Artificial Intelligence Train Delay

Towner Hale

February 14, 2019

1 Introduction

This report will be discussing the effect of a multi-layer perceptron (MLP) on predicting train delay for four stations in the Greater Anglia railway: Liverpool Street (LIVST), Chlemsfield (CHLMSFLD) Colchester (CLHST), and Norwich. There were over 30 features used in predicting each station, and as the train model progressed from LIVST to Norwich, more features with input from the previous stations were added to the next station to determine if they could improve the accuracy in predicting train delay. The performance of the seven days of the week, hourly time, peak or not peak, and delay or not delay features will be used as baseline for each of the four station models. Additionally, four station models were used in predicting departing delay, and another four station models were used in predicting the arrival delay to analyze the change in performance of the neural network. 6 months of the 2018 data (January through June) were used for the training and test data. The multi-layer perceptron, a supervised learning algorithm that was imported from scikit, was used as a neural network and had multiple hidden layers (at least 30) between the input and output layer. Section 2 gives a brief description of the features used as inputs for the MLP, section 3 gives a brief description of the methology for training the MLP and how the MLP works, section 4 gives the results and evaluation of the performance MLP on the given models, and section 5 is the conclusion.

2 Features

2.1 Days of the week

Each day of the week was used as a feature in the multi-perceptron layer due to varying differences in train delay between stations depending on the day. For all of the stations, the average delay on the weekend was lower than the delay during the week, which is shown in Figure 1. The average departing delay on the weekend is 41.9 percent lower than the week delay for LIVST, 34.6 percent lower for CHLMSFLD, and 26.6 percent lower for CLHST; this discrepancy demonstrates that there is a significant difference in delay times between the weekend and weekday, therefore an additional feature value was added that contained a 1 for the weekend, 0 for the weekday. Additionally, seven features, one for each day's average, were also included in order to help accurately predict the specific delay for the day. At the Liverpool and Chlemsfield stations, the weekend delay times were closer to the 60 seconds delay that is often seen at the stations; with this thinking, a feature was added and named morelikely60, in which it is believed that there is a higher chance of a minute delay if it falls on the weekend.

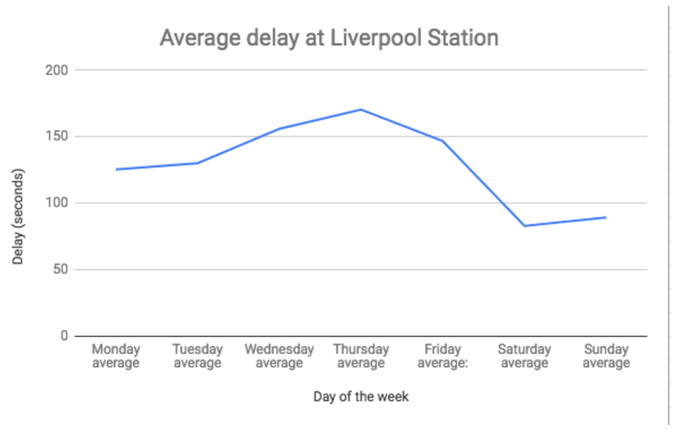


Figure 1: Average delay for the day of the week at Liverpool station

2.2 Time of the day

It is predicted that the delay varies depending on the time of the day as well as peak and offpeak times; features were added for each hour that the station is open (from 6 am to 2 am). The time was categorized by peak/offpeak as defined on the Greater Anglia website: peak departing time is any time before 09:30 am and between 16:29-18:34. Peak time does not apply towards the weekend, so it is always offpeak on Saturday and Sunday. It was predicted that peak times would have higher delays because of the amount of traffic going through the station would be greater, which can lead to potential delays and clog ups. Therefore, it is believed that by adding two features determining peak and offpeak time, the accuracy of the model would improve.

2.3 Arrival times, delays, and additional features

Arrival time delays were added as a feature to determine if it indicated a certain amount of delay that would be relative for the departing delay. It is predicted that if there is an arrival time delay of a minute for a specific station, then there would be a departing delay around the same time. Since the MLP was concerned with predicting the amount of the delay and not if there is a delay, features delay and extraweightnodelay were added to predict a 0 second delay if there was no delay, and guess the amount of delay if there was one. Lastly, four models were created in order to determine if a delay time at the previous station would have a correlation with the delay time of the next station (so if there was a departing delay of 120 seconds at Norwich, there would be an arrival delay around that time at Chlemsford).

3 Methodology

3.1 Multi-layer Perceptron

The multi-layer perceptron network consist of an input, and output, and hidden layers that consists of hidden units that are perceptrons. In an ordinary perceptron, there are inputs that contain a weight for each input, as well as one overall bias, that is sent to the activation function to produce an output. The multi-layer perceptron simply adds layers, where each hidden layer (which don't directly see the input of features) contain an array of perceptrons and has an output that is expressed as a function, seen in Figure 2.

$$f(x = \mathrm{G}W^{T}x + b)$$

$$f : R^{D} \rightarrow R^{L},$$

Figure 2: Hidden layer perceptron function

D is the size of the input vector x (which contains each feature), while L is the size of the output vector. G represents the activation function, which play a key part in the neural network: weighted sum of the inputs as well as the bias for each input is calculated to determine which node to activate in the particular layer. Each hidden layer maps the vector of dimensionality (which corresponds to the number of inputs) to the vector containing the number of outputs. The activation function is applied to a weight matrix and bias vectors contained in the hidden layer to produce the output. A visualization of the MLP and its layers is seen in Figure 7.

