7\lib\site-packages\sklearn\svm\base.py:244: ConvergenceWarning: Solver terminated
  % self.max_iter, ConvergenceWarning)
C:\Python 3.7\lib\site-packages\sklearn\svm\base.py:244: ConvergenceWarning: Solver terminated
  % self.max_iter, ConvergenceWarning)
C:\Python 3.7\lib\site-packages\sklearn\svm\base.py:244: ConvergenceWarning: Solver terminated
  % self.max_iter, ConvergenceWarning)
C:\Python 3.7\lib\site-packages\sklearn\svm\base.py:244: ConvergenceWarning: Solver terminated
  % self.max_iter, ConvergenceWarning)
C:\Python 3.7\lib\site-packages\sklearn\svm\base.py:244: ConvergenceWarning: Solver terminated
  % self.max_iter, ConvergenceWarning)
C:\Python 3.7\lib\site-packages\sklearn\svm\base.py:244: ConvergenceWarning: Solver terminated
  % self.max_iter, ConvergenceWarning)
C:\Python 3.7\lib\site-packages\sklearn\svm\base.py:244: ConvergenceWarning: Solver terminated
```

```
  % self.max_iter, ConvergenceWarning)
C:\Python 3.7\lib\site-packages\sklearn\svm\base.py:244: ConvergenceWarning: Solver terminated
  % self.max_iter, ConvergenceWarning)
C:\Python 3.7\lib\site-packages\sklearn\svm\base.py:244: ConvergenceWarning: Solver terminated
  % self.max_iter, ConvergenceWarning)
C:\Python 3.7\lib\site-packages\sklearn\svm\base.py:244: ConvergenceWarning: Solver terminated
  % self.max_iter, ConvergenceWarning)
C:\Python 3.7\lib\site-packages\sklearn\svm\base.py:244: ConvergenceWarning: Solver terminated
  % self.max_iter, ConvergenceWarning)
C:\Python 3.7\lib\site-packages\sklearn\svm\base.py:244: ConvergenceWarning: Solver terminated
  % self.max_iter, ConvergenceWarning)
C:\Python 3.7\lib\site-packages\sklearn\svm\base.py:244: ConvergenceWarning: Solver terminated
  % self.max_iter, ConvergenceWarning)
C:\Python 3.7\lib\site-packages\sklearn\svm\base.py:244: ConvergenceWarning: Solver terminated
  % self.max_iter, ConvergenceWarning)
C:\Python 3.7\lib\site-packages\sklearn\svm\base.py:244: ConvergenceWarning: Solver terminated
  % self.max_iter, ConvergenceWarning)
C:\Python 3.7\lib\site-packages\sklearn\svm\base.py:244: ConvergenceWarning: Solver terminated
  % self.max_iter, ConvergenceWarning)
C:\Python 3.7\lib\site-packages\sklearn\svm\base.py:244: ConvergenceWarning: Solver terminated
  % self.max_iter, ConvergenceWarning)
C:\Python 3.7\lib\site-packages\sklearn\svm\base.py:244: ConvergenceWarning: Solver terminated
  % self.max_iter, ConvergenceWarning)
C:\Python 3.7\lib\site-packages\sklearn\svm\base.py:244: ConvergenceWarning: Solver terminated
  % self.max_iter, ConvergenceWarning)
C:\Python 3.7\lib\site-packages\sklearn\svm\base.py:244: ConvergenceWarning: Solver terminated
  % self.max_iter, ConvergenceWarning)
C:\Python 3.7\lib\site-packages\sklearn\svm\base.py:244: ConvergenceWarning: Solver terminated
  % self.max_iter, ConvergenceWarning)
```

```
Out[39]: {'C': 1, 'gamma': 0.01}
```

```
In [ ]: # Output: {'C': 1, 'gamma': 0.01}
```

2.3.1 Note, when gamma of 0.01 is used instead of 'auto', the test accuracy decreases slightly. As a result, gamma is set to 'auto' in the final model

```
In [ ]:
```

2.4 Finally, pre-proccess the test set, and calculate the final prediction

```
In [26]: test = pd.read_csv("exercise_02_test.csv")
```

```
In [27]: len(test.index)
```

```
Out[27]: 10000
```

```
In [28]: predictCols = list(test)
         for col in predictCols:
             if test[col].dtype in [np.float64,np.int64]:
                 #print(col)
                 test[col].fillna(test[col].mean(skipna = True), inplace=True)

In [29]: test.x34.fillna(test.x34.mode()[0], inplace=True)

In [30]: # Day Column
         test.x35.replace(['thurday', 'thur'], ['thursday','thursday'], inplace=True)
         test.x35.replace(['wed'], ['wednesday'], inplace=True)
         test.x35.replace(['fri'], ['friday'], inplace=True)
         test.x35.fillna(test.x35.mode()[0], inplace=True)

In [31]: # Convert currency column to float, remove nan's
         test['x41'] = test['x41'].astype(str)
         test['x41'] = test['x41'].map(lambda x: x.lstrip('$'))
         test['x41'] = test['x41'].astype(np.float16)
         test['x41'].fillna(0, inplace=True) # probably safer to replace nan's with 0, not mea

In [32]: # Convert percentage column to float, remove nan's
         test['x45'] = test['x45'].astype(str)
         test['x45'] = test['x45'].map(lambda x: x.rstrip('%'))
         test['x45'] = test['x45'].astype(np.float16)
         test['x45'].fillna(train['x45'].mean(skipna = True), inplace=True) # since very few u

In [33]: # Month Column
         test.x68.replace(['Dev'], ['Dec'], inplace=True) # because I'm OCD
         test.x68.replace(['sept.'], ['Sep'], inplace=True)
         test.x68.replace(['January'], ['Jan'], inplace=True)
         test.x68.replace(['July'], ['Jul'], inplace=True)
         test.x68.fillna(test.x68.mode()[0], inplace=True)

In [34]: # Region
         test.x93.replace(['euorpe'], ['europe'], inplace=True)
         test.x93.fillna(test.x93.mode()[0], inplace=True)
         print(test['x93'].isna().sum())
         # Region seems significant, and there's only 7 NA's, so remove rows with this as NA

0

In [35]: test = pd.get_dummies(test)

In [36]: # Ensure all columns in test are also in train after the one-hot encoding
         any(elem in list(test) for elem in list(train))

Out[36]: True
```

In [38]:
```
# Scale the test data
for col in test.columns:
    scaler = StandardScaler()
    if test[col].dtype != 'float64': # convert data type to avoid warning
        test[col] = np.float64(test[col])
    scaler.fit(test[[col]])
    test.loc[:,col] = scaler.transform(test[[col]])
```

2.5 Now retrain the model with the final hyperparameters using the full training set

In [40]:
```
svclassifier = svm.SVC(kernel='rbf', C = 1.0,gamma = 'auto',probability = True)
svclassifier.fit(train2, y)
```

Out[40]:
```
SVC(C=1.0, cache_size=200, class_weight=None, coef0=0.0,
  decision_function_shape='ovr', degree=3, gamma='auto', kernel='rbf',
  max_iter=-1, probability=True, random_state=None, shrinking=True,
  tol=0.001, verbose=False)
```

2.6 Now generate the final test output

In [41]:
```
final_y = svclassifier.predict_proba(test) # return class probabilities
```

In [43]:
```
final_y = final_y[:,1] # return only the probability of the 1's class
```

In [44]:
```
np.savetxt("results2.csv", final_y, delimiter = ",")
```