Predicting Taxi Fares with Deep Feedforward Networks

We will create a model that will provide fare pricing upfront before a client hail a cab with consideration of various environmental factors such as traffic condition, time of day and pick up and drop off locations .

The dataset we will use is provided be Kaggle and contains trip records from New York City. Because the dataset contains 55 million trips, we will use only one million to train our model.

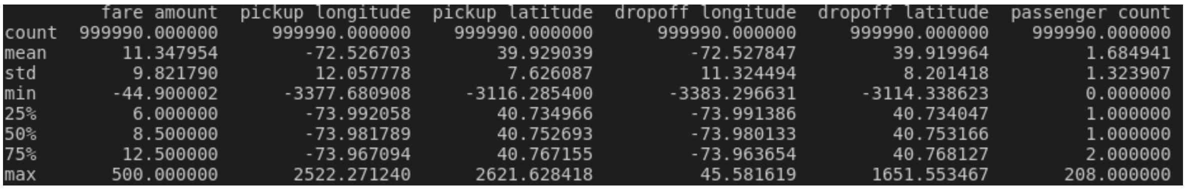
There are 8 columns in the dataset:

- *fare_amount* : This is the target variable we are trying to predict.
- *pickup_datetime* : This column contains information on the pickup date (year, month, day of month), as well as the time (hour, minutes, seconds).
- *pickup_longitude* and *pickup_latitude* : The longitude and latitude of the pickup location.
- *dropoff_longitude* and *dropoff_latitude* : The longitude and latitude of the drop off location.
- *passenger_count* : The number of passengers
- *key* : identical to *pickup_datetime* so we will remove it.

Data Preprocessing

First of all, we will check if there are missing values and we will remove them.

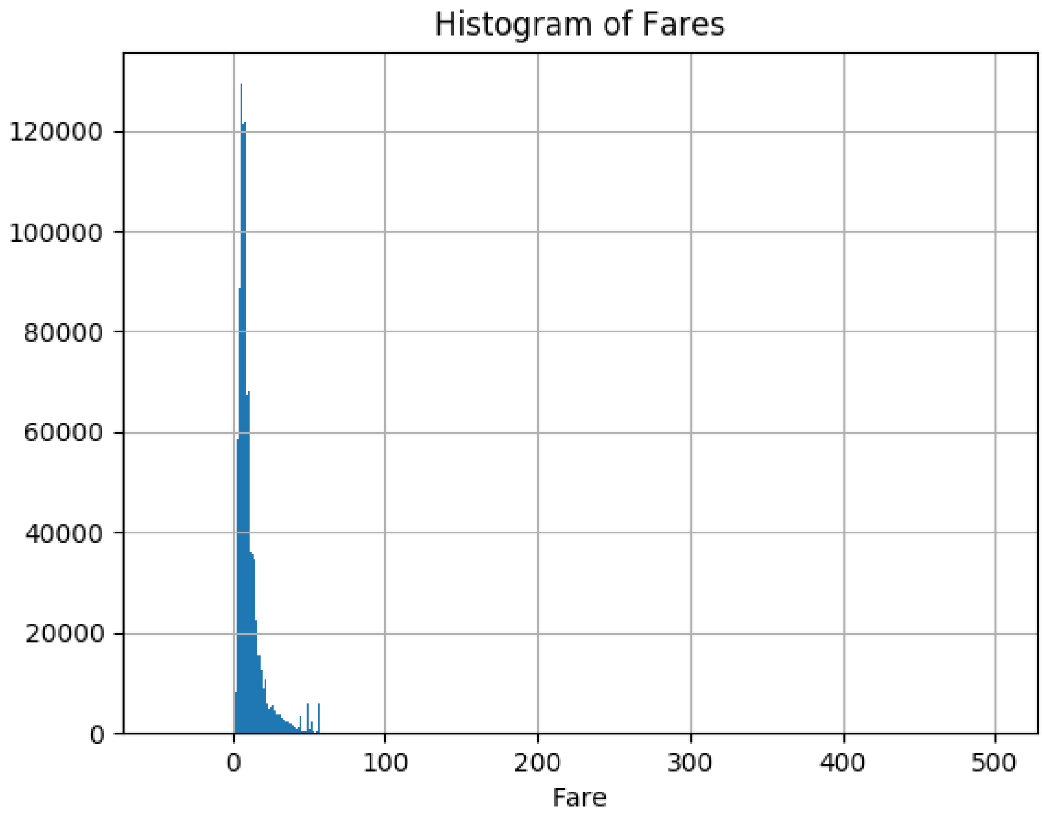
The information for our dataset is showed below:



```
        fare amount  pickup longitude  pickup latitude  dropoff longitude  dropoff latitude  passenger count
count  999990.000000     999990.000000    999990.000000      999990.000000     999990.000000    999990.000000
mean       11.347954        -72.526703        39.929039         -72.527847         39.919964         1.684941
std         9.821790         12.057778         7.626087          11.324494          8.201418         1.323907
min       -44.900002      -3377.680908     -3116.285400       -3383.296631      -3114.338623         0.000000
25%         6.000000        -73.992058        40.734966         -73.991386         40.734047         1.000000
50%         8.500000        -73.981789        40.752693         -73.980133         40.753166         1.000000
75%        12.500000        -73.967094        40.767155         -73.963654         40.768127         2.000000
max       500.000000       2522.271240      2621.628418          45.581619       1651.553467       208.000000
```

