Gulp. Clean this up.

In [49]:
```
import pandas as pd
import numpy as np
from pickle import load
from pickle import dump
pd.set_option("max_columns", None)
pd.set_option('max_rows', None)
from sklearn.model_selection import train_test_split
from sklearn.pipeline import make_pipeline
from sklearn.preprocessing import StandardScaler
from sklearn import svm
from sklearn.model_selection import GridSearchCV
from sklearn.model_selection import cross_val_predict
from sklearn.metrics import confusion_matrix
from imblearn.pipeline import Pipeline as imbpipe
from sklearn.preprocessing import MinMaxScaler
from imblearn.over_sampling import SMOTE
from sklearn.metrics import confusion_matrix, plot_confusion_matrix
from sklearn.metrics import accuracy_score, make_scorer, f1_score, precision_scor
```

In [2]:
```
from sklearn.metrics import roc_auc_score, accuracy_score, precision_score, recal
from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.svm import SVC
from sklearn.ensemble import RandomForestClassifier
from sklearn.discriminant_analysis import LinearDiscriminantAnalysis, QuadraticDi
from sklearn.cluster import KMeans
from sklearn.neighbors import KNeighborsClassifier
from sklearn.naive_bayes import GaussianNB
from sklearn.metrics import precision_recall_fscore_support
from sklearn.metrics import mean_squared_error as mse
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import cross_validate
from sklearn.model_selection import train_test_split
from sklearn.ensemble import ExtraTreesClassifier
from sklearn.feature_selection import SelectFromModel
from sklearn.svm import LinearSVC
from sklearn.ensemble import VotingClassifier
from sklearn.feature_selection import RFECV
from sklearn.pipeline import Pipeline
from pipelinehelper import PipelineHelper
from sklearn.preprocessing import MaxAbsScaler
```

FSM

In [3]:
```
fsm = load(open('df_all.pkl', 'rb'))
```

In [4]:
```
fsm = fsm.drop(['County_x', 'State_x', 'County_y', 'state', 'county', 'id'], axis
```

In [5]:
```
state_dummies = pd.get_dummies(fsm['State_y'], drop_first = True)
central_outlying = pd.get_dummies(fsm['central_outlying'], drop_first = True)
```

In [6]:
```
fsm = fsm.drop(['State_y', 'central_outlying'], axis = 1)
```

In [7]:
```
fsm = pd.concat([fsm, state_dummies, central_outlying], axis = 1)
```

In [8]:
```
fsm.Target = fsm.Target.map({'Trump': 0, 'Clinton': 1})
```

In [9]:
```
fsm_X = fsm.drop(['Target'], axis = 1)
fsm_y = fsm.Target
```

In [10]:
```
f1_scores = make_scorer(f1_score)
```

In [11]:
```
fsm_X_train, fsm_X_test, fsm_y_train, fsm_y_test = train_test_split(fsm_X, fsm_y,
```

In [12]:
```
param = {'svc__kernel': ['rbf', 'poly', 'linear']}
pipe2 = make_pipeline(StandardScaler(), svm.SVC())
grid = GridSearchCV(pipe2, param, scoring= f1_scores)
```

In [13]:
```
grid.fit(fsm_X_train, fsm_y_train)
```

Out[13]:
```
GridSearchCV(estimator=Pipeline(steps=[('standardscaler', StandardScaler()),
                                       ('svc', SVC())]),
             param_grid={'svc__kernel': ['rbf', 'poly', 'linear']},
             scoring=make_scorer(f1_score))
```

In [14]:
```
grid.best_score_
```

Out[14]:
```
0.6827086673959848
```

In [15]:
```
grid.best_params_
```

Out[15]:
```
{'svc__kernel': 'linear'}
```

In [16]:
```
estimator = make_pipeline(StandardScaler(), svm.SVC(kernel = 'linear'))
```

