CS 418: Introduction to Data Science
Project 02: Regression, Classification and Clustering
Fall 2019

Project Report

Task 1 : (5 pts.) Partition the merged dataset into a training set and a validation set using the holdout method or the cross-validation method. *How did you partition the dataset?*

Answer : The data is split using the holdout method and 25% of the data set is used for the test set while the other 75% of the data is used for the training set.

Task 2 : (5 pts.) Standardize the training set and the validation set.

Answer : The data is standardized by fitting it using the training data and the data was transformed using the training and test sets.

Task 3 : (25 pts.) Build a linear regression model to predict the number of votes cast for the Democratic party in each county. Consider multiple combinations of predictor variables. Compute evaluation metrics for the validation set and report your results. *What is the best performing linear regression model? What is the performance of the model? How did you select the variables of the model?*

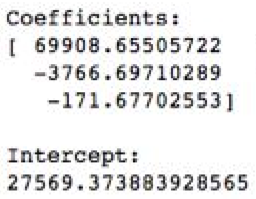
- Repeat this task for the number of votes cast for the Republican party in each county.

Answer :

Validation Set Democratic	R squared	Adjusted R squared	Root Mean Square
Linear Regression	0.885792	0.882641	13691.51523
RIDGE Regression	0.885824	0.882674	13689.58144
LASSO Regression	0.885827	0.882678	13689.38097

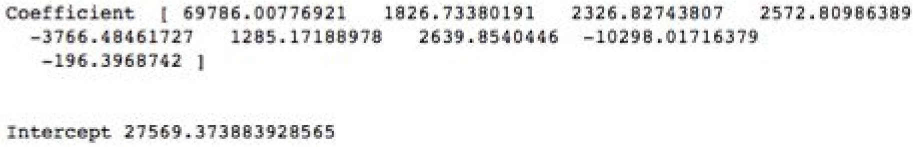
Validation Set Republican	R squared	Adjusted R squared	Root Mean Square
Linear Regression	0.692376	0.683890	16175.406414
RIDGE Regression	0.692486	0.684003	16172.507693
LASSO Regression	0.692389	0.683903	16175.049588

Linear regression **democratic** party coefficients and intercept for the training set



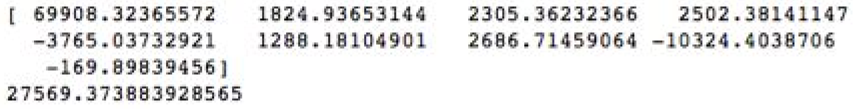
```
Coefficients:
[ 69908.65505722   1830.37860438   2307.30051784   2503.35963873
  -3766.69710289   1289.34667978   2692.06881351 -10326.83277279
   -171.67702553]

Intercept:
27569.373883928565
```

Ridge regression **democratic** party coefficients and intercept for the training set



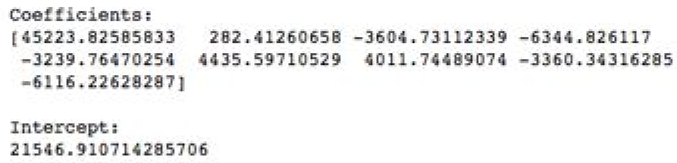
```
Coefficient  [ 69786.00776921   1826.73380191   2326.82743807   2572.80986389
  -3766.48461727   1285.17188978   2639.8540446  -10298.01716379
   -196.3968742 ]


Intercept 27569.373883928565
```

Lasso regression **democratic** party coefficients and intercept for the training set



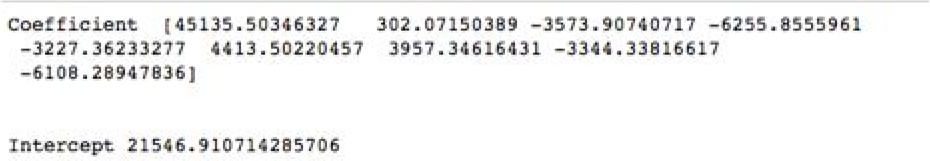
```
[ 69908.32365572   1824.93653144   2305.36232366   2502.38141147
  -3765.03732921   1288.18104901   2686.71459064 -10324.4038706
   -169.89839456]
27569.373883928565
```

Linear regression **republican** party coefficients and intercept for the training set



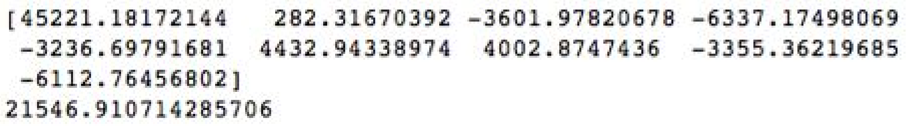
```
Coefficients:
[45223.82585833   282.41260658 -3604.73112339 -6344.826117
 -3239.76470254  4435.59710529  4011.74489074 -3360.34316285
 -6116.22628287]

Intercept:
21546.910714285706
```

Ridge regression **republican** party coefficients and intercept for the training set



```
Coefficient  [45135.50346327   302.07150389 -3573.90740717 -6255.8555961
 -3227.36233277  4413.50220457  3957.34616431 -3344.33816617
 -6108.28947836]


Intercept 21546.910714285706
```

Lasso regression **republican** party coefficients and intercept for the training set



```
[45221.18172144   282.31670392 -3601.97820678 -6337.17498069
 -3236.69791681  4432.94338974  4002.8747436  -3355.36219685
 -6112.76456802]
21546.910714285706
```

Three regression models were performed: linear, ridge, and lasso. The adjusted R squared values for the three models are listed in the table above. For the republican the best model according to the adjusted r squared is the ridge regression and for the democratic values the best model according to the adjusted r squared value is the lasso. Similarly for republican values the best model according to the root mean square is the ridge regression and for democrats the best model is the lasso regression.

The r squared value measures what proportion of the variance in the response variable is explained by the model. A higher r squared correlates to a better model. When there are multiple predictor variables used, however, the adjusted r squared value must be used. This is done so that the value of the predictor variables can be taken into account. If a predictor variable does not improve the model, I will result in a more complex model which can lead to overfitting. The adjusted R square penalizes any variables that are unnecessarily added and do not improve the model.

The variables picked for the models were based on project 1. The box plot of each variable was analyzed to see which variables had the biggest difference in terms of the mean, median, 1st and 3rd quartile between democrats and republicans. The following predictor variables were used: FIPS, total population, percent white, percent black, percent foreign born, percent 29 year or younger, median household income, percent less than high school degree, percent less than a bachelor's degree, and percent rural. All these variables has a big enough difference in the republican and democratic party, as observed in the box plots, to allow for a more accurate prediction of the validation set. Other variables such as percent female and percent Hispanic or Latino were very nearly equal in both the parties and were thus not included as a predictor variable.