As showed from dataset description, we must evaluate each variable for anomalies.

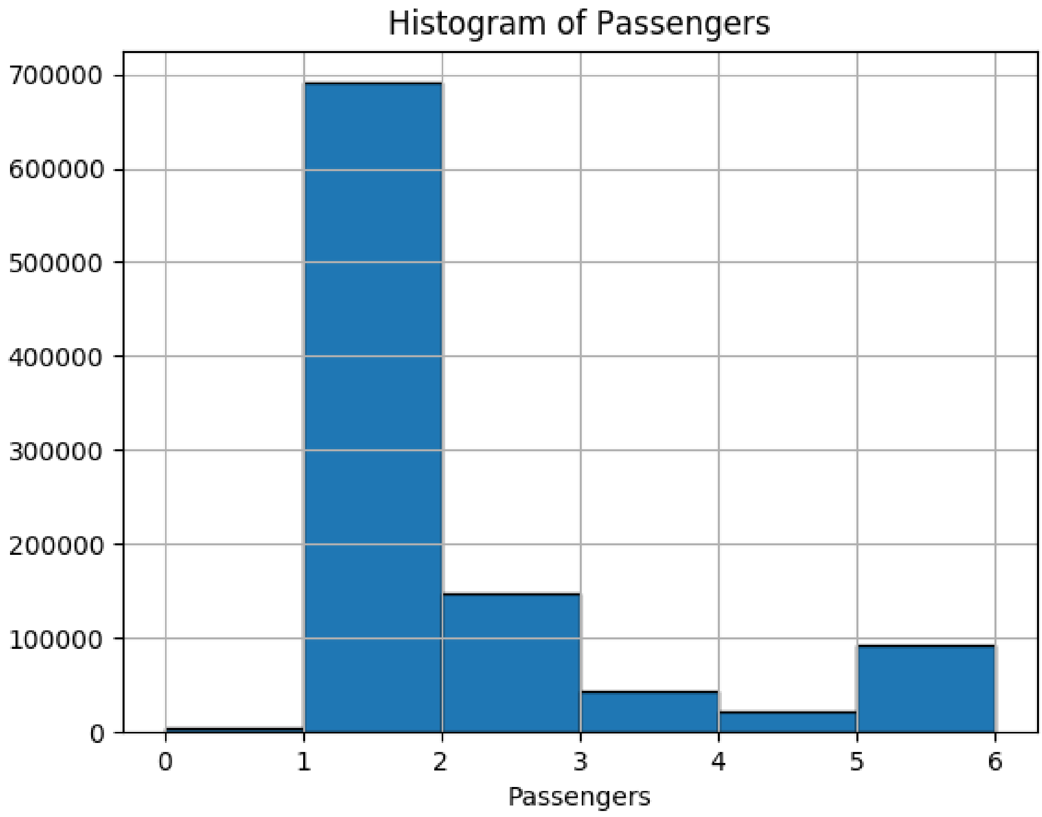
Starting from *fare_amount*, we can see that it has minimum value at -44.9 $ and maximum value at 500 $. By plotting its histogram, as shown below, it is safe to assume that *fare_amount* value must be in [0.0,100.0] range. So, any value that is not in [0.0,100.0] is removed.



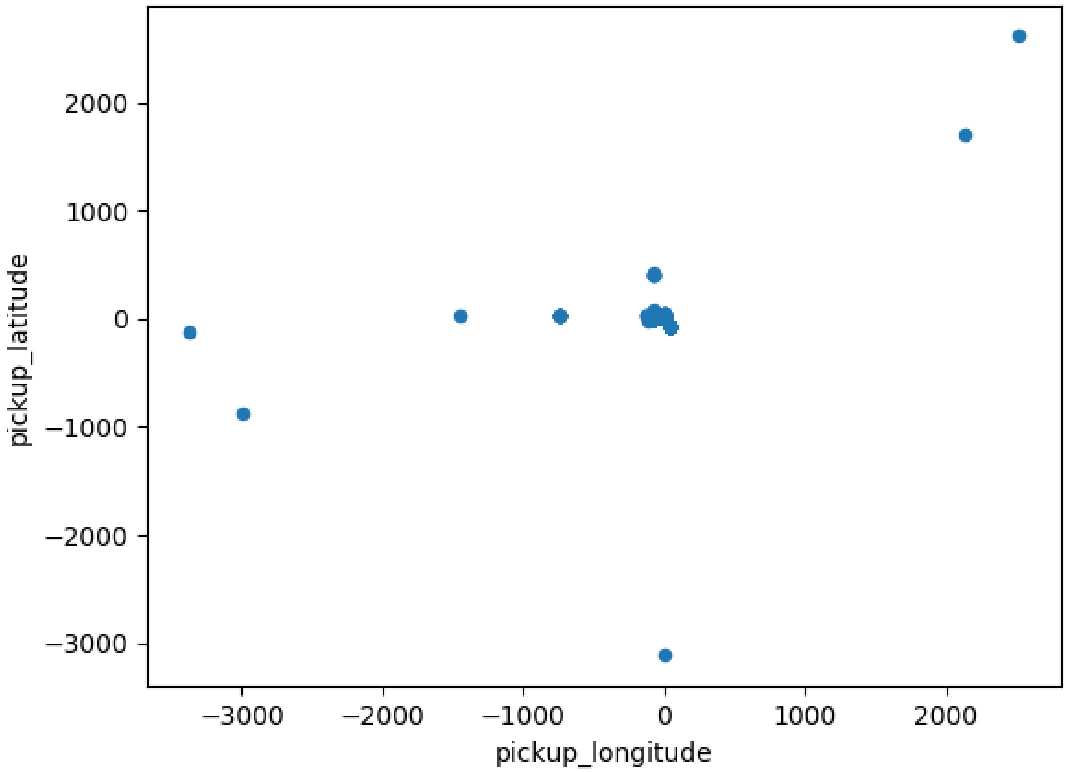
By **Histogram of Fares**, we observe that there is a small spike at 50$ that can be used at feature extraction.

From description above, we see some outliers at *passenger_count*. The maximum values is set at 208, which is unbelievable. So we have to remove values that are bigger than 8 passengers.

From *passenger_count* histogram, we can observe a small percentage of values with 0 passengers. Instead of removing them, we will replace them with 1 passenger.

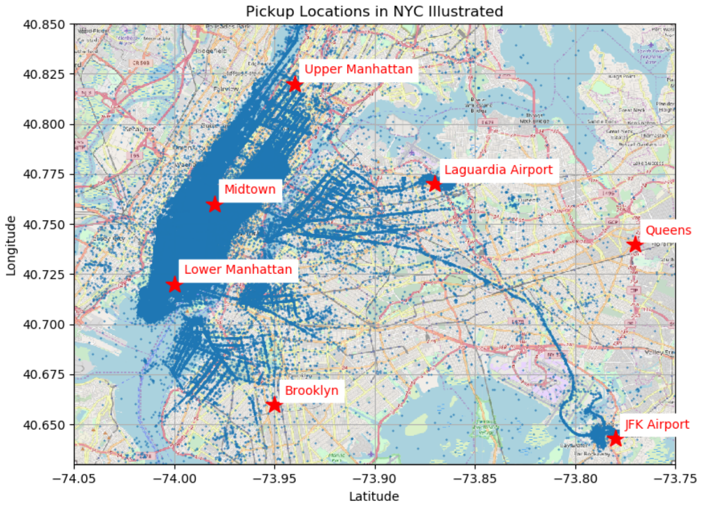


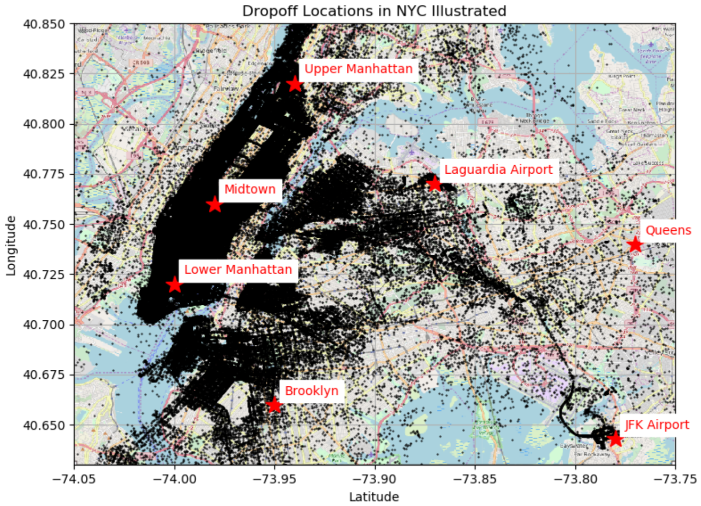
Lastly, in order to inspect the latitude and longitude values, we have to plot the scatterplot of those variables.



We can observe extreme values such as -3000 and 2000 that are not existing. The outliers of New York City longitude and latitude are in range of [-74.05 , -73.75] and [40.63 , 40.85] so we have to remove the entries that are not inside those limits.

The actual pickup and drop off locations with a few of New York's famous landmarks are shown below:



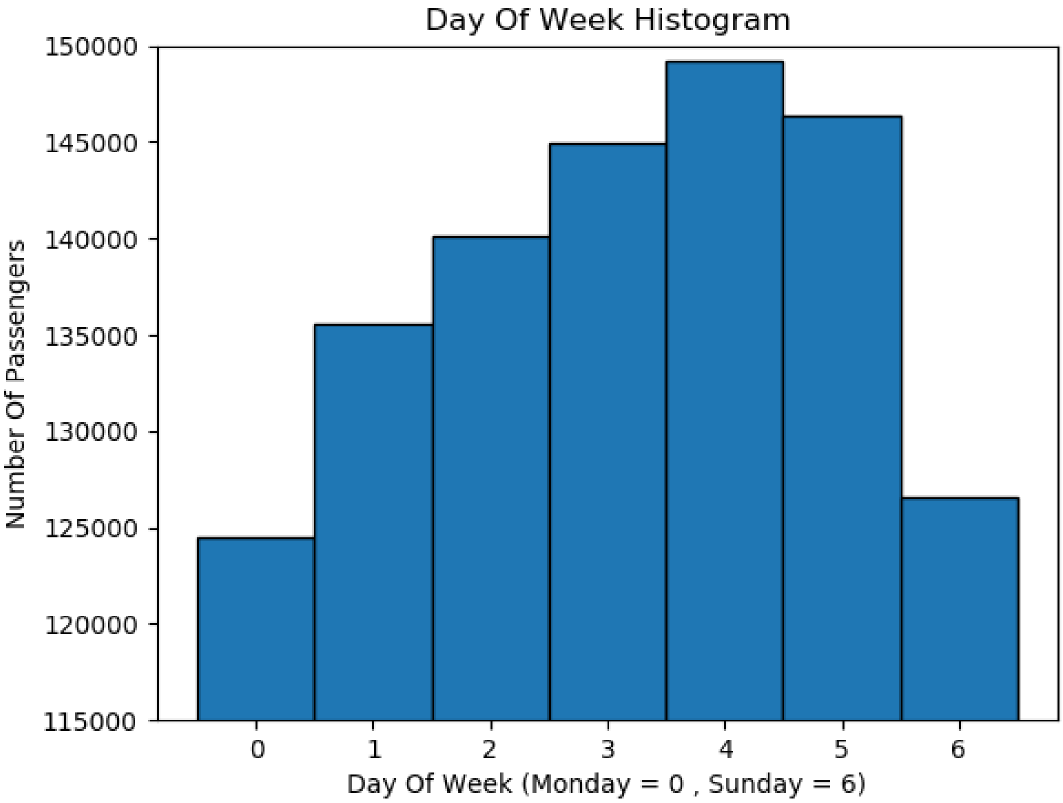


Feature engineering

We create new features for our dataset based on pickup date and time and location based features.

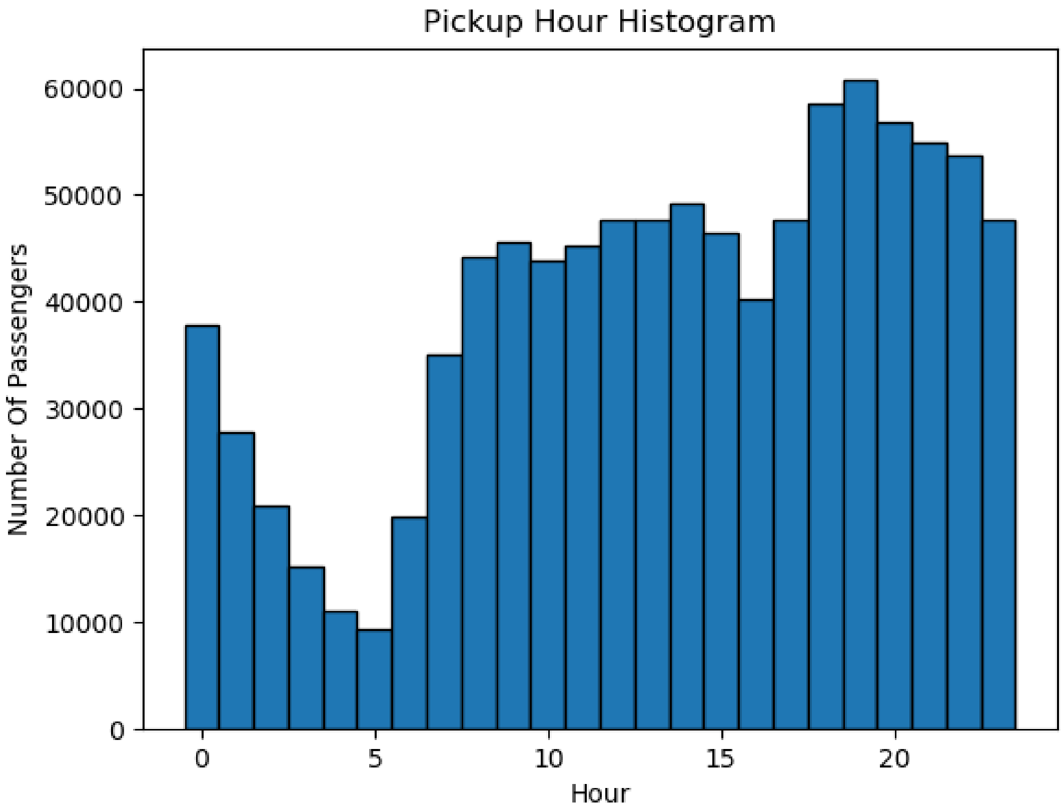
We will separate the *pickup_datetime* column into *year*, *month*, *day*, *day_of_week* and *hour* columns in order to find a correlation between taxi schedules and days of week and another correlation with hour of departure.

The histogram for week days is shown below:



We can see that the number of rides is not evenly distributed across each weekday. Instead, the number of rides increases linearly from Monday through Friday, and peaking on Friday. The weekends see a slight drop in the number of rides on Saturday, before falling sharply on Sunday.

The histogram for daytime hours is shown below:



We can see that there are more rides during the evening rush hour, as compared to the morning rush hour. In fact, the number of rides is pretty constant throughout the day. Starting at 06:00, the number of rides increases and peaks at 19:00 , before falling from 23:00 onwards.

Since the dataset contains pickup and drop off coordinates, we have to calculate the distance between those coordinates. Our hypothesis is that the distance will be closely correlate with the taxi fare.

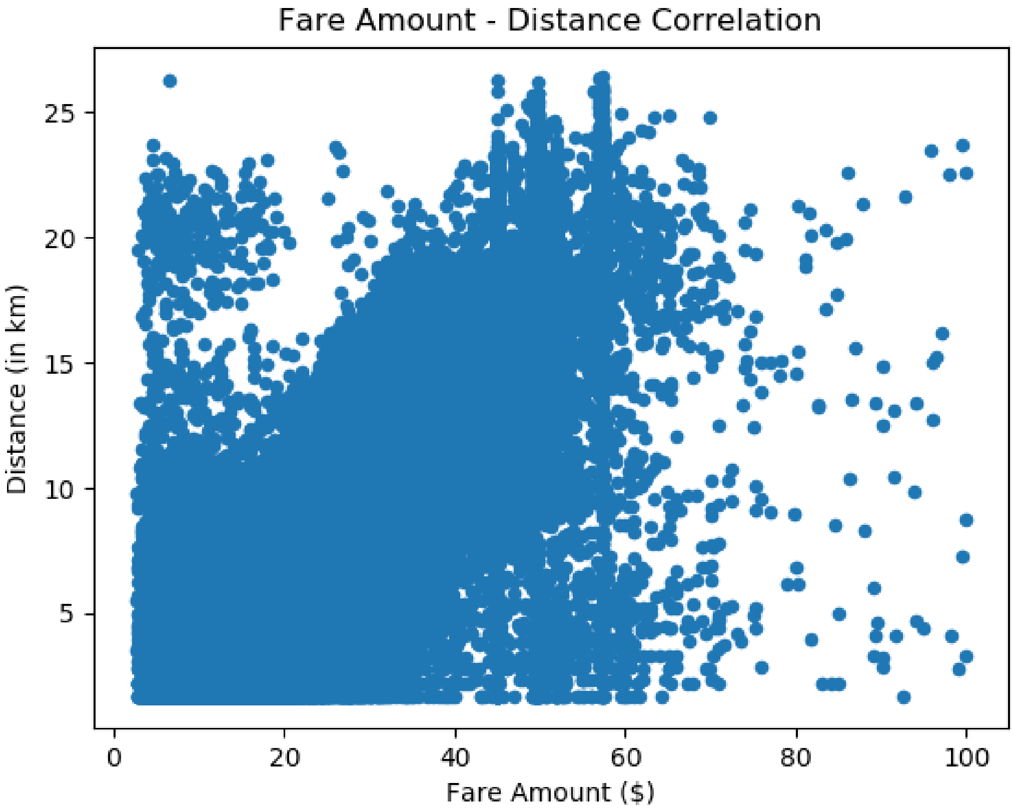
As for the distance function, we will use the **Haversine Distance** (https://en.wikipedia.org/wiki/Haversine_formula). The distance formula (in km) is :

$$d = 2r \arcsin\left(\sqrt{\sin^2\left(\frac{\phi_2 - \phi_1}{2}\right) + \cos(\phi_1)\cos(\phi_2)\sin^2\left(\frac{\lambda_2 - \lambda_1}{2}\right)}\right)$$

where:

$$r \rightarrow Earth\ radius\ (in\ km)$$
$$\phi_1, \phi_2 \rightarrow pickup\ and\ drop\ off\ latitudes\ (in\ radians)$$
$$\lambda_1, \lambda_2 \rightarrow pickup\ and\ drop\ off\ longitudes\ (in\ radians)$$

By defining the distance function, we get the following scatterplot:



We can see that our hypothesis is right. However, if we look at center of this plot, there are three vertical lines between 40$ and 60$. These data seems to suggest that there are certain trips where the distance traveled did not have an impact on the fare amount.

Clearly, we need to engineer a new feature that informs our neural network of the pickup and drop off distance from the major New York City airports. The major airports latitudes and longitudes are showed below

```
airports = {'JFK Airport': (-73.78, 40.643),
            'Laguardia Airport': (-73.87, 40.77),
            'Newark_Airport' : (-74.18, 40.69)}
```

Our dataset now contains the *pickup* and *dropoff* distances from both of airports.

Also, we see that minimum fare is 2.5$ and there are no more than 70$ courses in NYC the last 5 years. So, we modify our *fare_amount* features in those limits.

As a final step, we will have to scale our features before passing them to the neural network. In order to have good predictions, we have to leave *fare_amount* parameter as it is.

Neural Network Model

For this problem, we will use a deep feedforward network with **five hidden layers**. The first layer will have 128 nodes, with each successive hidden layer having halve of the nodes of its predecessor.