In [17]:
```
fsm_y_hat_train = cross_val_predict(estimator, fsm_X_train, fsm_y_train)
```

In [18]:
```
confusion_matrix(fsm_y_train, fsm_y_hat_train)
```

Out[18]:
```
array([[1969,   20],
       [ 166,  200]], dtype=int64)
```

Grid Search

Best: {'classifier__selected_model': ('svm', {'class_weight': 'balanced'}), 'scaler__selected_model': ('std', {'with_mean': False, 'with_std': False})}

Best F1 Score: 0.8725346478911151

So... I'm having a difficult time understanding how these scaler params are any different than... just not scaling? Fiddling with that in my next model.

In [19]:
```
df = load(open('df_all.pkl', 'rb'))
```

In [20]:
```
df = df.drop(['County_x', 'id', 'State_x', 'County_y', 'State_y', 'county'], axis
```

In [21]:
```
state_dummies = pd.get_dummies(df['state'], drop_first = True)
central_outlying = pd.get_dummies(df['central_outlying'], drop_first = True)
```

In [22]:
```
df = df.drop(['state', 'central_outlying'], axis = 1)
```

In [23]:
```
df = pd.concat([df, state_dummies, central_outlying], axis = 1)
```

In [24]:
```
df.Target = df.Target.map({'Trump': 0, 'Clinton': 1})
```

In [25]:
```
X = df.drop(['Target'], axis = 1)
y = df.Target
```

In [26]:
```
f1_scores = make_scorer(f1_score)
```

In [27]:
```
X_train, X_test, y_train, y_test = train_test_split(X, y, stratify = y, random_st
```

In [28]:
```
pipe = Pipeline([
    ('scaler', PipelineHelper([
        ('std', StandardScaler()),
        ('max', MaxAbsScaler()),
        ('minmax', MinMaxScaler())
    ])),
    ('classifier', PipelineHelper([
        ('svm', LinearSVC()),
        ('rf', RandomForestClassifier()),
        ('logreg', LogisticRegression()),
        ('dt', DecisionTreeClassifier())
    ])),
])
```

In [29]:
```
params = {
    'scaler__selected_model': pipe.named_steps['scaler'].generate({
        'std__with_mean': [True, False],
        'std__with_std': [True, False],
        'max__copy': [True],  # just for displaying
    }),
    'classifier__selected_model': pipe.named_steps['classifier'].generate({
#         'svm__C': [None, 1.0],
#         'svm__kernel': ['rbf', 'poly', 'linear'],
#         'svm__penalty': ['l1', 'l2'],
        'svm__class_weight': [None, 'balanced'],
        'rf__max_depth': [None, 5, 10, 30],
        'rf__class_weight': [None, 'balanced'],
        'rf__n_estimators': [100, 20],
        'logreg__penalty': [None, 'l1', 'l2', 'elasticnet'],
        'logreg__C': [0.1, 1.0],
        'logreg__class_weight': [None, 'balanced'],
        'logreg__solver': ['lbfgs', 'liblinear', 'sag', 'saga'],
        'dt__class_weight': [None, 'balanced']
    })
}
```

In [30]:
```
grid = GridSearchCV(pipe, params, scoring= f1_scores)
```

In [31]:
```
grid.fit(X_train, y_train)
```

```
  warnings.warn("Liblinear failed to converge, increase "
C:\Users\angie\anaconda3\envs\learn-env\lib\site-packages\sklearn\svm\_base.p
y:976: ConvergenceWarning: Liblinear failed to converge, increase the number
of iterations.
  warnings.warn("Liblinear failed to converge, increase "

C:\Users\angie\anaconda3\envs\learn-env\lib\site-packages\sklearn\svm\_base.p
y:976: ConvergenceWarning: Liblinear failed to converge, increase the number
of iterations.
  warnings.warn("Liblinear failed to converge, increase "
C:\Users\angie\anaconda3\envs\learn-env\lib\site-packages\sklearn\svm\_base.p
y:976: ConvergenceWarning: Liblinear failed to converge, increase the number
of iterations.
  warnings.warn("Liblinear failed to converge, increase "
C:\Users\angie\anaconda3\envs\learn-env\lib\site-packages\sklearn\svm\_base.p
y:976: ConvergenceWarning: Liblinear failed to converge, increase the number
of iterations.
  warnings.warn("Liblinear failed to converge, increase "
C:\Users\angie\anaconda3\envs\learn-env\lib\site-packages\sklearn\svm\_base.p
y:976: ConvergenceWarning: Liblinear failed to converge, increase the number
```