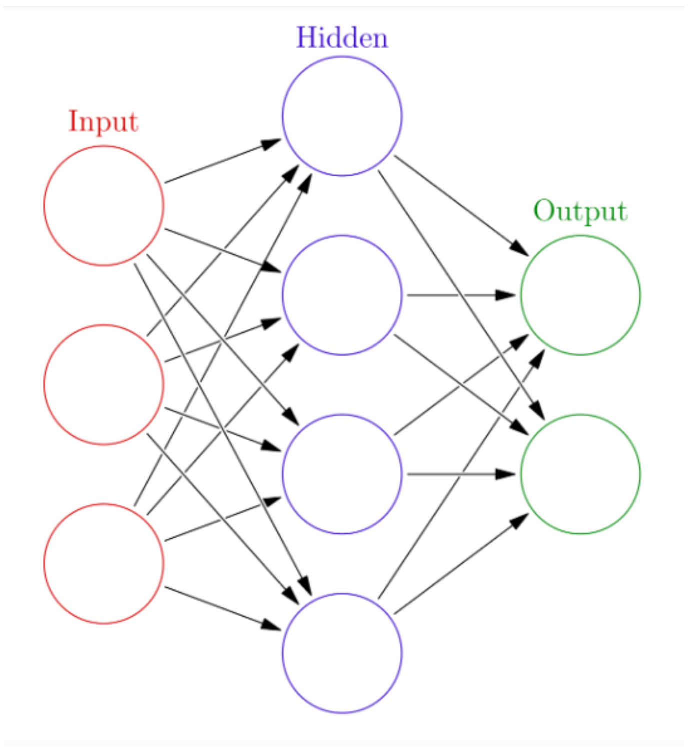


Figure 3: Visualization of the multi-layer perceptron

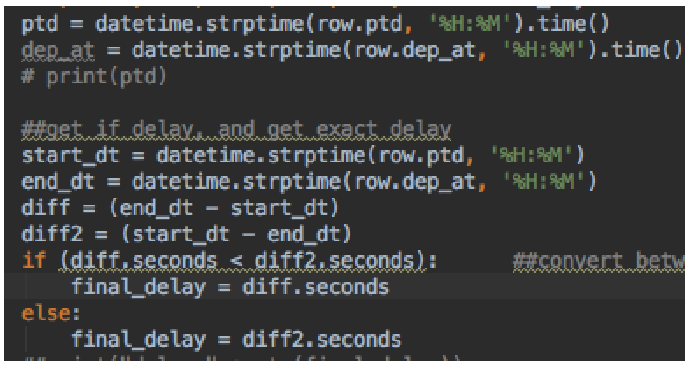
For the implementation of the MLP, SciKit-Learn was downloaded in Python and the traintestsplit function from sklearn.modelselection was initially applied to the data, in which the x matrix was split into two variables, xtrain and xtest, and the y matrix was split into ytrain and ytest. One issue with the MLP is that it is sensitive to non normalised data, where the range of values within a data set vary and can skew the weight of each feature. Feature scaling resolves this issue and is used via the built-in StandardScaler function; the Standardscaler standardizes features by removing the mean and scaling to unit variance. It does so by subtracting the training sample's mean from the score of the sample and by dividing by the standard deviation of the training samples. The fit function performs the above function, and then the transform function is called so that the new, standardised data is created into an array. The training and test data for the x vector are both transformed, and then the MLP classifier fits both x and y training data. The MLP classifier's default values were first used with hidden layer n equal to the size of the features m (so one hidden layer for each feature). This methology for establishing a deep neural network is found in the book Deep Learning:

> Empirically, greater depth does seem to result in better generalization for a wide variety of tasks. This suggests that using deep architectures does indeed express a useful prior oer the space of functions the model learns.

(Brownlee, 2018). The parameters were then changed to see their effect upon predicting the train delay. The predict method from the classifier was then used on the test data of the X vector to create a confusion matrix from the X and Y test data (Portilla, 2018).

3.2 Initial models

The models used the dataset for the first 6 months of 2018, which were all combined in the file location-2018-2 csv file. The first four initial model names all began with sort-times then the station name (so sort-times-Liverpool, sort-times-Chlemsfield...), and were created with different intentions than the last four: they were concerned with testing the MLP's accuracy in predicting the departing delay at each specific station. This was done by checking each row to determine if there was a discrepancy between the ptd (planned departure time) and dep-at (actual departure time). If there was a backslash N in one or both of the values, then the row was skipped; this would serve as a problem later in reduced data set size since there were a lot of N values in the data. Seen below in Figure 4, the datetime function was in charge of converting the time into minutes and seconds, which would then be sent to the determinePeak and determineTime methods for the features.



```
ptd = datetime.strptime(row.ptd, '%H:%M').time()
dep_at = datetime.strptime(row.dep_at, '%H:%M').time()
# print(ptd)

##get if delay, and get exact delay
start_dt = datetime.strptime(row.ptd, '%H:%M')
end_dt = datetime.strptime(row.dep_at, '%H:%M')
diff = (end_dt - start_dt)
diff2 = (start_dt - end_dt)
if (diff.seconds < diff2.seconds):       ##convert betw
    final_delay = diff.seconds
else:
    final_delay = diff2.seconds
```

Figure 4: Example code of retrieving the amount of delay

One problem with the datetime function was that it could not differentiate the day (i.e. a delay of 23:55 and 01:00 would be 22 hours when it should be an hour and 5 minutes), so by taking the smallest difference between the two time values, this problem was avoided. The feature set containing the specific day of the week was used with the findDay method, which looked at the row id to determine which day it was as well as which month.

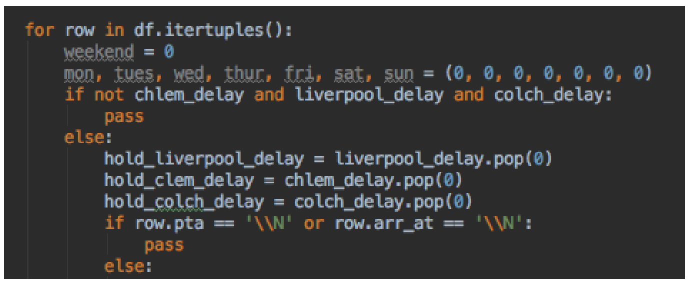
If there was a delay, then the delay feature that was in charge of holding a 1 for delay and 0 for non-delay held a 1. The Y vector held this delay (which could be 0) and was returned at the end of the model to be sent to the multi-layer perceptron. One added feature was the extra weight no delay feature, which held a 0 if there was no delay and was imperative for the model to always predict a 0 for the 0 delay.

Averages for the departure delay were calculated via the sortTimes-get-average method for each day of the week in order to see if it could help improve the performance of the model. An if statement checked to determine if variables saturday or sunday contained a 1, and if so then the peak value would hold a 0 (because there are no peak times on the weekend as determined by the Greater Anglia website), and the weekend feature would contain the average delay over the weekend.

Vector X is a two dimensional array that contained all the features that were chosen to be input into the neural network. One difference between the Liverpool and Norwich model was their limitation in terms of only being able to get either the departure delay (for Liverpool since all trains start at this station) or arrival delay (all trains arrive at Norwich). The other two stations were able to use both arrival and departure delays to help improve the performance of the model.

3.3 Secondary models

The secondary models also used the dataset for the first 6 months of 2018 and were named with the station name + model (so Norwich-model, Liverpool-model..) and shared the baseline features of the prior models. The main difference was that the secondary models were in charge of predicting arrival delay instead of departure delay, and used additional features such as the departure delay of the previous n models. In Figure 5, the final Norwich model contains three additional input features added to the baseline, each additional feature containing the departure delay from the previous stations.

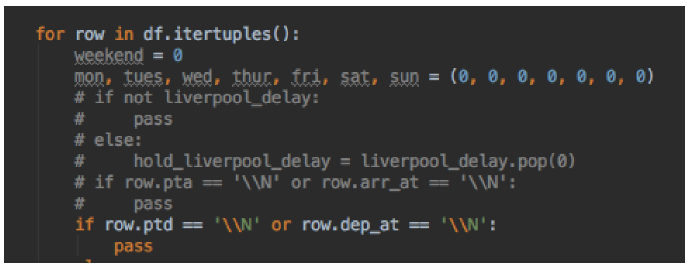


```
for row in df.itertuples():
    weekend = 0
    mon, tues, wed, thur, fri, sat, sun = (0, 0, 0, 0, 0, 0, 0)
    if not chlem_delay and liverpool_delay and colch_delay:
        pass
    else:
        hold_liverpool_delay = liverpool_delay.pop(0)
        hold_clem_delay = chlem_delay.pop(0)
        hold_colch_delay = colch_delay.pop(0)
        if row.pta == '\\N' or row.arr_at == '\\N':
            pass
        else:
```

Figure 5: Example code of the Norwich model with features given from the prior three models

The line: if not chlem-delay and liverpool-delay and colch-delay, determined that if journey in the dataset does not contain all three departure delays as well as the arrival time for the Norwich model, then skip that journey in the dataset. Consequently, this resulted in a limited data set which could have an effect on the performance of the model.

The y vector was changed to hold the arrival delay time instead of the departure delay time that was used in the original four models. One note when looking through the code is that as the experiments were conducted, after the model was trained and the next model was used, the planned and actual arrival time variables of the previous model were changed to planned and actual departure variables in order to be inputted into the next model (so in 6 the commented out code was the original code used on the Chlemsfield model before moving on to the Colchester model.



```
for row in df.itertuples():
    weekend = 0
    mon, tues, wed, thur, fri, sat, sun = (0, 0, 0, 0, 0, 0, 0)
    # if not liverpool_delay:
    #     pass
    # else:
    #     hold_liverpool_delay = liverpool_delay.pop(0)
    # if row.pta == '\\N' or row.arr_at == '\\N':
    #     pass
    if row.ptd == '\\N' or row.dep_at == '\\N':
        pass
```

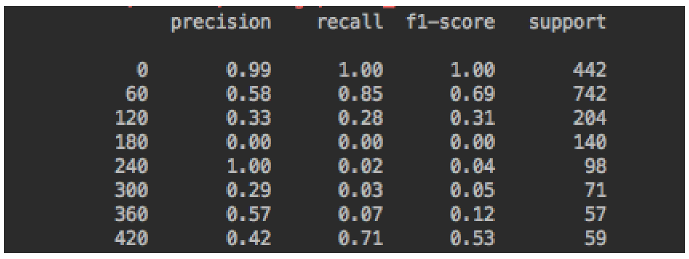
Figure 6: Example code of the Chlemsfield model but with changed variables

4 Results and Analysis

Precision, the ratio of correctly predicted positive observations to total predicted positive observations, recall, the ratio of correctly predicted positive observations to all observations in the class, and f1-score, the weighted average of precision and recall, were all used to categorize the performance of each model. A higher precision rate was preferred because it relates to low false positives, while a recall above .5 was also recommended to determine true positives. The f1-score is preferred over precision when there is an uneven class distribution, where false positives and false negatives have different costs, and was treated as the most important measurement in evaluating the quality of each model.

4.1 Liverpool Street Model

An initial feature set was used to establish a baseline accuracy by the MLP to determine changes needed to help improve accuracy. For the LIVST model, the days of the week, peak and off peak, delay or no delay, extra weight for no delay, and each hourly time were the 30 features used to send to the MLP (which subsequently had 30 hidden layers). The MLP's weighted average for its precision, recall, and f1-score was .60, .62, and .56 respectively.



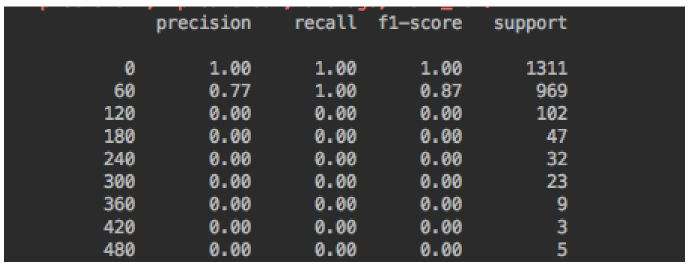
```
             precision    recall  f1-score   support

          0       0.99      1.00      1.00       442
         60       0.58      0.85      0.69       742
        120       0.33      0.28      0.31       204
        180       0.00      0.00      0.00       140
        240       1.00      0.02      0.04        98
        300       0.29      0.03      0.05        71
        360       0.57      0.07      0.12        57
        420       0.42      0.71      0.53        59
```

Figure 7: MLP initial results of the Liverpool model

It was able to correctly predict 442 results of a 0 delay with .99 precision, and 742 results of 60 second delays with a .58 precision, .85 recall, and .69 f1-score. Besides the 0 delay, the highest performing precision, recall, and f1-score measurement with at least 50 results was the 240 second delay at 1.00 precision, 60 second delay with .85 recall, and 60 second delay with a .69 f1 score. The 240 second delay, while at 1.00 precision, had an abysmal 0.2 recall and .04 f1 score, which demonstrates that the cost of false positives is larger than false negatives. Therefore, when it comes to predicting the 240 second delay, while the predictions for 240 second delay are valid, very few of these predictions actually capture the positive class. Using the f1-score, the best performing delay values the model had success predicting were (in order from highest to lowest) the 60 second delay and 420 second delay. Predicting the 900 second delay, which had a relatively small sample size of 19, also performed well, with an precision of .64, recall of .84, and f1 score of .73. The worst performing prediction with a large sample size was the 180 second delay, in which none of the 149 results were correctly predicted. One explanation could be that there was no discernible pattern in the data when a 180 delay occurred.

After the baseline results of the MLP were retrieved, the weighted average delay of each day was added as features to the MLP, with the idea that an average delay based on the day would help guide the MLP to correcting its predictions. The MLP performed notably worse: it had a precision of .54, recall of .61, and f1 score of .54. While the precision only had a drop off of .4, the recall dropped by .24, demonstrating that the model could not predict the positive class at a similar rate to the MLP with less features. The MLP was also run with the initial 30 hidden layers as well as 37 hidden layers but it was determined to have no effect upon performance. Therefore, it can be concluded that for the Liverpool model, the average delay for each day was not indicative of predicting specific delays for that day, as the delays were likely to vary widely for each day (and averaging these delays lead to no discernible information for the MLP to use).

Lastly, one additional feature was added to the baseline set of features that determined if it was the weekend or not, and if it were, to add the average time of delay for the weekend. This was done with the prediction that the delays on the weekend were noticeably different from the delay on the weekdays, which was seen during the calculations of the average delays for the LIVST. The precision, recall, and f1-score increased drastically at .80, .89, and .84 respectively. In Figure 8, the 60 second delay had a 1.00 recall and f1 score of .87, demonstrating that there is a correlation between a 60 second delay occurring on the weekend rather than the week. Consequently, the model however could not correctly predict any other delays, demonstrating that it would only perform best for predicting a 60 second delay.

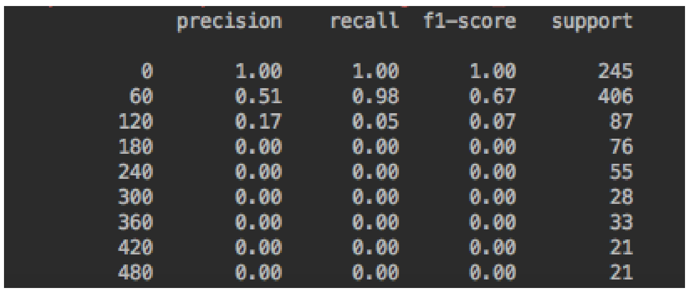


```
             precision    recall  f1-score   support

          0       1.00      1.00      1.00      1311
         60       0.77      1.00      0.87       969
        120       0.00      0.00      0.00       102
        180       0.00      0.00      0.00        47
        240       0.00      0.00      0.00        32
        300       0.00      0.00      0.00        23
        360       0.00      0.00      0.00         9
        420       0.00      0.00      0.00         3
        480       0.00      0.00      0.00         5
```

Figure 8: MLP results of Liverpool model with additional weekend feature added

4.2 Chelmsfield Model

The Chelmsfield model, used with the baseline set of features, performed substantially lower when compared to the Liverpool model; it had a precision, recall, and f1 score of .45, .61, and .49 respectively.

