CS 496: Data Science Seminar

Checkpoint 4

The Wise Kingfishers: Roman Svintitskyy and Shalini Roy

In our project, we are trying to analyze the North vs. South sides of Chicago police complaints data to investigate the correlation between class + racial differences and police violence. Having performed a quantitative analysis with SQL for Checkpoint 1 and a qualitative visual analysis with D3.js, we understood the importance of performing a more solid, substantive analysis of the questions that arose during previous checkpoints. Therefore, for this checkpoint, we will be developing:

- **Q1**: Two time-series predictive models that predict the number of police officers for the North and South side of Chicago, respectively, while accounting for the number of allegations per 10,000 people. To further investigate the effect of various features on the number of police officers per police district, we built another linear regression model and performed Recursive Feature Elimination.
- **Q2:** A machine learning model predicting the number of TRRs per year in the north and south sides of Chicago, accounting for such features as demographics of the officers and the communities we serve. We will also attempt to analyze how the effect of hyperparameter tuning.

Both these themes are important to research to better understand potential police misconduct anomalies.

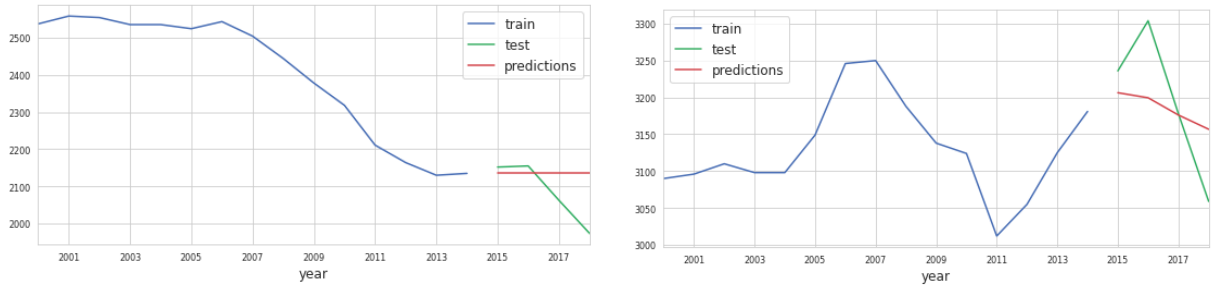
Question 1

Link to code:

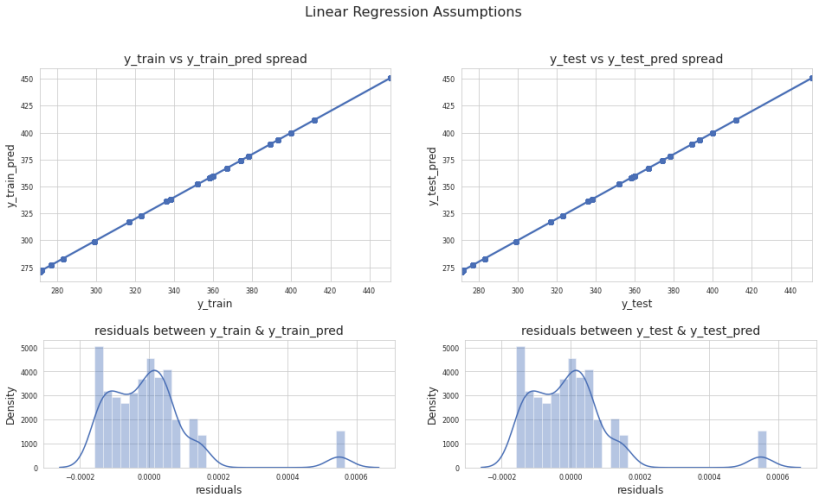
https://github.com/romansvintitskyy/The-Wise-Kingfishers/blob/main/checkpoint_4.ipynb

To build the datasets for both parts of Question 1, we used SQL; the resulting datasets will be in the src folder. Our RandomForestRegressor's performance has delivered promising results but could definitely be improved. In the figures below, we can see that the model for the South (right) predicts better than on the left. However, even after hyperparameter tuning, both had a rather high RMSE. **In short, the model for the North (left) predicts that the number of police officers will remain relatively stable with little fluctuation. On the other hand, the model for the South predicts that there will be a gradual decline in the number of police officers assigned to the**

area. It would be interesting to further investigate whether this, in fact, happens, and how it will affect the per capita number of police misconduct cases.



For the second part of the question, we trained a Ridge regression model that predicts the number of police officers for a given set of features. For this part, we constructed a different dataset which consisted of several numerical features (officer count, racial distribution of officers, civilian count racial distribution of the civilian community, and the per capita number of allegations) and categorical variables ('years', 'unit_name', 'race'). Surprisingly, this model scored exceptionally well, the R^2 was close to 1 on both train and test datasets. Before training the model, we normalized the distribution of variables with skewness > 0.5 with coxblox transformation, transformed categorical variables with dummy variables, scaled the values, and finally performed Recursive Feature Elimination to select the most impactful predictors out of the resulting 48 and avoid the curse of dimensionality. However, unfortunately, despite all the pre-processing, the plot for the distribution of residuals between actual and predicted values shows that said distribution is slightly right-skewed. Therefore, further parameter tuning and pre-processing are required. **With regard to our research question**, RFE has determined that per capita allegation numbers are less impactful predictors than other features.

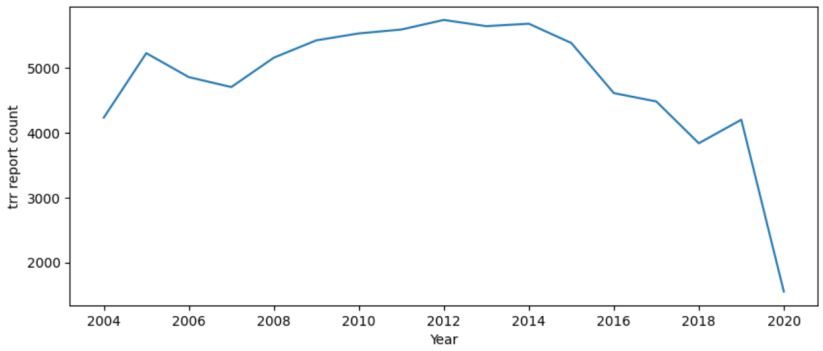


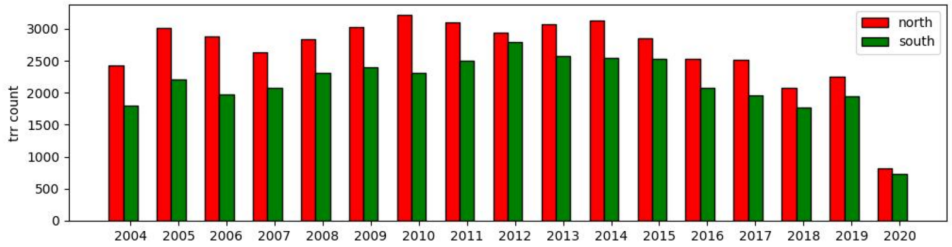
Question 2

Link to code:

https://github.com/ShaliniR8/WiseKingfishers/blob/master/cpdbML.ipynb

What is TRR? TRR is the abbreviation for *Tactical Response Reports*. Usually, an event has one complaint with a unique id (CRID) linked to it. But in cases when an event has one or more use-of-force reports linked to it, it comes under TRR data. Let's check out the trr counts from each year. First, we'll look at the overall trr count yearly, and then we will visualize the trr count by year distinguished by north and south sides.





Data from df0 -

	count	side	years
0	2399	south	2009
1	2570	south	2013
2	1951	south	2019
3	1981	south	2006
4	2858	north	2015
5	2880	north	2006
6	2430	north	2004

Observation: It is interesting to note that in all cases, multiple cases of complaint reports related to a single event usually comes from the north side. This contradicts our initial data exploration in SQL, where we deduced that there are higher cases of violations. This will require more exploration later. Overall, 2020 has far fewer trr reports. This could be because the data for 2020 is more recent and possibly lacking.

Question: *Can we use this information to create a reliable model that can predict the trr count per year?*

We are going to use df2 for this part, which contains all the beat names and is categorized according to trr count per year per beat. We are going to define our goal first.

	beat	count	years
0	1923	7	2004
1	1421	7	2015
2	2525	9	2013
3	2524	14	2013
4	1524	25	2019
5	1021	36	2013
6	633	14	2016
7	1114	21	2016
8	912	15	2009

This is a 3540 x 885 dimensions table, larger than df0 because we added beat number.

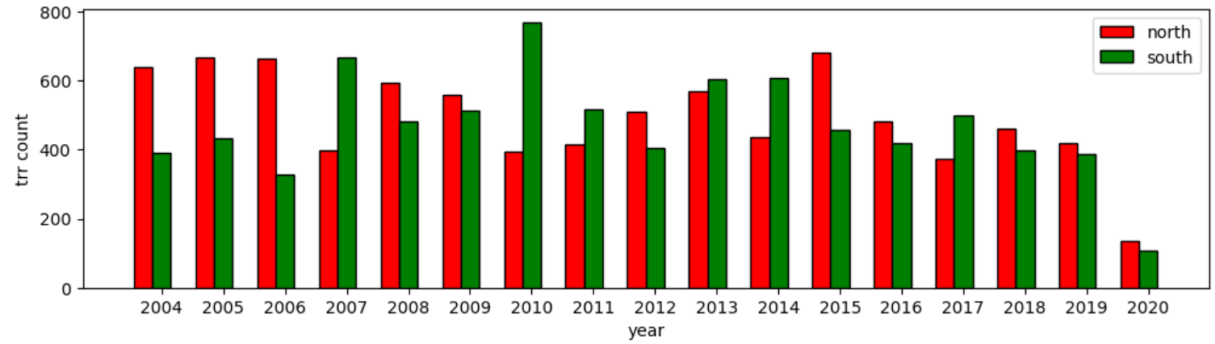
GOAL : Predict the yearly trr count by city side.

Question: Why are we using df2?

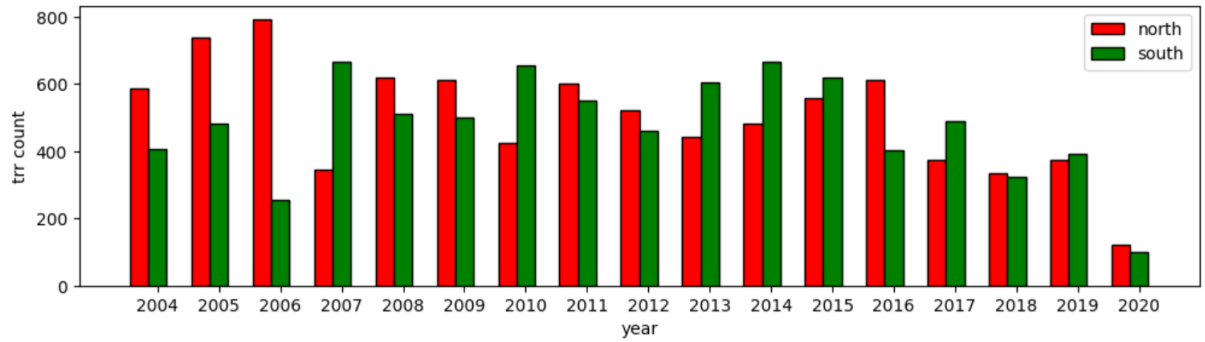
df0 only concerns two columns: city side and trr count. This is a very small dataset and, therefore, not reliable enough to train our model. df2, however, is concerned with more detail- the beats. Therefore, this is more reliable for training our model. These are the results.

	model	best hparams	best score
0	LinearRegression()	{'lreg__positive': False}	0.28358040933802797
1	Ridge()	{'rreg__alpha': 1, 'rreg__positive': False}	0.2395358245453677
2	GammaRegressor()	{'greg__alpha': 0.01, 'greg__warm_start': True}	0.27506295104893674
3	SGDRegressor()	{'sgdreg__learning_rate': 'adaptive', 'sgdreg__max_iter': 10000, 'sgdreg__penalty': 'l2'}	0.24076661520635434
4	KernelRidge()	{'kreg__alpha': 5e-05, 'kreg__kernel': 'rbf'}	0.7032042975630555

Predicted trr report counts per year for beats included in test data using Kernal Ridge



Actual trr counts per year for beats included in test data



Although a 70% score is poor, Kernel ridge shows a massive improvement over other models while drawing information from only year and beat (which indirectly implies the side of the city).