In [32]:
```
grid.best_score_
```

Out[32]:
```
0.8679769922229997
```

In [33]:
```
grid.best_params_
```

Out[33]:
```
{'classifier__selected_model': ('logreg',
  {'C': 0.1, 'class_weight': 'balanced', 'penalty': 'l2', 'solver': 'lbfgs'}),
 'scaler__selected_model': ('std', {'with_mean': False, 'with_std': False})}
```

In [34]:
```
estimator = make_pipeline(StandardScaler(with_mean = False, with_std = False), Li
```

In [35]:
```
y_hat_train = cross_val_predict(estimator, X_train, y_train)
```

```
C:\Users\angie\anaconda3\envs\learn-env\lib\site-packages\sklearn\svm\_base.py:
976: ConvergenceWarning: Liblinear failed to converge, increase the number of i
terations.
  warnings.warn("Liblinear failed to converge, increase "
C:\Users\angie\anaconda3\envs\learn-env\lib\site-packages\sklearn\svm\_base.py:
976: ConvergenceWarning: Liblinear failed to converge, increase the number of i
terations.
  warnings.warn("Liblinear failed to converge, increase "
C:\Users\angie\anaconda3\envs\learn-env\lib\site-packages\sklearn\svm\_base.py:
976: ConvergenceWarning: Liblinear failed to converge, increase the number of i
terations.
  warnings.warn("Liblinear failed to converge, increase "
C:\Users\angie\anaconda3\envs\learn-env\lib\site-packages\sklearn\svm\_base.py:
976: ConvergenceWarning: Liblinear failed to converge, increase the number of i
terations.
  warnings.warn("Liblinear failed to converge, increase "
C:\Users\angie\anaconda3\envs\learn-env\lib\site-packages\sklearn\svm\_base.py:
976: ConvergenceWarning: Liblinear failed to converge, increase the number of i
terations.
  warnings.warn("Liblinear failed to converge, increase "
```

In [36]:
```
confusion_matrix(y_train, y_hat_train)
```

Out[36]:
```
array([[1967,   22],
       [  53,  313]], dtype=int64)
```

Best Model So Far

In [37]:
```
df = load(open('df_all.pkl', 'rb'))
```

In [38]:
```
df = df.drop(['County_x', 'id', 'State_x', 'County_y', 'State_y', 'county'], axis
```

In [39]:
```
state_dummies = pd.get_dummies(df['state'], drop_first = True)
central_outlying = pd.get_dummies(df['central_outlying'], drop_first = True)
```

In [40]:
```
df = df.drop(['state', 'central_outlying'], axis = 1)
```

In [41]:
```
df = pd.concat([df, state_dummies, central_outlying], axis = 1)
```

In [42]:
```
df.Target = df.Target.map({'Trump': 0, 'Clinton': 1})
```

In [43]:
```
X = df.drop(['Target'], axis = 1)
y = df.Target
```

In [44]:
```
f1_scores = make_scorer(f1_score)
```

In [45]:
```
X_train, X_test, y_train, y_test = train_test_split(X, y, stratify = y, random_st
```

In [50]:
```
pipeline = imbpipe(steps = [
    ('sm', SMOTE()),
    ('ss', StandardScaler(with_mean = False, with_std = False)),
    ('linsvc', LinearSVC(class_weight = 'balanced'))
])
```