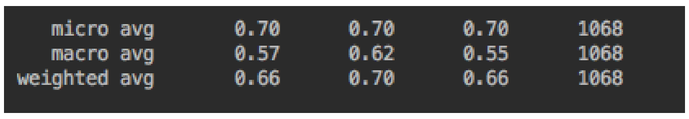


```
             precision    recall  f1-score   support

          0       1.00      1.00      1.00       245
         60       0.51      0.98      0.67       406
        120       0.17      0.05      0.07        87
        180       0.00      0.00      0.00        76
        240       0.00      0.00      0.00        55
        300       0.00      0.00      0.00        28
        360       0.00      0.00      0.00        33
        420       0.00      0.00      0.00        21
        480       0.00      0.00      0.00        21
```

Figure 9: MLP results of Chlemsfield model with baseline features

It noticeably under performed when predicting the 60 second delay, receiving a precision score of .51, .07 lower than the Liverpool model. It did, however, have a better recall at .98 when compared to Liverpool's .85. The only other delays it was able to predict was the 2,2280 second delay, the 120 second delay at a .17 precision rate, and 540 second delay at a 1.00 precision rate.

One attempt at improving the model was adding the arrival delay at the station. This was done with the reasoning that if there was an arrival delay, the departure delay would be around the same time (so if the train was delayed by a minute for arrival, the departure would subsequently be delayed by a minute). This intuition turned out to be correct, seen in Figure 8 below: the weighted average for precision, recall, and f1 score improved to .66, .70, and .66 respectively.



```
   micro avg       0.70      0.70      0.70      1068
   macro avg       0.57      0.62      0.55      1068
weighted avg       0.66      0.70      0.66      1068
```

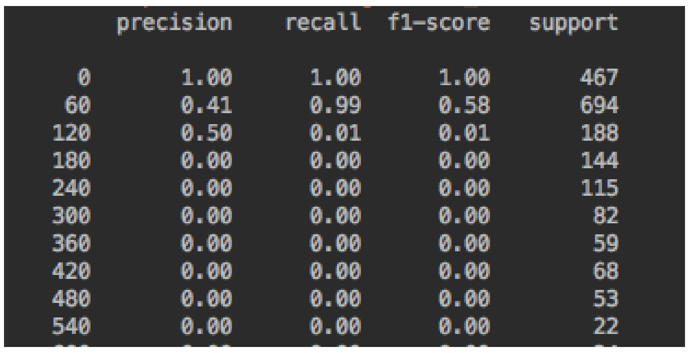
Figure 10: MLP results of Chlemsfield model with arrival delay feature

The highest prediction out of at least 50 results was the 60 second delay at .63 precision, while the 420 second delay had a notably high precision of .83 with 22 results. The largest delays 3540, 3480, 3000 all had precision, recall, and f1 scores of 1.00; therefore, it can be concluded that a high arrival delay at the station has a direct correlation to a high departing delay.

The average delay per day features were added to the model on top of the arrival delay feature to determine if the model could be improved, but similar to the Liverpool model, the precision, recall, and f1 score went down at .63, .66, and .61 respectively. Another attempt at improving the model involved averaging the weekend time and adding it as the feature just like the Liverpool model: precision, recall, and f1 score all performed worse than just with the arrival delay feature with a precision at .59, recall at .65, and f1 score of .59. Consequently, there is no link between a specific delay value occurring more often on the weekend for the Chlemsfield model.

4.3 Colchester Model

The baseline Colchester model received a weighted average precision, recall, and f1 score of .40, .54, and .41, which was noteably worse than the previous models. There were no correct predictions further than the 120 second delay, and it was able to correctly predict the 120 second delay with a .50 precision and a .01 recall, which suggest that it was predicting false positives. It had a .41 precision when predicting the 60 second delay, however it had a .99 recall, which gave it the best f1-score at .58. Consequently, the baseline features did not perform well when it came to predicting most delays, especially delays greater than 60 seconds.

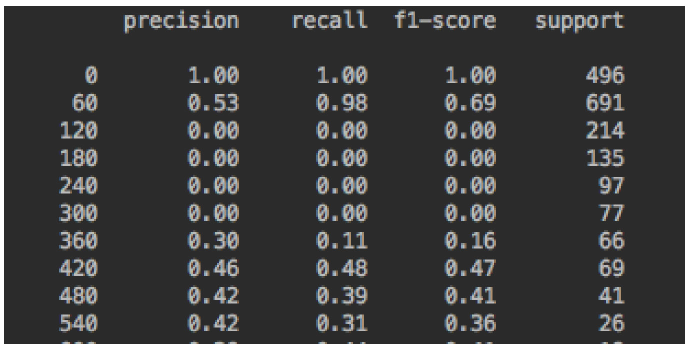


```
          precision    recall  f1-score   support

   0          1.00      1.00      1.00       467
  60          0.41      0.99      0.58       694
 120          0.50      0.01      0.01       188
 180          0.00      0.00      0.00       144
 240          0.00      0.00      0.00       115
 300          0.00      0.00      0.00        82
 360          0.00      0.00      0.00        59
 420          0.00      0.00      0.00        68
 480          0.00      0.00      0.00        53
 540          0.00      0.00      0.00        22
```

Figure 11: MLP results of Colchester model with baseline features

In order to improve performance, the average delay by day features were added to the MLP with the prediction that the model would be able to predict more delays. This however worsened its performance, giving a precision, recall, and f1 score of .38, .54, and .41, which suggests that the average delay has no impact in predicting delay for this model.

Next, the arrival delay value was used as a feature, and significantly improved the baseline model with a precision, recall, and f1 score of .50, .63, and .54. The 60 second delay achieved a recall of .98 with a precision of .53. The highest f1-score belonged to the 1440 and 1500 second delays with a score of .75 out of 10 and 5 results.



```
          precision    recall  f1-score   support

   0          1.00      1.00      1.00       496
  60          0.53      0.98      0.69       691
 120          0.00      0.00      0.00       214
 180          0.00      0.00      0.00       135
 240          0.00      0.00      0.00        97
 300          0.00      0.00      0.00        77
 360          0.30      0.11      0.16        66
 420          0.46      0.48      0.47        69
 480          0.42      0.39      0.41        41
 540          0.42      0.31      0.36        26
```

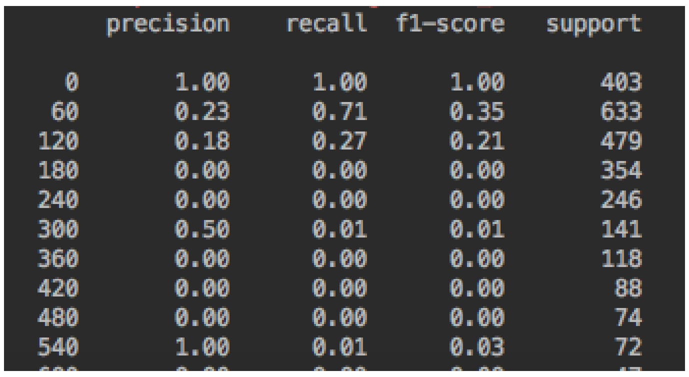
Figure 12: MLP results of Colchester model with arrival delay

It is clear when looking at Figure 12 above that the model was able to use the arrival delay value to correctly predict departing delays.

In an effort to approve the model further, a feature was set to 1 if there was an arrival delay (otherwise input a zero) to determine if there was a correlation between the significance of having an arrival delay and how it would affect the departing delay. This was able to give the Colchester model the best performance in terms of precision, with the precision, recall, and f1 score of .53, .60, and .54 respectively. While its precision increased by .03, its recall decreased by .03, which gave the model the same f1-score as the model with only the arrival delay value feature. The lack of improvement when judging the performance of the model in relation to the f1 score suggest that there is no improvement when adding this feature.

4.4 Norwich Model

The baseline features for the Norwich model performed the worst out of all four models, seen in Figure 13 below.



```
        precision    recall  f1-score   support

   0         1.00      1.00      1.00       403
  60         0.23      0.71      0.35       633
 120         0.18      0.27      0.21       479
 180         0.00      0.00      0.00       354
 240         0.00      0.00      0.00       246
 300         0.50      0.01      0.01       141
 360         0.00      0.00      0.00       118
 420         0.00      0.00      0.00        88
 480         0.00      0.00      0.00        74
 540         1.00      0.01      0.03        72
```

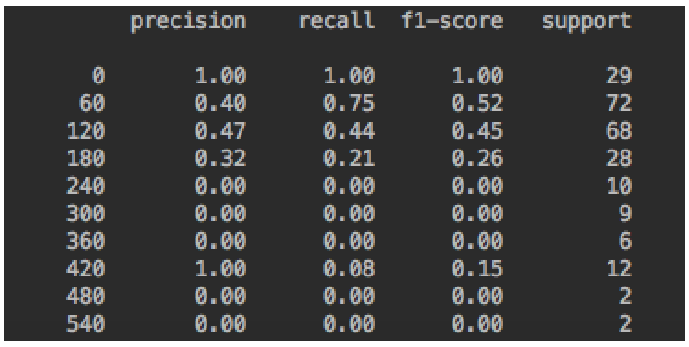
Figure 13: MLP results of Norwich model with baseline features

Its weighted average for precision, recall, and f1 score was at .25, .32, and .24 respectively. One immediate reasoning for its lack of success is that the delay arrival values are largely dependent on the prior stations; each station's overall baseline prediction, recall, and f1 scores went down as each station is further down the line in the journey. By adding the arrival delay as a feature, the other models were able to improve based off this value, but the Norwich model does not have that option. This problem leads to the development of using train journeys as a feature for the MLP, which will be discussed in the section below.

4.5 Train journeys and arrival delay

One significant issue that was discussed earlier in the report is each model's ability to correctly predict a delay as each station is further along in the route, due to a variance of delays. The solution for improving the declining baseline performance for each station was by using the previous model/s as a feature for the current model, which can be done by using the arrival delay as the delay that the model needs to predict, and have an additional feature from the previous model which contains the model's departing time. By using this method, each model is built upon each other with the goal of improving baseline performance for each station and not resulting in a gradual decline the further along the station is on the route.

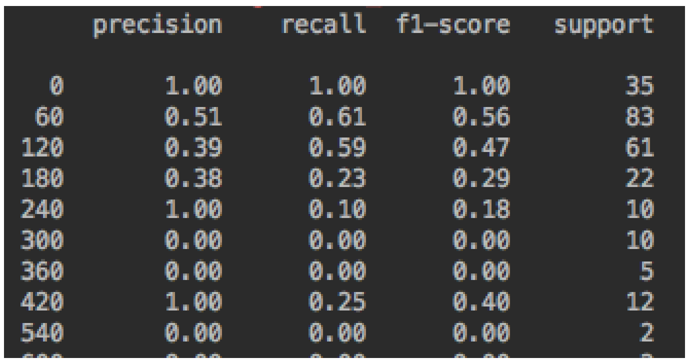
The Chelmsfield model used the baseline features without the departure delay from the LIVST model to determine its initial precision, recall, and f1 score. By filtering out unfinished values such as if there is no arrival value or departing value for a journey, the number of results was limited which has an impact in predicting the performance of the model. With the baseline features and no departure delay value from the Liverpool value, the model had a precision of .44, recall of .48, and f1-score of .43 out of 249 results. The model was only able to predict an arrival delay of 50 seconds with .40 precision, and a .75 recall out of 72 results, seen in Figure 14 below.



```
        precision    recall  f1-score   support

   0         1.00      1.00      1.00        29
  60         0.40      0.75      0.52        72
 120         0.47      0.44      0.45        68
 180         0.32      0.21      0.26        28
 240         0.00      0.00      0.00        10
 300         0.00      0.00      0.00         9
 360         0.00      0.00      0.00         6
 420         1.00      0.08      0.15        12
 480         0.00      0.00      0.00         2
 540         0.00      0.00      0.00         2
```

Figure 14: MLP results of Chlemsfield model without the depart delay value of the Norwich model

After adding the departing delay feature from the Norwich model, the Chelmsfield model noticeably improved with a precision, recall, and f1-score of .53, .53, and .49 respectively. The model was able to predict a 60 second arrival delay with a .51 precision and .61 recall, and was also able to predict the 120 second delay with a f1-score of .47, seen in Figure 15.

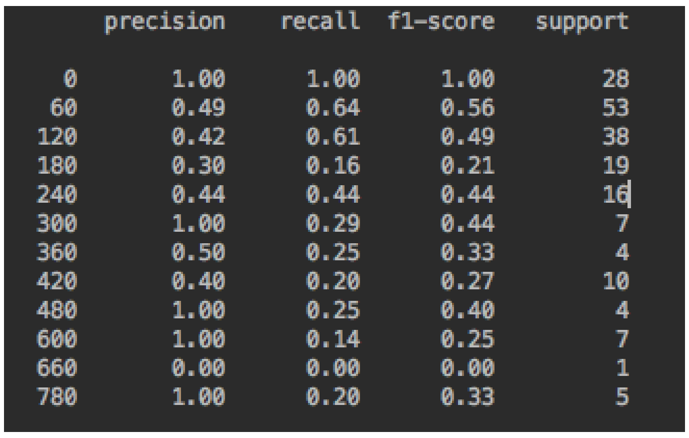


```
        precision    recall  f1-score   support

   0         1.00      1.00      1.00        35
  60         0.51      0.61      0.56        83
 120         0.39      0.59      0.47        61
 180         0.38      0.23      0.29        22
 240         1.00      0.10      0.18        10
 300         0.00      0.00      0.00        10
 360         0.00      0.00      0.00         5
 420         1.00      0.25      0.40        12
 540         0.00      0.00      0.00         2
```

Figure 15: MLP results of Chlemsfield model with the depart delay value of the Norwich model

This substantial improvement demonstrates that there is a correlation between the departing time delay of the previous model and can improve the performance of the current model.

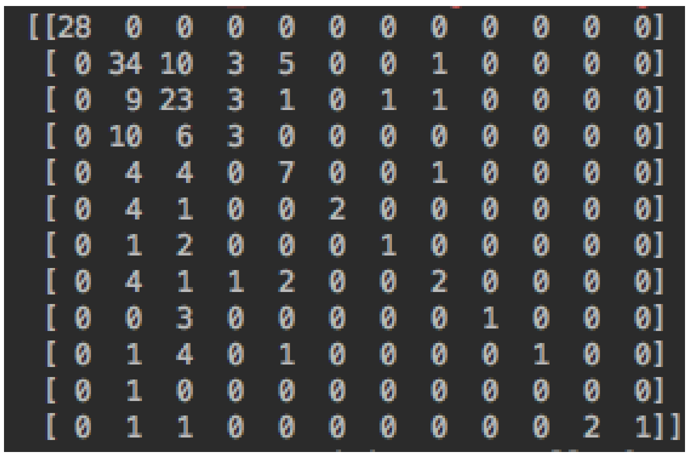
The Colchester model first uses the baseline features without any departing delay from a past model to determine its performance: it achieved a .37 precision rate, .48 recall, and .40 f1 score on 198 results. After including the departing delay feature from the previous Chlemsfield model, its performance rose to a a precision of .45, recall of .50, and .44 f1 score. While its performance did increase, the model was not at a good performance standard, and so a third final feature, the departing delay from the Liverpool model, was added to the model. This feature was able to improve the weighted average of the precision, recall, and f1 score to .58, .54, and .52 respectively, seen in Figure 16.



```
      precision    recall  f1-score   support

   0       1.00      1.00      1.00        28
  60       0.49      0.64      0.56        53
 120       0.42      0.61      0.49        38
 180       0.30      0.16      0.21        19
 240       0.44      0.44      0.44        16
 300       1.00      0.29      0.44         7
 360       0.50      0.25      0.33         4
 420       0.40      0.20      0.27        10
 480       1.00      0.25      0.40         4
 600       1.00      0.14      0.25         7
 660       0.00      0.00      0.00         1
 780       1.00      0.20      0.33         5
```

Figure 16: MLP results of the Colchester model with the depart delay value of the Liverpool and Chlemsfield model

In Figure 18, the confusion matrix is shown where each delay value corresponds to a marker on the x and y axis and the top left and bottom right value is at 0 delay, and bottom left and top right is the 780 delay.

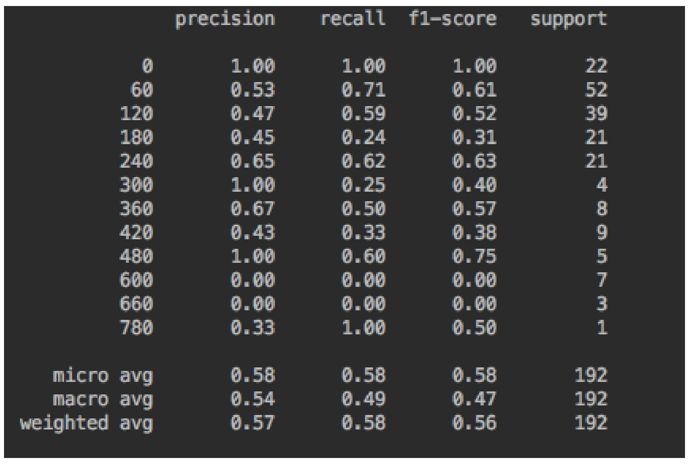


```
[[28  0  0  0  0  0  0  0  0  0  0  0]
 [ 0 34 10  3  5  0  0  1  0  0  0  0]
 [ 0  9 23  3  1  0  1  1  0  0  0  0]
 [ 0 10  6  3  0  0  0  0  0  0  0  0]
 [ 0  4  4  0  7  0  0  1  0  0  0  0]
 [ 0  4  1  0  0  2  0  0  0  0  0  0]
 [ 0  1  2  0  0  0  1  0  0  0  0  0]
 [ 0  4  1  1  2  0  0  2  0  0  0  0]
 [ 0  0  3  0  0  0  0  0  1  0  0  0]
 [ 0  1  4  0  1  0  0  0  0  1  0  0]
 [ 0  1  0  0  0  0  0  0  0  0  0  0]
 [ 0  1  1  0  0  0  0  0  0  0  2  1]]
```

Figure 17: Confusion Matrix results of the Colchester model with the depart delay value of the Liverpool and Chlemsfield model

The model was able to predict higher delay values that it previously didn't when given only the Chlemsfield departing delay, so it can be inferred that both departing values from the previous models indicate a pattern for a specific delay occurring.

In the final Norwich model, it was noted that the precision varied wildly due to a small relative data set; in order to more accurately predict the performance of the system, the recall and F1-score was used in measuring performance. The baseline features had a starting precision, recall, and f1-delay of .38, .46, and .40 out of 192 data results. After the Colchester departing delay feature was added, it received a score of .41, .47, and .42, demonstrating that the recall and f1-delay did improve based off the past model. Including the Chlemsfield departing delay also improved the model to a score of 0.45, 0.48, and 0.45. Lastly, adding the final feature of the Liverpool departing delay led to a final performance of .57, .58, and .56, seen in Figure 18 below, which gave it a better performance than the initial Norwich model.



```
              precision    recall  f1-score   support

           0       1.00      1.00      1.00        22
          60       0.53      0.71      0.61        52
         120       0.47      0.59      0.52        39
         180       0.45      0.24      0.31        21
         240       0.65      0.62      0.63        21
         300       1.00      0.25      0.40         4
         360       0.67      0.50      0.57         8
         420       0.43      0.33      0.38         9
         480       1.00      0.60      0.75         5
         600       0.00      0.00      0.00         7
         660       0.00      0.00      0.00         3
         780       0.33      1.00      0.50         1

   micro avg       0.58      0.58      0.58       192
   macro avg       0.54      0.49      0.47       192
weighted avg       0.57      0.58      0.56       192
```

Figure 18: Results of the Norwich model with the departing delays of all the previous models as features

The model proved to have a higher precision, recall, and f1 score, suggesting that there is a a link between the departing delay of the previous models and the arrival delay amount of the current. There are, however, several limitations with this approach; the small sample size, which is around 1/5th the size of the original Norwich model, demonstrates that these results might not hold up over a larger data set. The small sample size is due to the need for there to be a departing delay for each journey for all three previous models, which means that the data set was only able to find 192 rows that satisfied this criteria.

Another limitation that could have occurred was overfitting: the signal is the underlying pattern that the data has while the noise refers to the randomness of the data. The concern is that the algorithm memorizes the noise which will make predictions based on that subsequent noise. It would have been interesting to test the model on unseen data from the months in 2017 instead of 2018 and see how it did. Attempts to mitigate overfitting involved creating an x and y train and test split for each variable, so that the data isn't seen. Part of the problem with the MLPClassifier is that it is difficult to interprate the model:

the weights and biases won't be easily interpretable in relation to which features are important to the model itself (pi19404, 2014). The data set's biggest quantity was 2000, which is rather limited compared to the entirety of the data (since if there wasn't an actual arrival or delay value in the row, then the row was ignored). The MLP relies on having thousands of results per class due to its high-variance nature since it uses predictions based on specific data. Therefore, the MLP costs a lot more data in order to give good performance, which could have changed the results by a substantial amount.

5 Conclusion

In this paper, the performance of the eight models were evaluated to see which feature sets had the best precision, recall, and f1 score. The original Liverpool model had the highest score at .80, .89, and .84 respectively. This was largely due to the added average weekend delay time feature which found that a delay of 60 was most likely to occur on the weekend. It was noted that the other stations baseline feature results were substantially lower the further along the station was in the journey, which was due to the large variance in delay times. The weighted average delay features made each model notably worse, and demonstrated that they had no effect in improving performance. By adding the arrival delay time to the original Chlemsfield and Colchester models, both models had improved performance based on f1 scores of .66 and .54. The last original Norwich model had the worse performance out of the original models, and subsequently led to the thinking of adding prior model departure values in order to predict the arrival times at the stations. Both Chlemsfield and Colchester models achieved scores of .52 and .56 f1 scores when using the feature/s of the prior models in terms of departure delay. In the original model predictions, the stations further along in the journey had a noticeably worse performance than the previous, but due to this added feature, the further along a station, the better or equal its performance, which is demonstrated by the final Norwich model which had a precision, recall, and f1 score of .57, .58, and .56. It is therefore reasoned that adding features that contain data from prior models will help the performance of the multi-layer perceptron in predicting delays. Limitations include the limited data set that was a consequence of using prior model features, as well as over-fitting the data in which the model is training the noise of the data. One possible solution was to test the MLP on the 2017 data.

References

Brownlee, J. (2018). How to configure the number of layers and nodes in a neural network.

pi19404 (2014). Multilayer perceptron in python.

Portilla, J. (2018). A beginner's guide to neural networks in python and scikit learn 0.18.