Task 4 : (25 pts.) Build a classification model to classify each county as Democratic or Republican. Consider at least two different classification techniques with multiple combinations of parameters and multiple combinations of variables. Compute evaluation metrics for the validation set and report your results. *What is the best performing classification model? What is the performance of the model? How did you select the parameters of the model? How did you select the variables of the model?*

Answer :

We have considered four classifiers namely Decision Tree, K-Nearest Neighbors, Super Vector Machines and Random Forest with different combinations of parameters and different combinations of variables.

Selecting parameters :

For selecting the parameters, we have used Pipeline which takes StandardScalar and the classifier as input parameters and performs scaling and builds a model. We have also used GridSearchCV which takes the Classifier and the possible parameters of classifier as input, and searches the entire grid using cross validation and retrieves the best set of parameter values for which we get the best score on the training set.

Selecting variables:

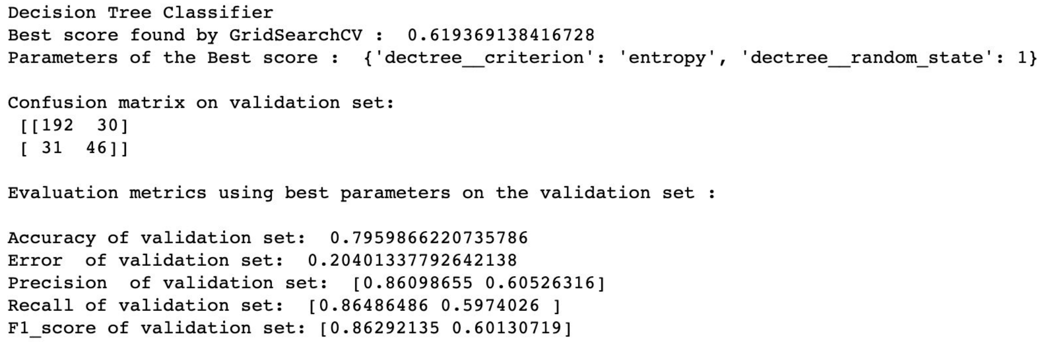
For selecting the variables, we have selected the variables which we identified as important in Project1 and tried different combinations of those variables to get the best set of variables which give the best score for the classifier.

Evaluations on different Classifiers :

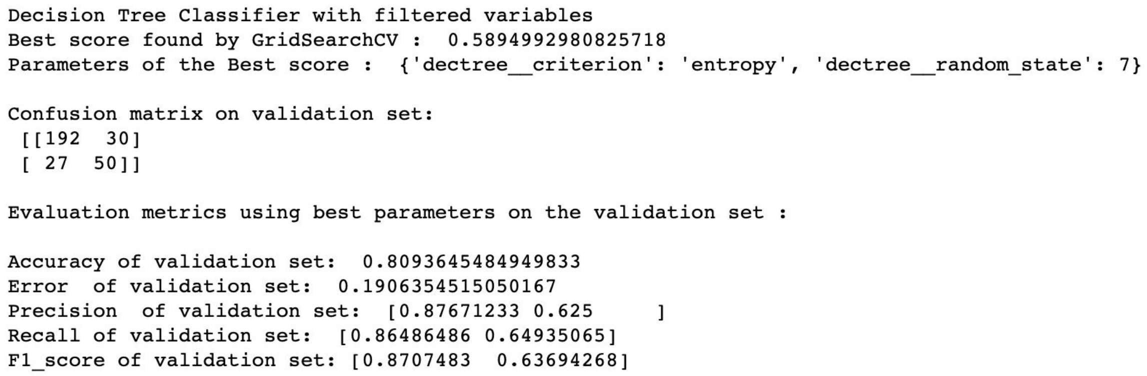
We have considered both accuracy and F1 score to identify the best classifier.

1. Decision Tree Classifier : We considered the decision tree classifier with all the variables and also filtering few variables with parameters like criterion and random_state.

   We got the best score for criterion = ‘entropy’ and random_state = 1 and for the variables 'Total Population', 'Percent White, not Hispanic or Latino', 'Percent Black, not Hispanic or Latino', 'Percent Hispanic or Latino', 'Percent Foreign Born', 'Percent Female', 'Median Household Income', 'Percent Unemployed', 'Percent Less than High School Degree', 'Percent Less than Bachelor\ 's Degree', 'Percent Rural'.



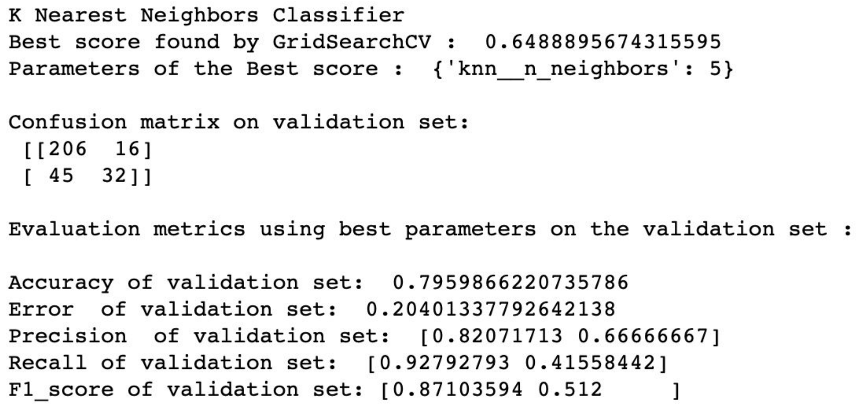
```
Decision Tree Classifier
Best score found by GridSearchCV :  0.619369138416728
Parameters of the Best score :  {'dectree__criterion': 'entropy', 'dectree__random_state': 1}

Confusion matrix on validation set:
 [[192  30]
 [ 31  46]]

Evaluation metrics using best parameters on the validation set :

Accuracy of validation set:  0.7959866220735786
Error  of validation set:  0.20401337792642138
Precision  of validation set:  [0.86098655 0.60526316]
Recall of validation set:  [0.86486486 0.5974026 ]
F1_score of validation set: [0.86292135 0.60130719]
```



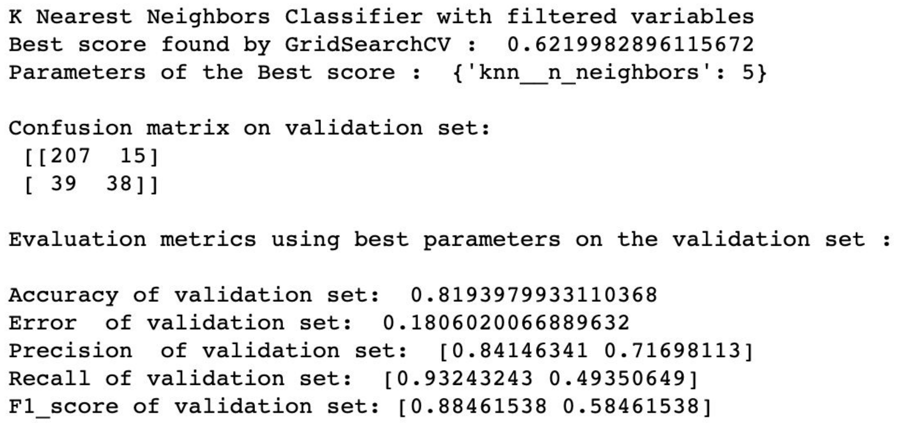
```
Decision Tree Classifier with filtered variables
Best score found by GridSearchCV :  0.5894992980825718
Parameters of the Best score :  {'dectree__criterion': 'entropy', 'dectree__random_state': 7}

Confusion matrix on validation set:
 [[192  30]
 [ 27  50]]

Evaluation metrics using best parameters on the validation set :

Accuracy of validation set:  0.8093645484949833
Error  of validation set:  0.1906354515050167
Precision  of validation set:  [0.87671233 0.625     ]
Recall of validation set:  [0.86486486 0.64935065]
F1_score of validation set: [0.8707483  0.63694268]
```

2. K Nearest Neighbors Classifier : We considered the decision K Nearest Neighbors Classifier with all the variables and also filtering few variables with parameters like n_neighbours.

   We got the best score for n_neighbours = 3 and for the variables 'Total Population', 'Percent Black, not Hispanic or Latino', 'Percent Hispanic or Latino', 'Percent Foreign Born', 'Percent Female', 'Percent Age 65 and Older', 'Median Household Income', 'Percent Unemployed', 'Percent Less than High School Degree', 'Percent Less than Bachelor\ 's Degree', 'Percent Rural'.



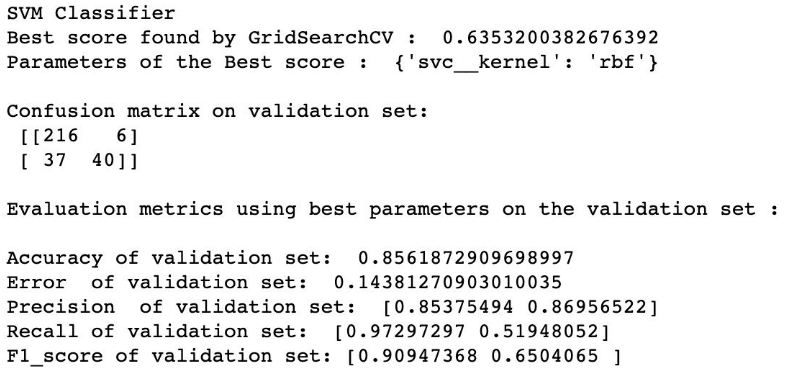
```
K Nearest Neighbors Classifier
Best score found by GridSearchCV :  0.6488895674315595
Parameters of the Best score :  {'knn__n_neighbors': 5}

Confusion matrix on validation set:
 [[206  16]
 [ 45  32]]

Evaluation metrics using best parameters on the validation set :

Accuracy of validation set:  0.7959866220735786
Error  of validation set:  0.20401337792642138
Precision  of validation set:  [0.82071713 0.66666667]
Recall of validation set:  [0.92792793 0.41558442]
F1_score of validation set: [0.87103594 0.512     ]
```



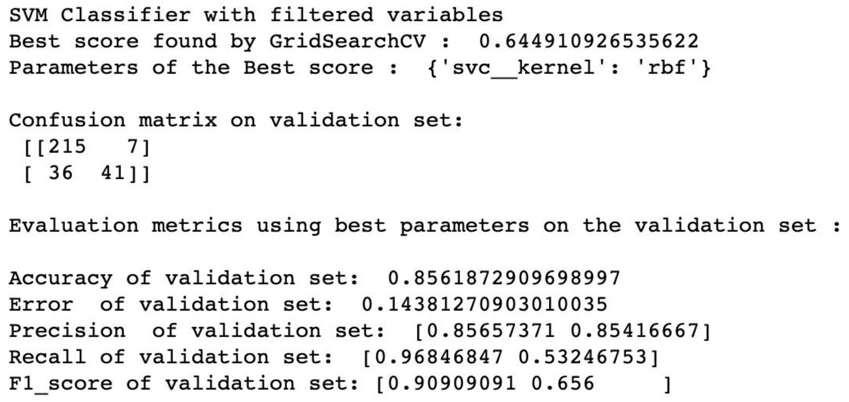
```
K Nearest Neighbors Classifier with filtered variables
Best score found by GridSearchCV :  0.6219982896115672
Parameters of the Best score :  {'knn__n_neighbors': 5}

Confusion matrix on validation set:
 [[207  15]
 [ 39  38]]

Evaluation metrics using best parameters on the validation set :

Accuracy of validation set:  0.8193979933110368
Error  of validation set:  0.1806020066889632
Precision  of validation set:  [0.84146341 0.71698113]
Recall of validation set:  [0.93243243 0.49350649]
F1_score of validation set: [0.88461538 0.58461538]
```

3. Super vector Classifier : We considered the decision Super vector classifier with all the variables and also filtering few variables with parameters like kernel.

   We got the best score for kernel = rbf and for the variables 'Total Population', 'Percent White, not Hispanic or Latino', 'Percent Black, not Hispanic or Latino', 'Percent Hispanic or Latino', 'Percent Foreign Born', 'Percent Female', 'Percent Age 29 and Under', 'Percent Age 65 and Older', 'Median Household Income', 'Percent Unemployed', 'Percent Less than High School Degree', 'Percent Less than Bachelor\ 's Degree', 'Percent Rural'.



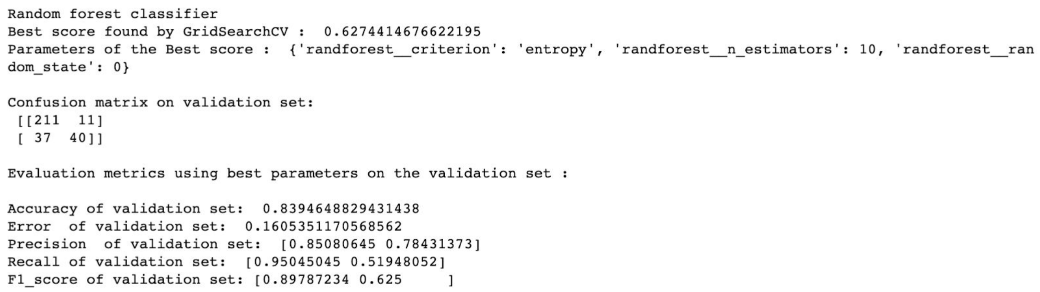
```
SVM Classifier
Best score found by GridSearchCV :  0.6353200382676392
Parameters of the Best score :  {'svc__kernel': 'rbf'}

Confusion matrix on validation set:
 [[216   6]
 [ 37  40]]

Evaluation metrics using best parameters on the validation set :

Accuracy of validation set:  0.8561872909698997
Error  of validation set:  0.14381270903010035
Precision  of validation set:  [0.85375494 0.86956522]
Recall of validation set:  [0.97297297 0.51948052]
F1_score of validation set: [0.90947368 0.6504065 ]
```



```
SVM Classifier with filtered variables
Best score found by GridSearchCV :  0.644910926535622
Parameters of the Best score :  {'svc__kernel': 'rbf'}

Confusion matrix on validation set:
 [[215   7]
 [ 36  41]]

Evaluation metrics using best parameters on the validation set :

Accuracy of validation set:  0.8561872909698997
Error  of validation set:  0.14381270903010035
Precision  of validation set:  [0.85657371 0.85416667]
Recall of validation set:  [0.96846847 0.53246753]
F1_score of validation set: [0.90909091 0.656     ]
```

4. Random Forest : We considered the decision tree classifier with all the variables and also filtering few variables with parameters like n_estimators, criterion and random_state.

   We got the best score for n_estimators = 10, criterion= ‘entropy’ and random_state = 0 and for the variables 'Total Population', 'Percent White, not Hispanic or Latino', 'Percent Black, not Hispanic or Latino', 'Percent Hispanic or Latino', 'Percent Foreign Born', 'Percent Female', 'Percent Age 29 and Under', 'Percent Age 65 and Older', 'Median Household Income', 'Percent Less than High School Degree', 'Percent Less than Bachelor\ 's Degree', 'Percent Rural'.



```
Random forest classifier
Best score found by GridSearchCV :  0.6274414676622195
Parameters of the Best score :  {'randforest__criterion': 'entropy', 'randforest__n_estimators': 10, 'randforest__ran
dom_state': 0}

Confusion matrix on validation set:
 [[211  11]
 [ 37  40]]

Evaluation metrics using best parameters on the validation set :

Accuracy of validation set:  0.8394648829431438
Error  of validation set:  0.1605351170568562
Precision  of validation set:  [0.85080645 0.78431373]
Recall of validation set:  [0.95045045 0.51948052]
F1_score of validation set: [0.89787234 0.625     ]
```

Out of all the classifiers tried, we got the best performance (best accuracy and F1 score) for SVC with kernel = ‘rbf’ and variables 'Total Population', 'Percent White, not Hispanic or Latino', 'Percent Black, not Hispanic or Latino', 'Percent Hispanic or Latino', 'Percent Foreign Born', 'Percent Female', 'Percent Age 29 and Under', 'Percent Age 65 and Older', 'Median Household Income', 'Percent Unemployed', 'Percent Less than High School Degree', 'Percent Less than Bachelor\ 's Degree', 'Percent Rural'.

Task 5 : (25 pts.) Build a clustering model to cluster the counties. Consider at least two different clustering techniques with multiple combinations of parameters and multiple combinations of variables. Compute unsupervised and supervised evaluation metrics for the validation set with the party of the counties (Democratic or Republican) as the true cluster and report your results. *What is the best performing clustering model? What is the performance of the model? How did you select the parameters of model? How did you select the variables of the model?*

Answer :

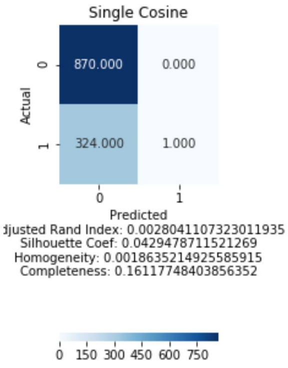
<u>Clustering Models Observed:</u> Hierarchical Single Linkage, Hierarchical Complete Linkage, Hierarchical Average Linkage, KMeans, DBSCAN

<u>Evaluation Metrics Calculated:</u> Silhouette Coefficient, Completeness, Homogeneity, Adjusted Rand Index

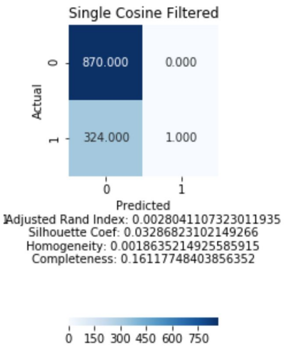
<u>Features:</u> For all clustering models, the set of all demographic features were considered when building the models as well as models with a set of filtered selection of variables which remained static for all models.

1. Hierarchical Single Linkage, Hierarchical Complete Linkage, Hierarchical Average Linkage:

   For these models, four distance formulas were considered: Euclidean, Manhattan , Minkowski and Cosine. The maximum cluster criterion was also common for all models. The adjusted rand index for single linkage was by far the worst for all distance metrics and variable groups.

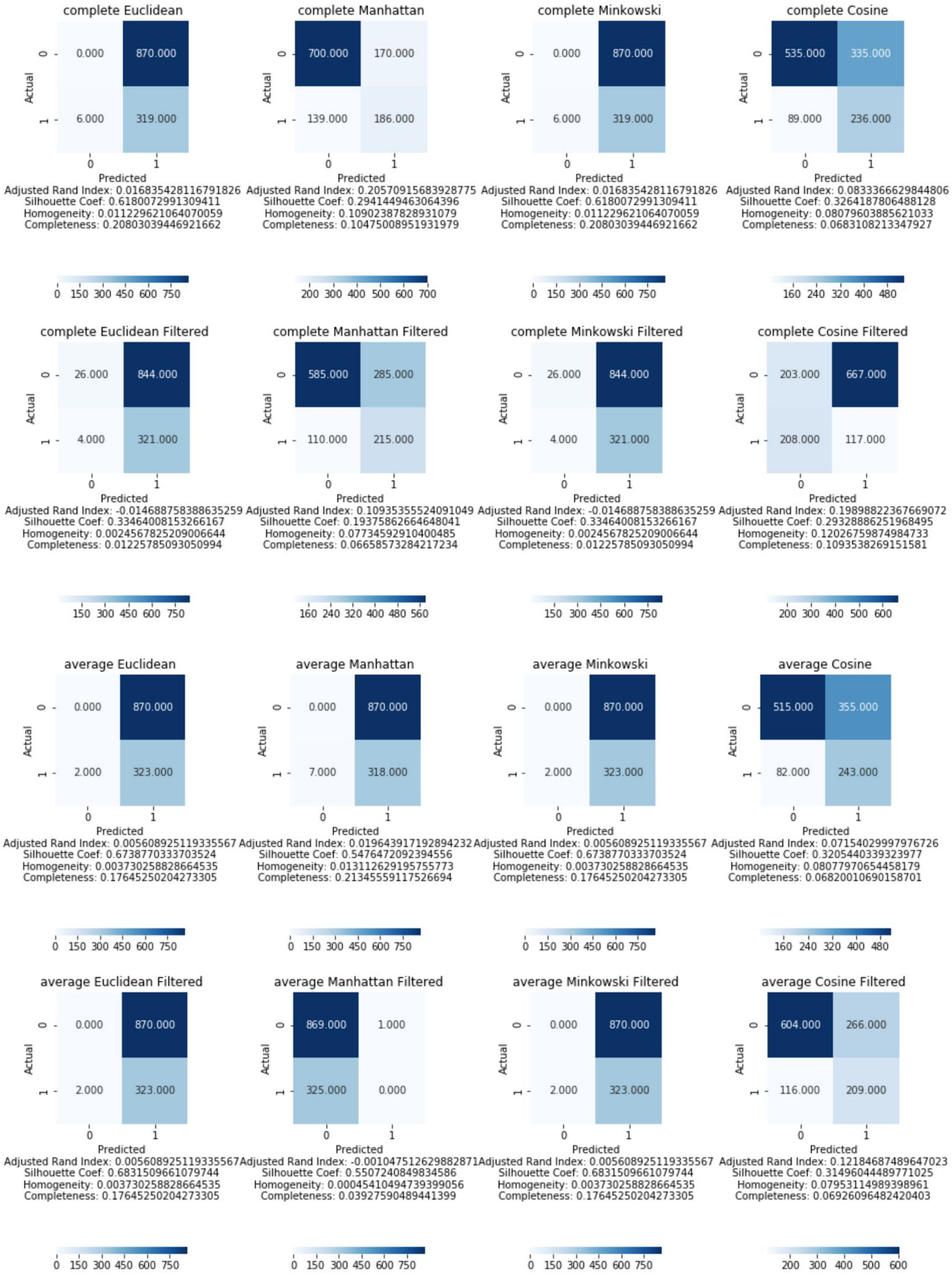


djusted Rand Index: 0.0028041107323011935
Silhouette Coef: 0.0429478711521269
Homogeneity: 0.0018635214925585915
Completeness: 0.16117748403856352



Adjusted Rand Index: 0.0028041107323011935
Silhouette Coef: 0.03286823102149266
Homogeneity: 0.0018635214925585915
Completeness: 0.16117748403856352

For complete linkage, Manhattan distance on all the variables generated the best Rand Index. Cosine distance performed the best for both silhouette and adj. rand index on average with complete linkage. This model benefited from the filtered set of variables. Cosine distance improves the performance given the high dimensionality of the dataset. The same is true for average linkage method.



complete Euclidean

```
Adjusted Rand Index: 0.016835428116791826
Silhouette Coef: 0.6180072991309411
Homogeneity: 0.011229621064070059
Completeness: 0.20803039446921662
```

complete Manhattan

```
Adjusted Rand Index: 0.20570915683928775
Silhouette Coef: 0.2941449463064396
Homogeneity: 0.10902387828931079
Completeness: 0.10475008951931979
```

complete Minkowski

```
Adjusted Rand Index: 0.016835428116791826
Silhouette Coef: 0.6180072991309411
Homogeneity: 0.011229621064070059
Completeness: 0.20803039446921662
```

complete Cosine

```
Adjusted Rand Index: 0.0833366629844806
Silhouette Coef: 0.3264187806488128
Homogeneity: 0.08079603885621033
Completeness: 0.0683108213347927
```

complete Euclidean Filtered

```
Adjusted Rand Index: -0.014688758388635259
Silhouette Coef: 0.33464008153266167
Homogeneity: 0.0024567825209006644
Completeness: 0.01225785093050994
```

complete Manhattan Filtered

```
Adjusted Rand Index: 0.10935355524091049
Silhouette Coef: 0.19375862664648041
Homogeneity: 0.07734592910400485
Completeness: 0.06658573284217234
```

complete Minkowski Filtered

```
Adjusted Rand Index: -0.014688758388635259
Silhouette Coef: 0.33464008153266167
Homogeneity: 0.0024567825209006644
Completeness: 0.01225785093050994
```

complete Cosine Filtered

```
Adjusted Rand Index: 0.19898822367669072
Silhouette Coef: 0.29328886251968495
Homogeneity: 0.12026759874984733
Completeness: 0.1093538269151581
```

average Euclidean

```
Adjusted Rand Index: 0.005608925119335567
Silhouette Coef: 0.6738770333703524
Homogeneity: 0.003730258828664535
Completeness: 0.17645250204273305
```

average Manhattan

```
Adjusted Rand Index: 0.019643917192894232
Silhouette Coef: 0.5476472092394556
Homogeneity: 0.013112629195755773
Completeness: 0.21345559117526694
```

average Minkowski

```
Adjusted Rand Index: 0.005608925119335567
Silhouette Coef: 0.6738770333703524
Homogeneity: 0.003730258828664535
Completeness: 0.17645250204273305
```

average Cosine

```
Adjusted Rand Index: 0.07154029997976726
Silhouette Coef: 0.3205440339323977
Homogeneity: 0.08077970654458179
Completeness: 0.06820010690158701
```

average Euclidean Filtered

```
Adjusted Rand Index: 0.005608925119335567
Silhouette Coef: 0.6831509661079744
Homogeneity: 0.003730258828664535
Completeness: 0.17645250204273305
```

average Manhattan Filtered

```
Adjusted Rand Index: -0.0010475126298828871
Silhouette Coef: 0.5507240849834586
Homogeneity: 0.00045410494739399056
Completeness: 0.03927590489441399
```

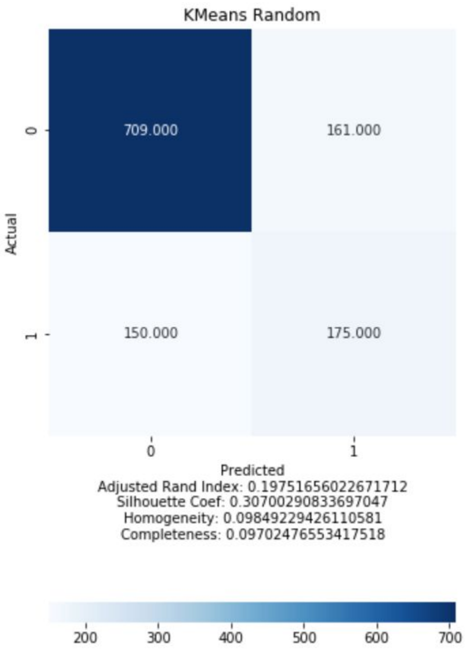
average Minkowski Filtered

```
Adjusted Rand Index: 0.005608925119335567
Silhouette Coef: 0.6831509661079744
Homogeneity: 0.003730258828664535
Completeness: 0.17645250204273305
```

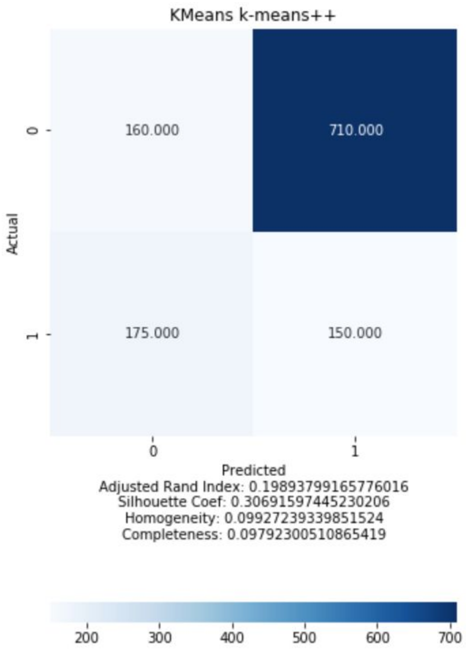
average Cosine Filtered

```
Adjusted Rand Index: 0.12184687489647023
Silhouette Coef: 0.31496044489771025
Homogeneity: 0.07953114989398961
Completeness: 0.06926096482420403
```

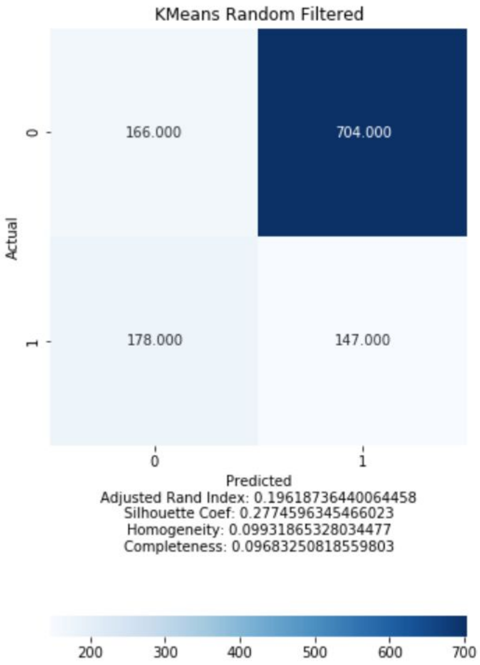
2. KMeans: For this model, Random and k-means++ initialization techniques were used. Using all variables resulted in slightly better performance for the silhouette coefficient. KMeans with all the parameters and random initialization performed the best of the 4.


KMeans Random

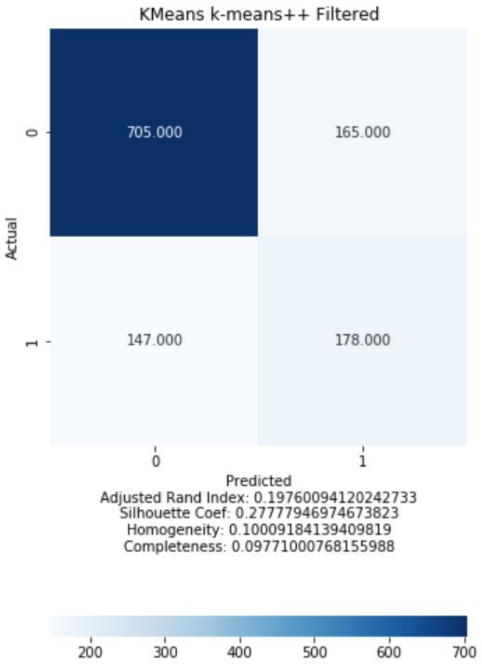
Adjusted Rand Index: 0.19751656022671712
Silhouette Coef: 0.30700290833697047
Homogeneity: 0.09849229426110581
Completeness: 0.09702476553417518


KMeans k-means++

Adjusted Rand Index: 0.19893799165776016
Silhouette Coef: 0.30691597445230206
Homogeneity: 0.09927239339851524
Completeness: 0.09792300510865419


KMeans Random Filtered

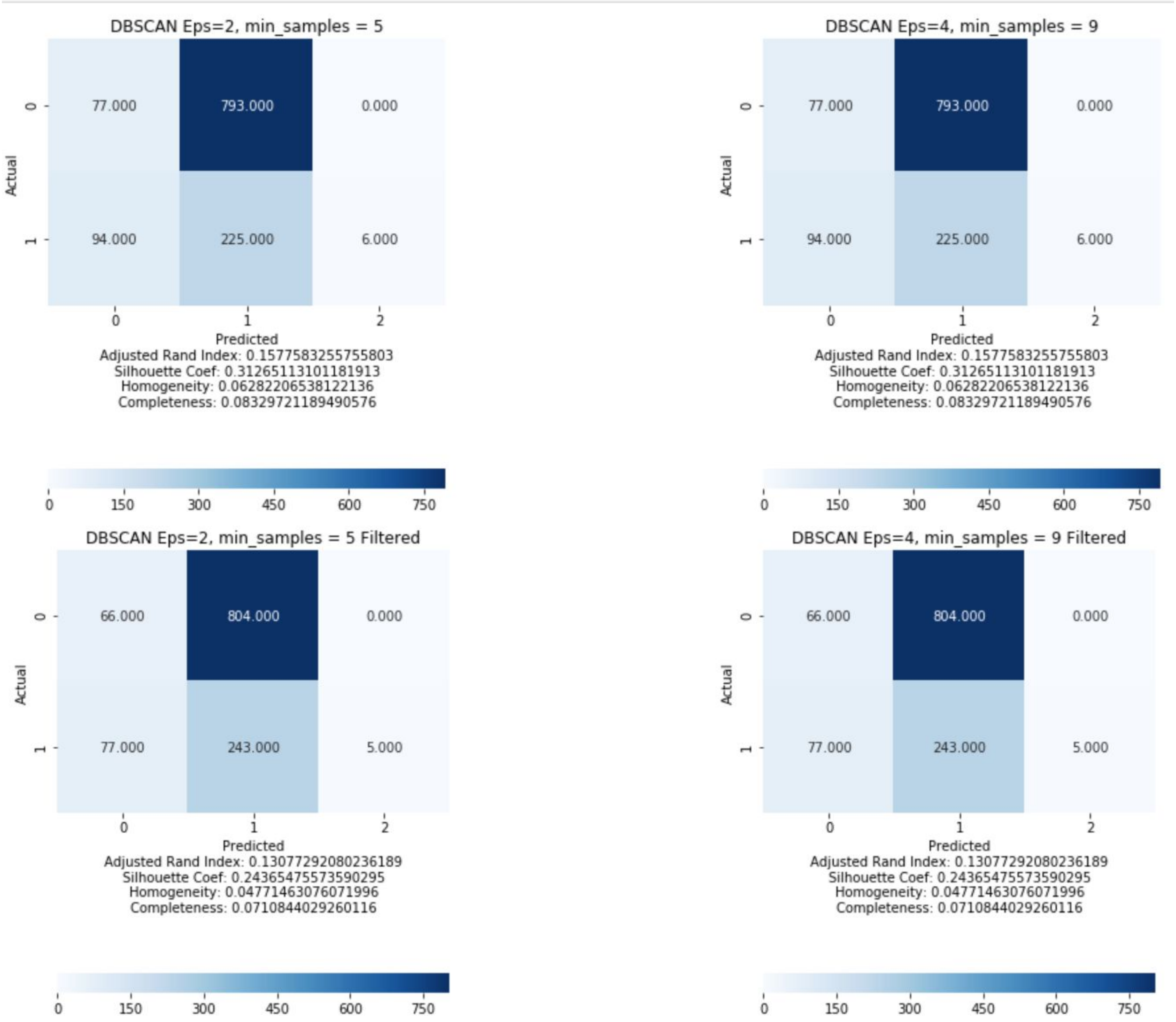
Adjusted Rand Index: 0.19618736440064458
Silhouette Coef: 0.2774596345466023
Homogeneity: 0.09931865328034477
Completeness: 0.09683250818559803


KMeans k-means++ Filtered

Adjusted Rand Index: 0.19760094120242733
Silhouette Coef: 0.27777946974673823
Homogeneity: 0.10009184139409819
Completeness: 0.09771000768155988

3. DBSCAN

   DBSCAN utilized different parameters for the epsilon distance and minimum number of samples. The trend in results are similar to that of KMeans with more variables being better.

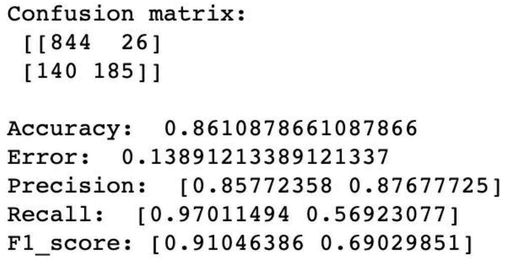


Of all the clustering models, Complete linkage with Manhattan distance had better than average internal and external evaluation metrics and is therefore the best performing model on this dataset.

Task 6: (10 pts.) Create a map of Democratic counties and Republican counties using the counties' FIPS codes and Python's Plotly library (plot.ly/python/county-choropleth/). Compare with the map of Democratic counties and Republican counties created in Project 01. *What conclusions do you make from the plots?*

Answer: From task 4, we got the best score for SVC with kernel = 'rbf'. So, we use the best model to predict the Party of the complete merged set.

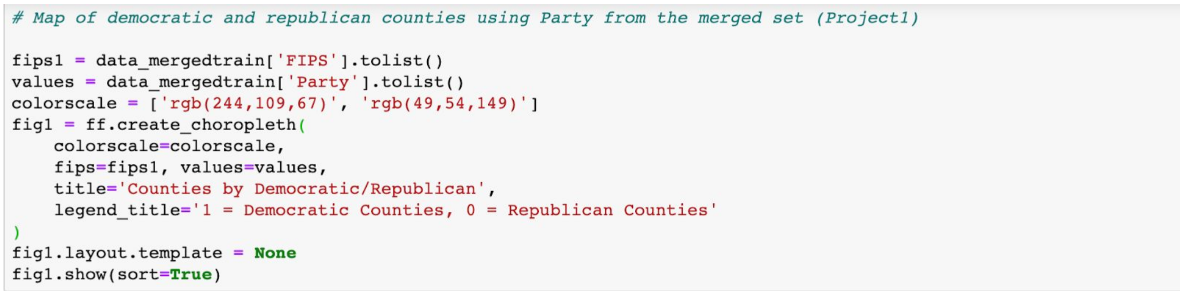
When we evaluate the predicted Party values with the known Party values, we get below metrics:



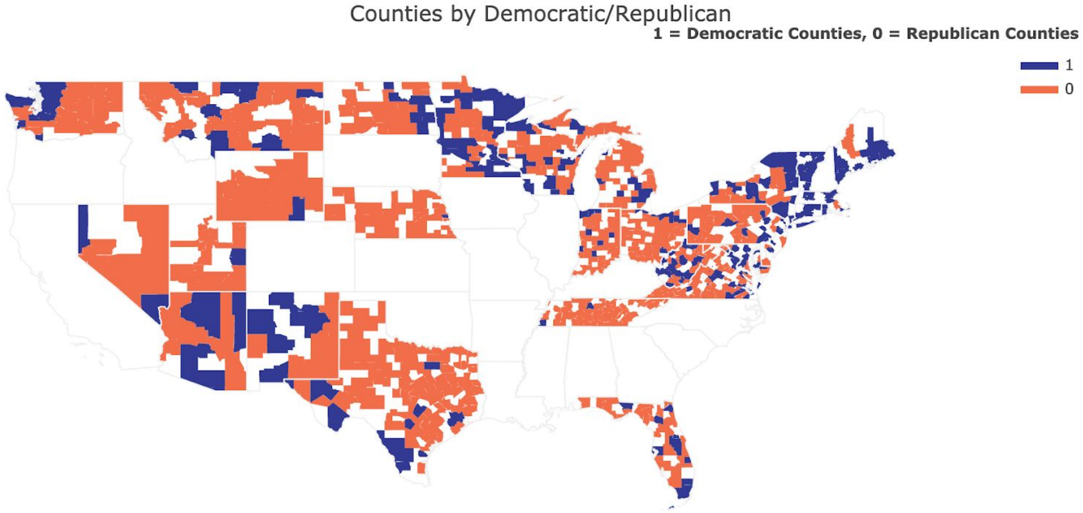
```
Confusion matrix:
 [[844  26]
 [140 185]]

Accuracy:  0.8610878661087866
Error:  0.13891213389121337
Precision:  [0.85772358 0.87677725]
Recall:  [0.97011494 0.56923077]
F1_score: [0.91046386 0.69029851]
```

We create one map based on the FIPS and Party values in given data merged_train.csv and other map using FIPS and Party_pred values which are predicted using the best classifier.

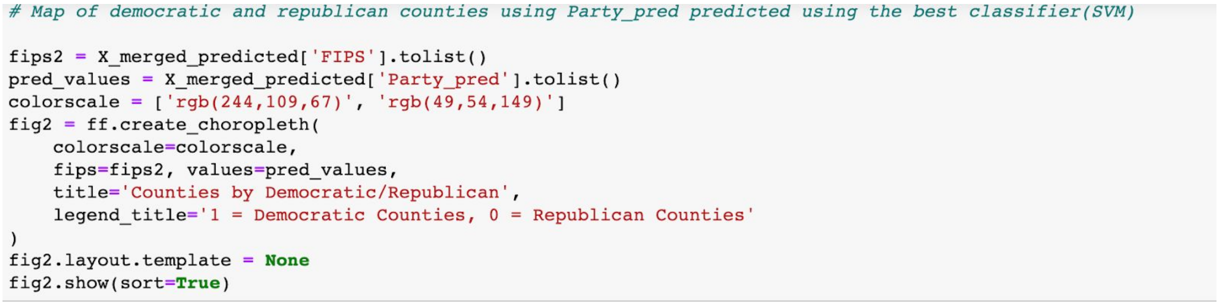
Map with known Party values (merged_train.csv from Project1) :



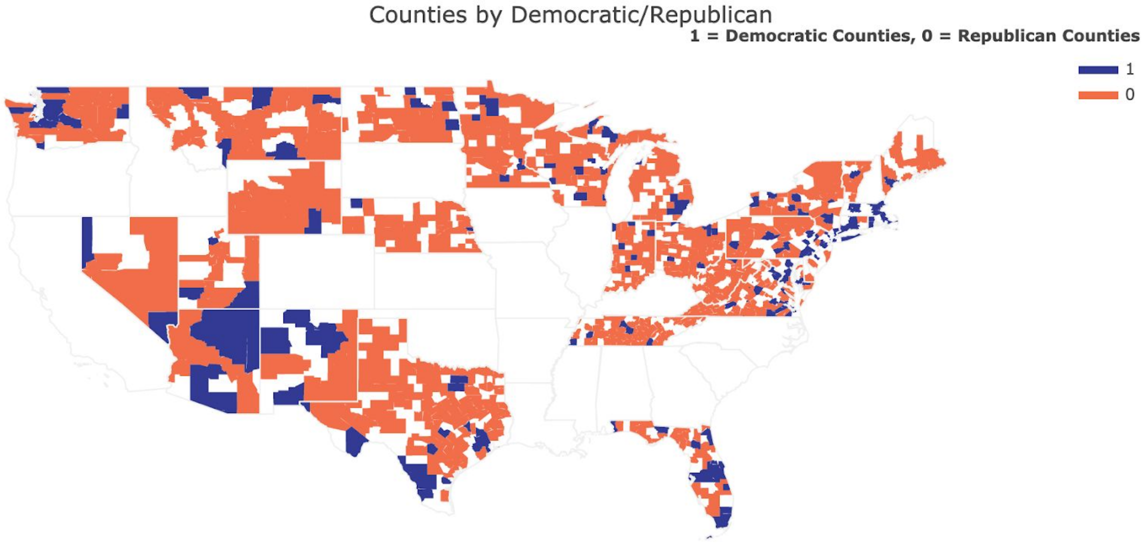
```
# Map of democratic and republican counties using Party from the merged set (Project1)

fips1 = data_mergedtrain['FIPS'].tolist()
values = data_mergedtrain['Party'].tolist()
colorscale = ['rgb(244,109,67)', 'rgb(49,54,149)']
fig1 = ff.create_choropleth(
    colorscale=colorscale,
    fips=fips1, values=values,
    title='Counties by Democratic/Republican',
    legend_title='1 = Democratic Counties, 0 = Republican Counties'
)
fig1.layout.template = None
fig1.show(sort=True)
```


Counties by Democratic/Republican
1 = Democratic Counties, 0 = Republican Counties
1
0

Map with predicted Party values (predicted using best classifier (SVC)) :



```
# Map of democratic and republican counties using Party_pred predicted using the best classifier(SVM)

fips2 = X_merged_predicted['FIPS'].tolist()
pred_values = X_merged_predicted['Party_pred'].tolist()
colorscale = ['rgb(244,109,67)', 'rgb(49,54,149)']
fig2 = ff.create_choropleth(
    colorscale=colorscale,
    fips=fips2, values=pred_values,
    title='Counties by Democratic/Republican',
    legend_title='1 = Democratic Counties, 0 = Republican Counties'
)
fig2.layout.template = None
fig2.show(sort=True)
```


Counties by Democratic/Republican
1 = Democratic Counties, 0 = Republican Counties
1
0

Task 7: (5 pts.) Use your best performing regression and classification models to predict the number of votes cast for the Democratic party in each county, the number of votes cast for the Republican party in each county, and the party (Democratic or Republican) of each county for the test dataset (*demographics_test.csv*). Save the output in a single CSV file. For the expected format of the output, see *sample_output.csv*.

Answer : We settled on LASSO for the best regression model to predict the Democratic and RIDGE for Republican vote tallies respectively. Both used (alpha = 1) as their parameter and trained on the same subset of the features. SVM model was used for classification of county party affiliation. The SVM used parameters {'svc__kernel': 'rbf'}. The results are entered into file ‘classifier_results.csv’.