DS502 Group Project - Can you predict the chance of covid infection by city?

Joshua, Muralidhar,
Kartik, Ryan.

Goal of the Project

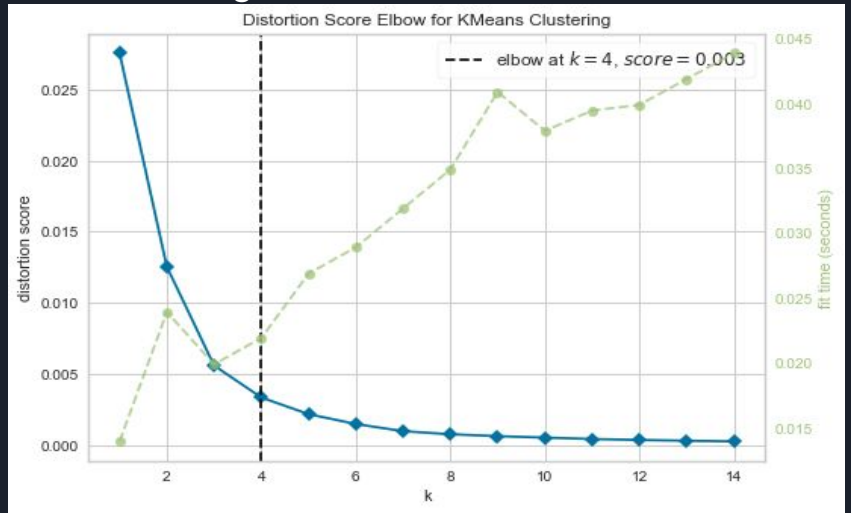
- Predict COVID-19 risk categorically. We derived our insights from these data sources:
  - Use SES Data [1] [2]
  - US Govt Data [2] [3]
  - Massachusetts Municipal Databank
  - Massachusetts Government Weekly Covid Reports
- We define risk categories in the following way:
  - 1) Bucketed infection percentage over the first 6 months of the pandemic (the most important time to respond to pandemics according to epidemiologists). [4]
  - 2) Bucketed infection rate in the last two weeks of the first 6 months of the pandemics, showing the risk of continued spread from that community.
- We value interpretability in our results so we chose models based on their interpretability.
- * [5] [6]

Data Description

- Dataset contains data for 351 cities with 28 columns describing their socio-economic status.
- Broadly the columns fit into 10 classes:
  - Median Household Income
  - Bond Rating - Better Rating means city is able to pay back debts.
  - Population
  - Massachusetts Department of Revenue Income Per Capita
  - Tax Ratings - Better Rating means City is more developed as it has more property taxes.
  - Tax Levy By Class - How much money the city extracts from each of its property asset classes.
  - Employment Statistics - who is employed, unemployed, and the labor force in the cities.
  - Total General Fund Revenue - How much money a city can raise across ALL the ways it can raise money. More affluent cities will have larger fund revenues.
  - Marijuana and Meals Tax - related to the business make-up of the city, the values of voters in the city, type of impact lowest household incomes in city face.

K-Means and Data Processing

- Before running K-Means the NaN values were replaced with zero and the 2019 Bond Rating Column strings were changed to integers on a scale from 1 (best) to 9 (worst) due to the ordinal nature of bond ratings. Data was then normalized.
- Because we have so many features, to plot the data, PCA was also used on the data restricting the number of components to 2.
- K-Means was used to find the labels (dependent variable) for all of our other models.
- Optimal K (4) was found using the elbow method.



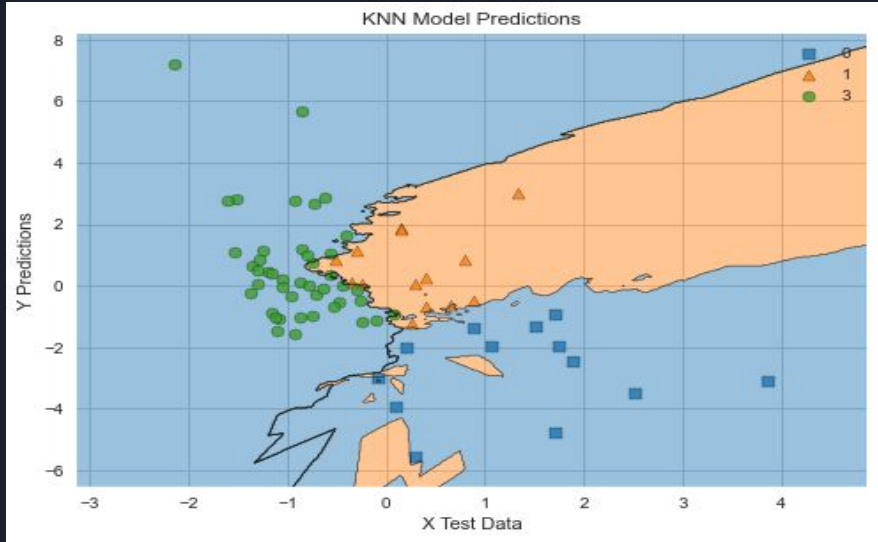
LOO CV vs Traditional CV vs Test set approach