In [51]:
```
pipeline.fit(X_train, y_train)
```

```
C:\Users\angie\anaconda3\envs\learn-env\lib\site-packages\sklearn\svm\_base.py:
976: ConvergenceWarning: Liblinear failed to converge, increase the number of i
terations.
  warnings.warn("Liblinear failed to converge, increase "
```

Out[51]:
```
Pipeline(steps=[('sm', SMOTE()),
                ('ss', StandardScaler(with_mean=False, with_std=False)),
                ('linsvc', LinearSVC(class_weight='balanced'))])
```

In [52]:
```
results = cross_validate(pipeline, X_train, y_train, return_train_score = True, s
```

```
C:\Users\angie\anaconda3\envs\learn-env\lib\site-packages\sklearn\svm\_base.py:
976: ConvergenceWarning: Liblinear failed to converge, increase the number of i
terations.
  warnings.warn("Liblinear failed to converge, increase "
C:\Users\angie\anaconda3\envs\learn-env\lib\site-packages\sklearn\svm\_base.py:
976: ConvergenceWarning: Liblinear failed to converge, increase the number of i
terations.
  warnings.warn("Liblinear failed to converge, increase "
C:\Users\angie\anaconda3\envs\learn-env\lib\site-packages\sklearn\svm\_base.py:
976: ConvergenceWarning: Liblinear failed to converge, increase the number of i
terations.
  warnings.warn("Liblinear failed to converge, increase "
C:\Users\angie\anaconda3\envs\learn-env\lib\site-packages\sklearn\svm\_base.py:
976: ConvergenceWarning: Liblinear failed to converge, increase the number of i
terations.
  warnings.warn("Liblinear failed to converge, increase "
C:\Users\angie\anaconda3\envs\learn-env\lib\site-packages\sklearn\svm\_base.py:
976: ConvergenceWarning: Liblinear failed to converge, increase the number of i
terations.
  warnings.warn("Liblinear failed to converge, increase "
```

In [53]:
```
results['test_score'].mean()
```

Out[53]:
```
0.8756135291732597
```

In [54]:
```
results['train_score'].mean()
```

Out[54]:
```
0.8785950018179409
```

In [56]:
```
y_hat_train = cross_val_predict(pipeline, X_train, y_train)
```

```
C:\Users\angie\anaconda3\envs\learn-env\lib\site-packages\sklearn\svm\_base.py:
976: ConvergenceWarning: Liblinear failed to converge, increase the number of i
terations.
  warnings.warn("Liblinear failed to converge, increase "
C:\Users\angie\anaconda3\envs\learn-env\lib\site-packages\sklearn\svm\_base.py:
976: ConvergenceWarning: Liblinear failed to converge, increase the number of i
terations.
  warnings.warn("Liblinear failed to converge, increase "
C:\Users\angie\anaconda3\envs\learn-env\lib\site-packages\sklearn\svm\_base.py:
976: ConvergenceWarning: Liblinear failed to converge, increase the number of i
terations.
  warnings.warn("Liblinear failed to converge, increase "
C:\Users\angie\anaconda3\envs\learn-env\lib\site-packages\sklearn\svm\_base.py:
976: ConvergenceWarning: Liblinear failed to converge, increase the number of i
terations.
  warnings.warn("Liblinear failed to converge, increase "
C:\Users\angie\anaconda3\envs\learn-env\lib\site-packages\sklearn\svm\_base.py:
976: ConvergenceWarning: Liblinear failed to converge, increase the number of i
terations.
  warnings.warn("Liblinear failed to converge, increase "
```

In [57]:
```
confusion_matrix(y_train, y_hat_train)
```

Out[57]:
```
array([[1909,   80],
       [  19,  347]], dtype=int64)
```

In [ ]:

In [ ]:

In [ ]: