Final Report

Conor Dalton

December 2021

1 Predicting Bike Station Occupancy

1.1 Feature Engineering

1.1.1 Selection of Station

In choosing the stations I was going to use I decided to display them on a map. This would make it easier to choose stations which would have different behaviour. I decided to choose station 97 (Kilmainham Gaol) as it is the furthest from the city center. I also chose station 109 (Buckingham Street) as it is beside Connolly and I assumed it would have drastically different behaviour than my other station.

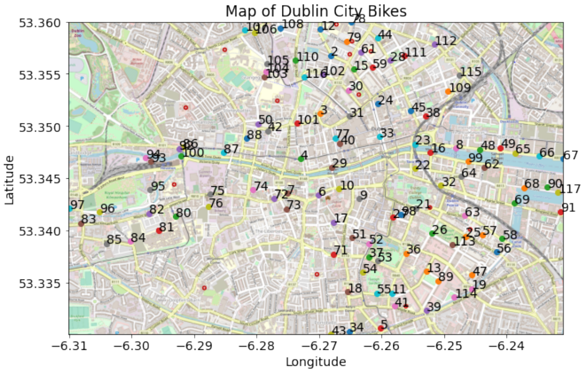


Figure 1: Map of Dublin Bike stations.

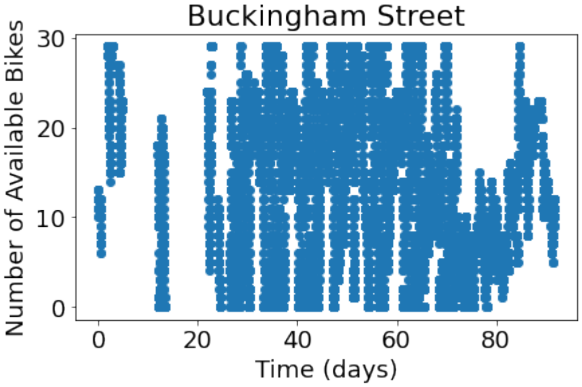


Figure 2: Availability of bikes over time at Buckingham Street

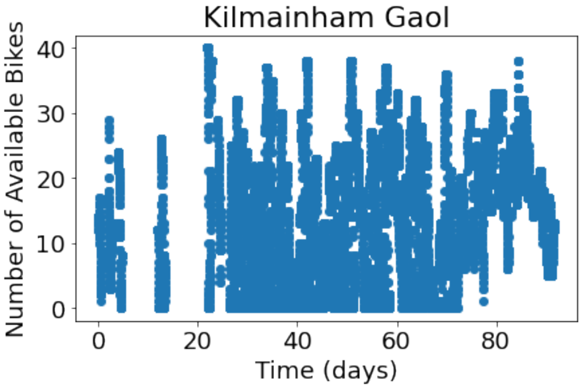


Figure 3: Availability of bikes over time at Kilmainham Gaol

1.1.2 Data Cleaning

As can be seen in figures 2 and 3, much of the data in the first few weeks is missing. I decided to simply remove the first 27 days in both datasets as there was so much missing that it would likely make it more difficult to work with the data in a way that didn't negatively affect the performance of the models. It is likely that there were other missing data points, this is evident in that not every day has the same number of entries (between 287 and 289). This is unlikey to have any major impact on my predictions as it is very close to 288 (the number of 5 minute time blocks in a day).

1.1.3 Feature Selection

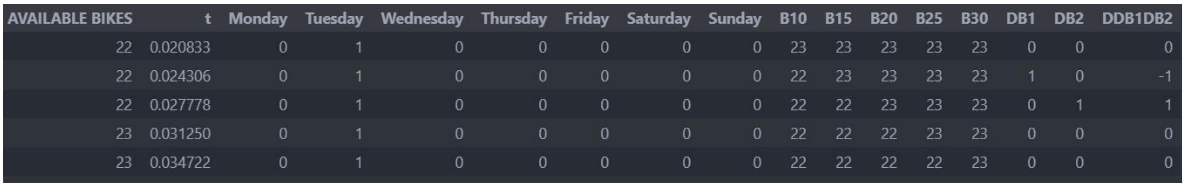
The features I took from the datasets are the time and the number of bikes at that time. This allows other features to be derived from this.

I turned the time of day into a float (it was slighlty skewed because the first reading of the year was around 6am.) This was obviously an important feature to take into account as different times of day obviously have different behaviour in terms of bike usage.

The day of week undoubtedly has an effect on how dublin bikes are used. This means that it is important to take the day of week into account when training a predictor on this data. Weekdays can be represented with numbers (e.g. from 1-7) but this could cause problems in linear models. To counteract this I used one-hot encoding, this allows categorical data such as weekday to be presented in such a way that is less likely to cause unnecessary bias. The downside of this is that it adds many more dimensions to the feature vectors which can make models unnecessarily complex.

As I was attempting to predict values for the number of bikes available, I used past values for available bikes. For each type of prediction I used the 5 previous values for available bikes (for 10 minute ahead prediction I used the values at 10, 15, 20, 25 and 30 minutes earlier. This is the same as what I did for 30 and 60 minute predictions.)

I also made some new features DB1, DB2, and DDB1DB2 as can be seen in figure 4. DB1 and DB2 are the differences between number of bikes in the most recent datapoints. DDB1DB2 is kind of like the 2nd derivative as it is the difference between DB1 and DB2. I thought that these features would be useful in prediction.



AVAILABLE BIKES	t	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday	B10	B15	B20	B25	B30	DB1	DB2	DDB1DB2
22	0.020833	0	1	0	0	0	0	0	23	23	23	23	23	0	0	0
22	0.024306	0	1	0	0	0	0	0	22	23	23	23	23	1	0	-1
22	0.027778	0	1	0	0	0	0	0	22	22	23	23	23	0	1	1
23	0.031250	0	1	0	0	0	0	0	22	22	22	23	23	0	0	0
23	0.034722	0	1	0	0	0	0	0	22	22	22	22	23	0	0	0

Figure 4: Example of feature dataframe for 10-minute-ahead prediction

For the ridge regression prediction I also trained similar models with polynomial combinations of the data to see if this better captured the relationships between the different features. I used every ccombination of features up to a power of 2 which gives 153 features. This obviously makes a far more complex model and it is far more confusing to figure out what models are doing when there are so many different features.

1.2 Ridge Regression

I initially decided to use ridge regression to predict future values. I wanted to use some form of linear regression because I used them for a similar problem in my group project. Lasso regression is more likely than Ridge to reduce coefficients to zero but I was curious to see how even the lightly weighted parameters were affected so I decided to use ridge regression instead.

I used 5 fold cross validation to find the ideal value for alpha to use and I found that varying alpha had very little effect until alpha became quite large as can be seen in figures.... I decided to use a value of 1 for every model as this value gave a near minimum mean-squared error for every prediction model.

The metric I used to evaluate performance is mean squared error. I feel that this is a suitable metric as it tells us how far off predictions from a model are.

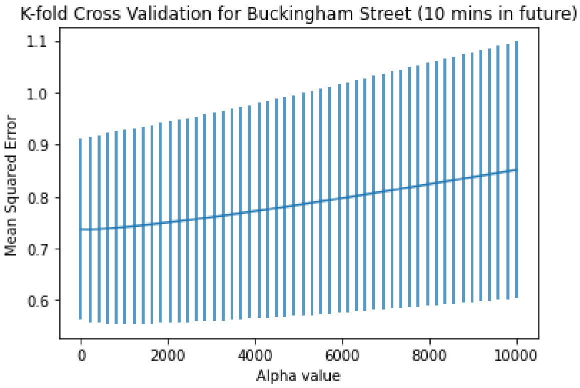


Figure 5: K-fold cross validation for ridge regression.

Training ridge regression on all of the datasets produced very similar graphs (see appendix) so an alpha value of 1 was used for all.

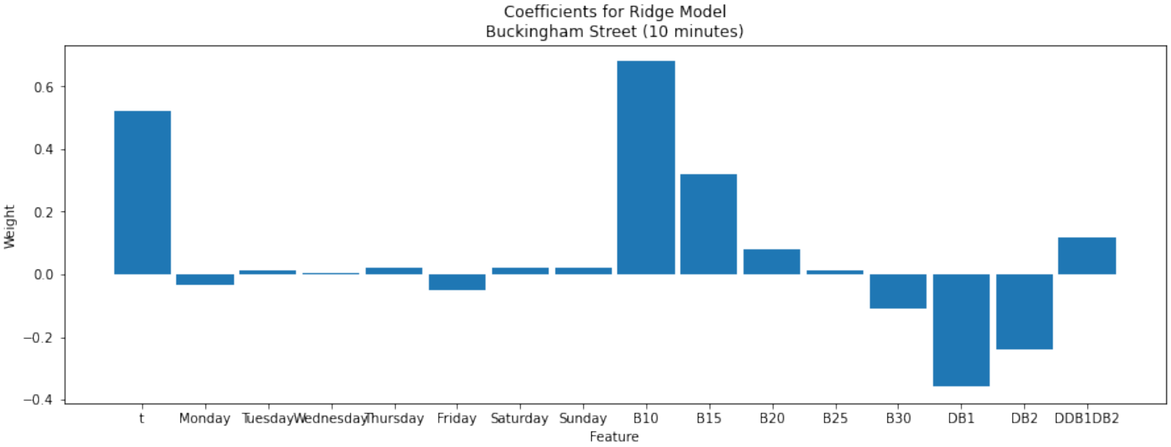


Figure 6: Weighting of coefficients for ridge regressor predicting 10 minutes ahead.

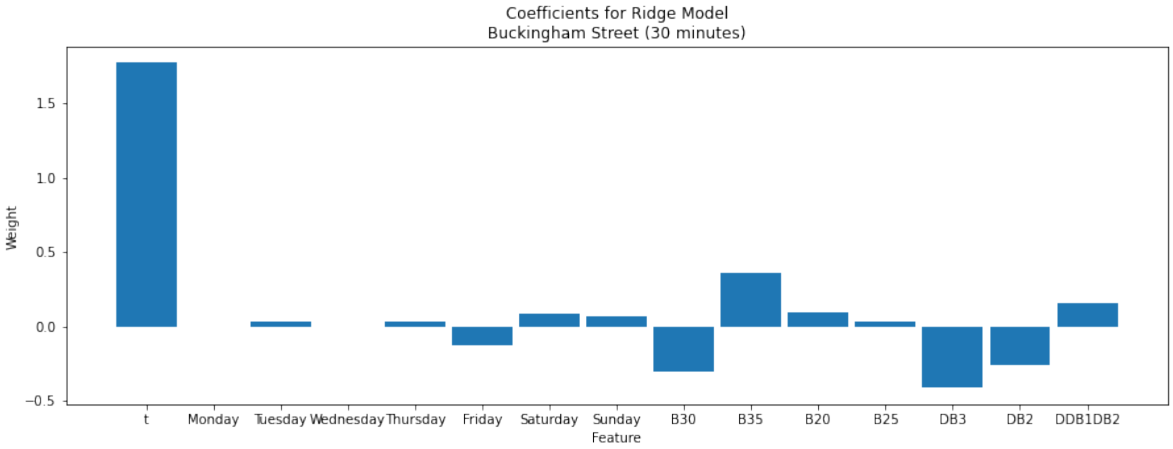


Figure 7: Weighting of coefficients for ridge regressor predicting 30 minutes ahead.

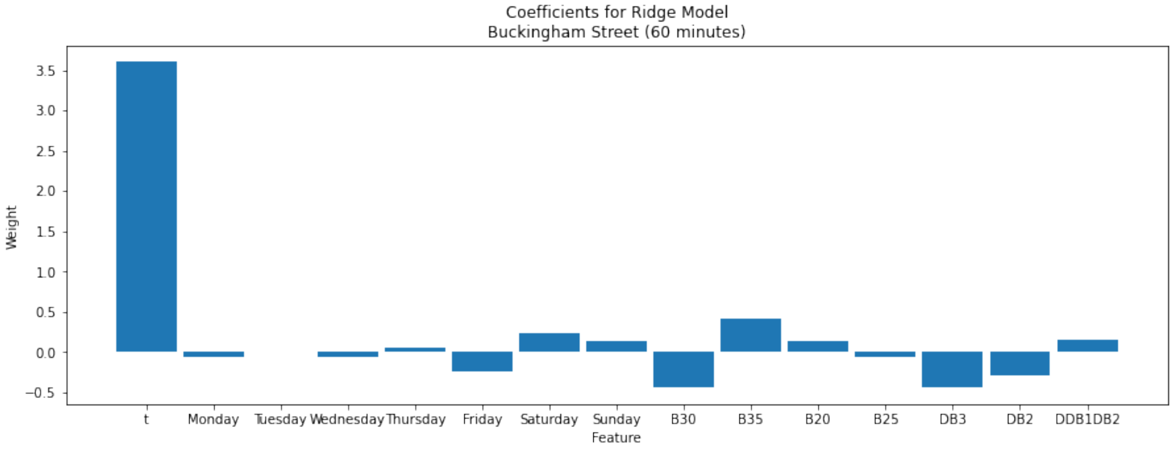


Figure 8: Weighting of coefficients for ridge regressor predicting 50 minutes ahead.

1.3 K Neighbours Regression

I decided to use K-neighbours regression as it works quite differently from ridge regression and is supposedly suitable for predictions in time series data. I used K-fold cross-validation on a number of parameters used in training. Changing the value of p had almost no effect ao I saw no reason to change it from the default. Changing the number of leaves had almost no effect so I saw no reason to change it from the default. Changing the number of neighbours had a massive effect as can be seen in figure 9.

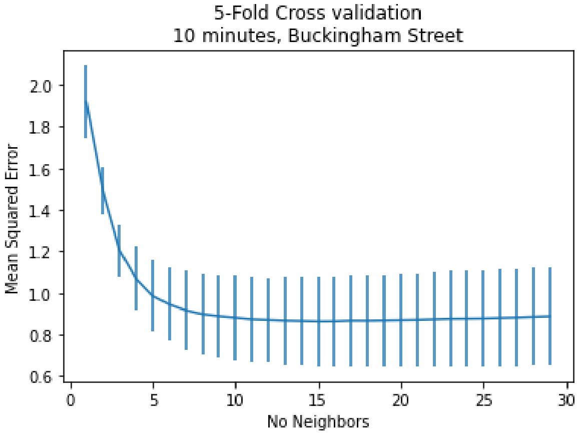


Figure 9: K-fold cross validation for k-neighbors regression.

All of the k-fold cross validation graphs produced similar shapes with the minimum mse being around 10 neighbours and getting diminishing returns for anything higher.

1.4 Evaluation

I decided to use what I felt was an appropriate dummy predictor. The dummy predictor simply predicts that the value will be the same as the previous value. for example, when predicting 10 minutes ahead the dummy will use the value 10-minutes before. This makes the dummy predictor quite accurrate and explains why the predictor models did not always outperform it.

1.4.1 Ridge Regression

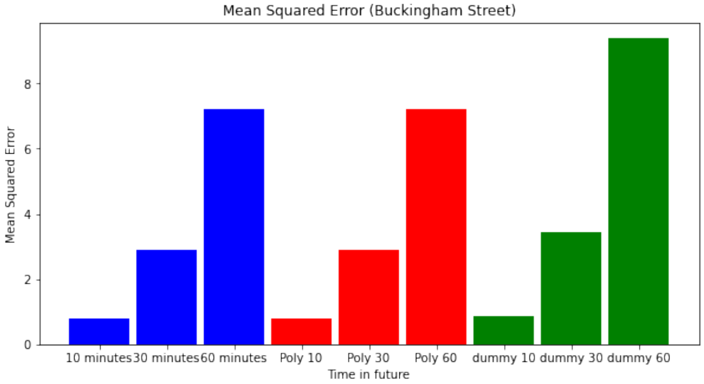


Figure 10: Mean squared error with ridge regression on Buckingham Street.

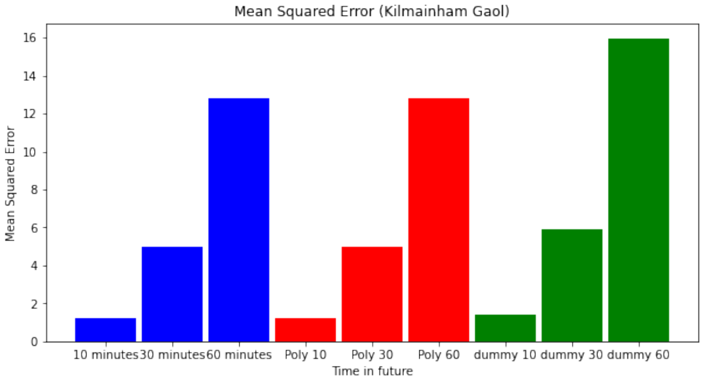


Figure 11: Mean squared error with ridge regression at Kilmainham Gaol.

As can be seen in figures 10 and 11, the ridge predictors performed slightly better than the dummy predictors. It is also clear that using the polynomial features had vey little effect on the performance.

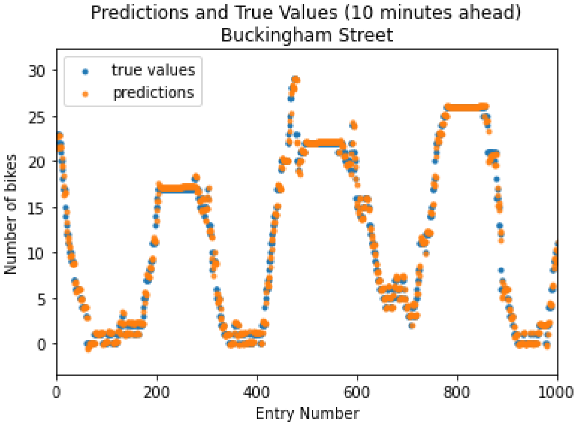


Figure 12: Predictions using ridge regression

1.4.2 K-neighbors Regression

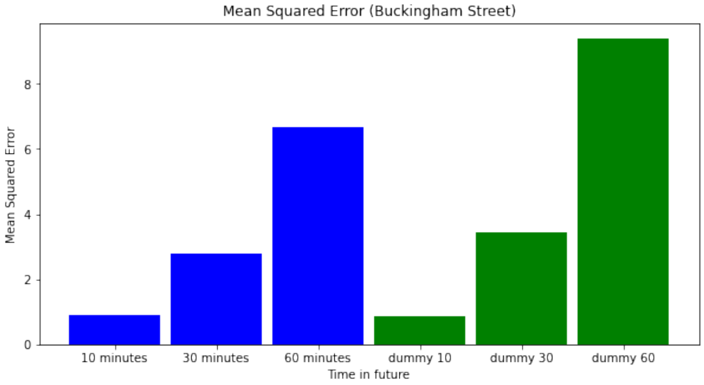


Figure 13: Mean squared error with K-neighbours on Buckingham Street.

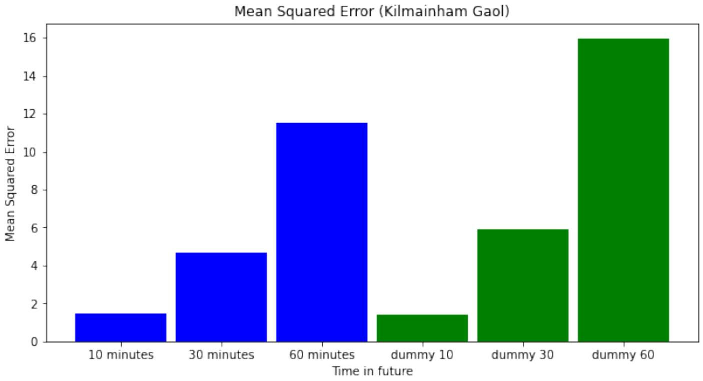


Figure 14: Mean squared error with K-neighbours at Kilmainham Gaol.

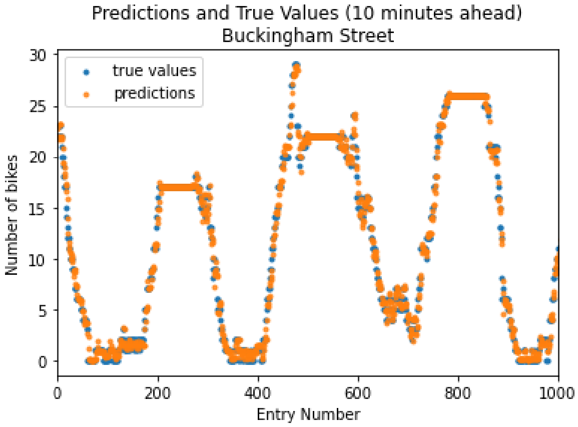


Figure 15: Predictions using K-neighbours regression

1.4.3 Comparison

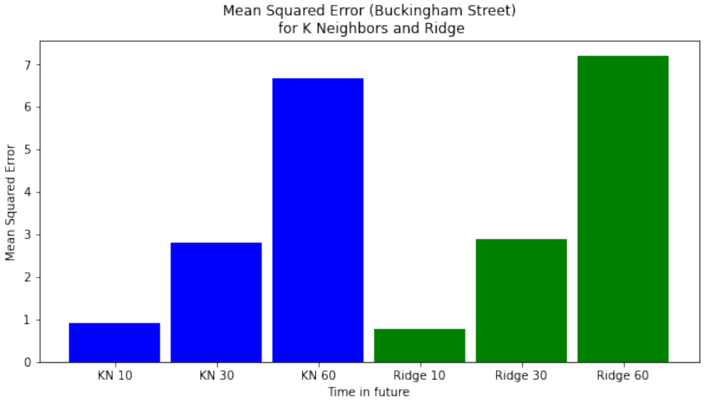


Figure 16: Mean squared error with ridge regression and K-neighbours on Buckingham Street.

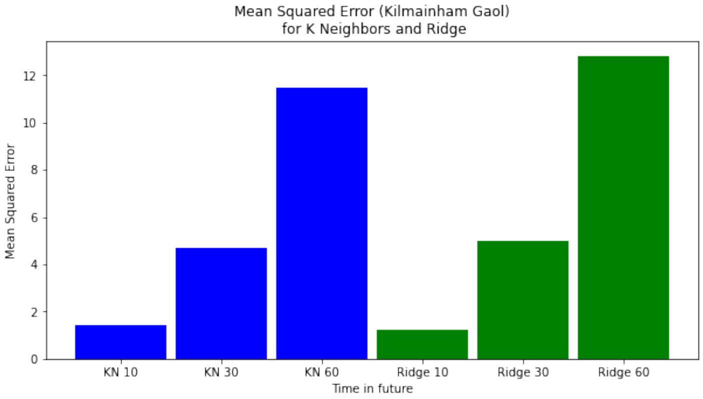


Figure 17: Mean squared error with ridge regression and K-neighbours at Kilmainham Gaol.

As can be seen in figures . . .

2 Shorter Questions

2.1 ROC Curve

An ROC curve is...

2.2 Situations Where Linear Regression will be Inaccurate

Linear regression is ...

2.3 SVM Classifier vs. Neural Net Classifier

An SVM Classifier... an Neural Net Classifier

2.4 Operation of a Convolutional Layer in a ConvNet

ConvNets are... Convolutional layers are. . .

2.5 K-fold Cross-Validation

Resampling is done to change amount of data.

A Appendix

asdasd