- Our dataset is very small
  - Because of this, traditional CV and Test set both have wildly varying results on the 4 class problem (7 examples in smallest class)
  - We can reduce this sampling problem by using LOO CV (ensuring we always have at least 6/7 represented in our fit dataset)

Model: K-Nearest Neighbor

- Data for KNN was processed in the same manner as before. 80% of the data was used for training. The splits were randomized.
- Grid Search Cross-Validation was used to determine the optimal number of nearest neighbors for KNN. The K values change with each run. For the run displayed it was 28.
- There should be 4 classes predicted but the outliers in our Y-labels caused there to be 3 classes.
    - Weakness of KNN with high K

```
% of population infected (total, in the last 14 days)
[[0.38948114 0.02749798]
 [1.95611493 0.04386887]
 [5.14728078 0.15225721]
 [1.02121059 0.02833615]]
note that there is a doubling in each class, roughly, as we would expect from a pandemic
```



Evaluating The Performance of KNN

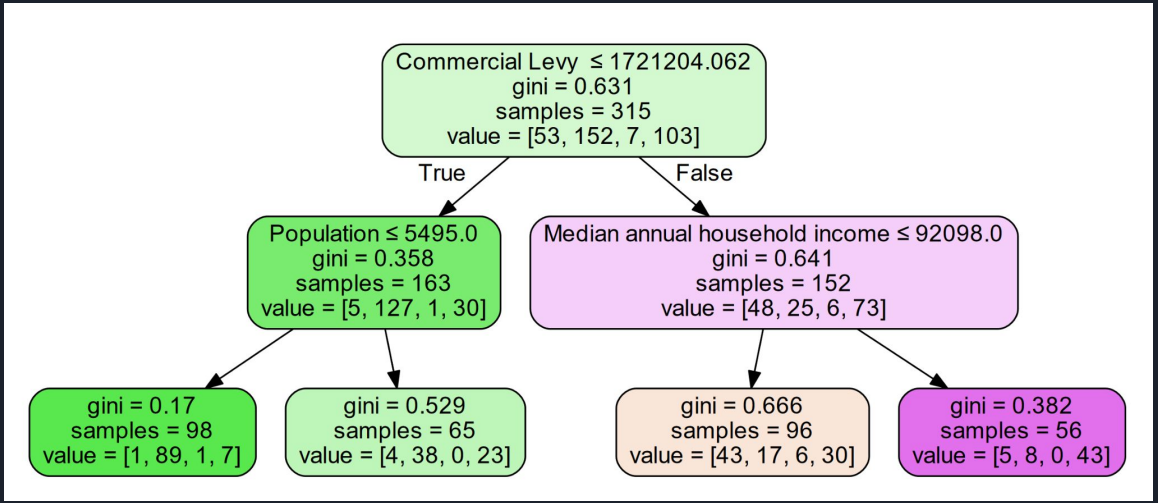
- Unfortunately, KNN was not the most accurate of models. Depending on the run the accuracy ranges from about 60% to 70%. LOOCV accuracy is 62%.
- Bootstrapping was then applied to the data but, unfortunately the accuracy dropped by about 20%.
- Other sampling and cross-validation techniques were also applied such as K-Fold and forward sequential feature selection with no improvement in model accuracy from the baseline.
- Data below from K = 28 run
- We expected KNN to perform the worst of the models because it makes no assumptions. We are using it as the baseline for all other models.

	precision	recall	f1-score	support
0	0.54	0.70	0.61	10
1	0.71	0.38	0.50	26
2	0.00	0.00	0.00	1
3	0.68	0.88	0.77	34
accuracy			0.66	71
macro avg	0.48	0.49	0.47	71
weighted avg	0.66	0.66	0.64	71

```
Classification Accuracy(Percent of Correct Predictions):  66.19718309859155
Misclassification Rate(Percent of Misclassified Predictions):  33.80281690140845
```

Model: Decision Tree

- We fit a decision tree using X-validation to determine the max-depth hyperparameter
- Best performance: depth 5
  - Weakness: uninterpretable
  - We used depth 2 to achieve a “good enough” Decision Tree that can act as a starting point
- Performance: Accuracy on Test Set is 68%, 66% for depth 5 and 2 respectively



Models: Adaboost and Random Forest

- We tried Adaboost using various hyperparameters but it never performed well
  - Max accuracy fluctuates on the test set, 51% on LOOCV
- Random Forest performed well (72% on the Test Set, 66% on LOOCV)
- This makes Random Forest our model of choice for pure accuracy
- Pictured: A classification breakdown on the held out test set (10%) for Random Forest

	precision	recall	f1-score	support
0	0.75	0.43	0.55	7
1	0.75	0.95	0.84	22
3	0.75	0.43	0.55	7
accuracy			0.75	36
macro avg	0.75	0.60	0.64	36
weighted avg	0.75	0.75	0.73	36

Model: Logistic Regression

- Different approaches for features were taken into consideration for this model and the hyperparameters were selected by using gridsearchCV
- Data was split by reserving 10% of samples for the test set to evaluate accuracy but the test accuracy varied significantly as the split changed indicating that the number of samples are less for training.
- Scaling the data improved the accuracy minutely.
- Following are accuracies reported by using Leave one out cross validation.