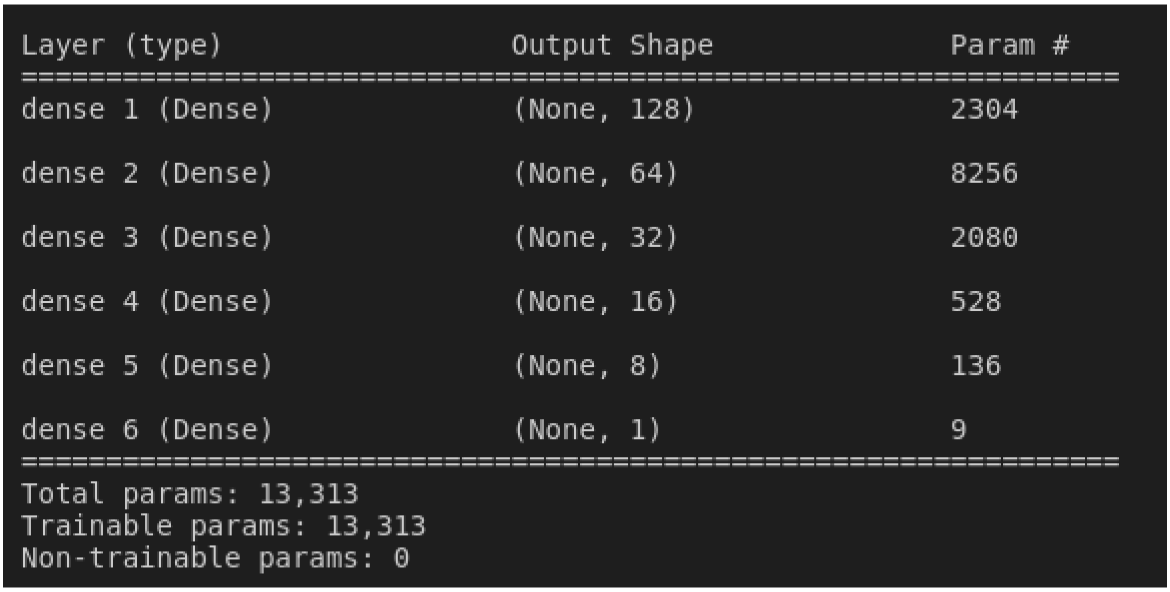
In between each layer, we will use the ***relu*** activation function to introduce non linearity in the model.

Since it is a regression problem, we will use only **one output layer**. In regression, we are trying to predict the value of a continuous variable.

For loss function, we are going to use the **root mean square error (RMSE)**. The formula for RMSE is as follows:

$$RMSE = \sqrt{(actual - predict)^2}$$

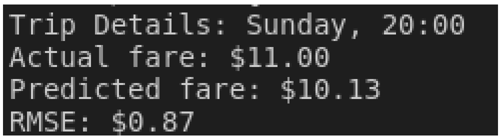
The model that will evaluate fare predictions is shown below:



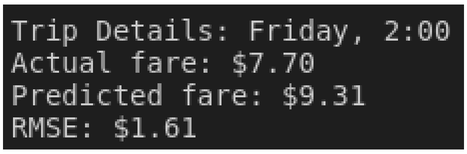
```
Layer (type)                 Output Shape              Param #
=================================================================
dense 1 (Dense)              (None, 128)               2304

dense 2 (Dense)              (None, 64)                8256

dense 3 (Dense)              (None, 32)                2080

dense 4 (Dense)              (None, 16)                528

dense 5 (Dense)              (None, 8)                 136

dense 6 (Dense)              (None, 1)                 9
=================================================================
Total params: 13,313
Trainable params: 13,313
Non-trainable params: 0
```

Now that our network is trained, we will use it to make predictions to understand its accuracy.

The trip details are showed below:



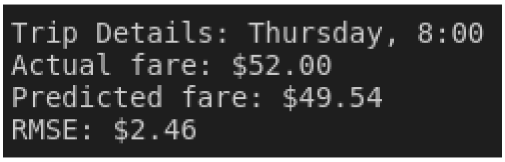
```
Trip Details: Sunday, 20:00
Actual fare: $11.00
Predicted fare: $10.13
RMSE: $0.87
```



```
Trip Details: Friday, 2:00
Actual fare: $7.70
Predicted fare: $9.31
RMSE: $1.61
```

We see that our network made fairly good predictions for those trips above.

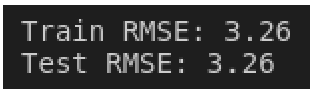
For fixed priced trips, such as those to JFK Airport that have fixed fare cost 52$, our network prediction is showed below



```
Trip Details: Thursday, 8:00
Actual fare: $52.00
Predicted fare: $49.54
RMSE: $2.46
```

Our neural network understands that the trip started from JFK airport, and hence the fare should be close to $52 . This was made possible through feature engineering, where we introduced new features that represents the pickup and drop off distance away from JFK Airport.

Finally, our network overall RMSE is :



```
Train RMSE: 3.26
Test RMSE: 3.26
```