Data	Test_Accuracy using LooCV
No Preprocessing	62.68
Normalized	**63.24**
PCA (25 components)	61.26

Model Evaluation:

- Running the model again using different hyperparameters produced similar results.
- To regularize and perform feature selection, L1 penalty was used on the model with different ratios of L1. But there was no significant change in the accuracies in the model.
- The model was unfortunately underfitting (train/test accuracy were nearly identical).
- 1 instance of the model is as shown:

	precision	recall	f1-score	support
0	0.60	0.38	0.46	8
1	0.40	0.44	0.42	9
2	0.00	0.00	0.00	1
3	0.81	0.94	0.87	18
accuracy			0.67	36
macro avg	0.45	0.44	0.44	36
weighted avg	0.64	0.67	0.64	36

Correlation/ViF

- To address multicollinearity, we further analysed the ViFs of all the columns
- By excluding the columns which had a very high ViF (inf) in this case, the model accuracy remained unaffected and we were able to achieve a score of 62% using LooCV
- Although there was not much improvement in the accuracy, it was clear that some columns added on to the dimensionality but did not contribute much.

const	7.941352e+01
Population	1.026695e+03
Median annual household income	2.037900e+00
Residential Tax Rating	2.820089e+00
Open Space Tax Rating	1.547467e+00
Commercial Tax Rating	4.503600e+15
Industrial Tax Rating	4.503600e+15
Personal Property Tax Rating	8.241730e+04
Residential APV	9.821888e+01
Open Space APV	2.441207e+01
Commercial APV	1.731525e+03
Industrial APV	7.005092e+01
Personal Property APV	3.368592e+02
Residential Levy	6.548465e+01
Open Space Levy	2.366170e+01
Commercial Levy	1.695012e+03
Industrial Levy	4.513354e+01
Personal Property Levy	3.416939e+02
Labor Force	inf
Employed	inf
Unemployed	inf
EQV Per Capita	1.850559e+00
Total Revenues	1.015937e+00
Meals Tax Rate	1.126668e+00
Marijuana Tax Rate	1.131093e+00
Scaled 2019 Bond Rating	1.446565e+00
dtype: float64	

Evaluation on Binary Classification

- Because the performance degraded heavily due to class imbalance
  - 3 Classes didn't improve things, still had a class with 7 examples
- We can classify "good" vs "bad" much better than our more precise task
- Because this was not the main objective we only show the LOO scores to save time
- KNN: 85%
- Decision Tree: 76%
- RF: 85.7%
- Logistic Regression: 86.8%

Conclusion

- Random guess: 33% Accuracy for 4 classes
- Our best model: 66-75% Accuracy, F1 Score on Test Set: 0.73
- Our best model (2 classes, K-means): 86% Accuracy (RF) (85% other methods)
- Our best model (2 classes, rate of change): 90% Accuracy (Logistic Regression)

- Weaknesses:
  - Could be improved by having much more data
  - Solving the class imbalance problem?
  - That would also solve issues around (stratified) CV not working
  - It would also increase the ability to do automatic feature selection (improve generalization?)

Citations

[1] Municipal Databank (Massachusetts Govt.) https://www.mass.gov/municipal-databank-data-analytics-including-cherry-sheets

[2] Boston Globe Median Household Income
https://www.bostonglobe.com/metro/2018/12/11/full-list-massachusetts-median-household-incomes-town/eZpgJkpB1uF2FVmpM4O8XO/story.html

[3] Weekly Covid Health Reports (ours taken w.o. August 5th) (Mass. Govt.)
https://www.mass.gov/info-details/archive-of-covid-19-weekly-public-health-reports

[4] Pandemic Preparedness Findings from the Council on Foreign Relations
https://www.cfr.org/report/pandemic-preparedness-lessons-COVID-19/findings/

[5] Hawkins, R. B., Charles, E. J., & Mehaffey, J. H. (2020). Socio-economic status and COVID-19-related cases and fatalities. Public health, 189, 129–134. https://doi.org/10.1016/j.puhe.2020.09.016

[6] Khanijahani, A., Iezadi, S., Gholipour, K., Azami-Aghdash, S., & Naghibi, D. (2021). A systematic review of racial/ethnic and socioeconomic disparities in COVID-19. International journal for equity in health, 20(1), 248. https://doi.org/10.1186/s12939-021-01582-4