NYC Taxi Fare: Deep Learning

Brashon Ford

Setup

Library import

We import all the required Python libraries

In [ ]:
```
pip install utils
```

In [2]:
```
# Data manipulation
import pandas as pd
import numpy as np

# Options for pandas
pd.options.display.max_columns = 50
pd.options.display.max_rows = 30

# Visualizations
import plotly
import plotly.graph_objs as go
import plotly.offline as ply
plotly.offline.init_notebook_mode(connected=True)



import matplotlib as plt
%matplotlib inline

import matplotlib.pyplot as plt
import numpy as np
import matplotlib
matplotlib.use("TkAgg")
import pandas as pd
import numpy as np
from sklearn.preprocessing import scale
from sklearn.model_selection import train_test_split
from keras.models import Sequential
from keras.layers import Dense
from keras.layers import BatchNormalization
```

```
from sklearn.metrics import mean_squared_error
```

In [3]:
```
def preprocess(df):
    # remove missing values in the dataframe
    def remove_missing_values(df):
        df = df.dropna()
        return df
```

In [4]:
```
# remove outliers in fare amount
def remove_fare_amount_outliers(df, lower_bound, upper_bound):
    df = df[(df['fare_amount'] > lower_bound) & (df['fare_amount'] <= upper_
    return df
```

In [ ]:
```
def feature_engineer(df):
    # create new columns for year, month, day, day of week and hour
def create_time_features(df):
    df['year'] = df['pickup_datetime'].dt.year
    df['month'] = df['pickup_datetime'].dt.month
    df['day'] = df['pickup_datetime'].dt.day
    df['day_of_week'] = df['pickup_datetime'].dt.dayofweek
    df['hour'] = df['pickup_datetime'].dt.hour
    df = df.drop(['pickup_datetime'], axis=1)
return df
```

Data import

We retrieve all the required data for the analysis.

In [5]:
```
import os
os.getcwd()
```

Out[5]: '/Users/brashonford/new-york-city-taxi-fare-prediction'

In [6]:
```
df = pd.read_csv('/Users/brashonford/new-york-city-taxi-fare-prediction/NYC_
```

Data processing / Handling Missing Values

In [7]:
```
df.head()
```

Out[7]:

	key	fare_amount	pickup_datetime	pickup_longitude	pickup_latitu
0	2009-06-15 17:26:21.0000001	4.5	2009-06-15 17:26:21+00:00	-73.844311	40.7213
1	2010-01-05 16:52:16.0000002	16.9	2010-01-05 16:52:16+00:00	-74.016048	40.7113
2	2011-08-18 00:35:00.00000049	5.7	2011-08-18 00:35:00+00:00	-73.982738	40.7612
3	2012-04-21 04:30:42.0000001	7.7	2012-04-21 04:30:42+00:00	-73.987130	40.7331
4	2010-03-09 07:51:00.000000135	5.3	2010-03-09 07:51:00+00:00	-73.968095	40.7680

In [8]:
```
print(df.isnull().sum())
print('')
```

```
key                  0
fare_amount          0
pickup_datetime      0
pickup_longitude     0
pickup_latitude      0
dropoff_longitude    5
dropoff_latitude     5
passenger_count      0
dtype: int64

key                  0
fare_amount          0
pickup_datetime      0
pickup_longitude     0
pickup_latitude      0
dropoff_longitude    5
dropoff_latitude     5
passenger_count      0
dtype: int64
```

In [9]:
```
df = df.dropna()

print(df.describe())
```

```
         fare_amount  pickup_longitude  pickup_latitude  dropoff_longitude
dropoff_latitude  passenger_count
count  499995.000000     499995.000000    499995.000000      499995.000000
499995.000000    499995.000000
mean       11.358182        -72.520091        39.920350         -72.522435
39.916526         1.683445
std         9.916069         11.856446         8.073318          11.797362
7.391002         1.307391
min       -44.900000      -2986.242495     -3116.285383       -3383.296608
-2559.748913         0.000000
25%         6.000000        -73.992047        40.734916         -73.991382
40.734057         1.000000
50%         8.500000        -73.981785        40.752670         -73.980126
40.753152         1.000000
75%        12.500000        -73.967117        40.767076         -73.963572
40.768135         2.000000
max       500.000000       2140.601160      1703.092772          40.851027
404.616667          6.000000
```


```
         fare_amount  pickup_longitude  pickup_latitude  dropoff_longitude
dropoff_latitude  passenger_count
count  499995.000000     499995.000000    499995.000000      499995.000000
499995.000000    499995.000000
mean       11.358182        -72.520091        39.920350         -72.522435
39.916526         1.683445
std         9.916069         11.856446         8.073318          11.797362
7.391002         1.307391
min       -44.900000      -2986.242495     -3116.285383       -3383.296608
-2559.748913         0.000000
25%         6.000000        -73.992047        40.734916         -73.991382
40.734057         1.000000
50%         8.500000        -73.981785        40.752670         -73.980126
40.753152         1.000000
75%        12.500000        -73.967117        40.767076         -73.963572
40.768135         2.000000
max       500.000000       2140.601160      1703.092772          40.851027
404.616667          6.000000
```


In [10]:

```python
import matplotlib.pyplot as plt
df['fare_amount'].hist(bins=500)
plt.xlabel("Fare")
plt.title("Histogram of Fares")
plt.show()
```

In [11]:

```python
df = df[(df['fare_amount'] >=0) & (df['fare_amount'] <= 100)]
```

In [12]:

```python
df['passenger_count'].hist(bins=6, ec='black')
plt.xlabel("Passenger Count")
plt.title("Histogram of Passenger Count")
plt.show()
```

In [13]:
```
df.loc[df['passenger_count']==0, 'passenger_count'] = 1
```

Details About The Data

key: This column seems identical to the pickup_datetime column. It was probably used as an unique identifier in the database it was stored in. We can safely remove this column without any loss of information.

fare_amount: This is the target variable we are trying to predict, the fare amount paid at the end of the trip.

pickup_datetime: This column contains information on the pickup date (year, month, day of month), as well as the time (hour, minute, 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.

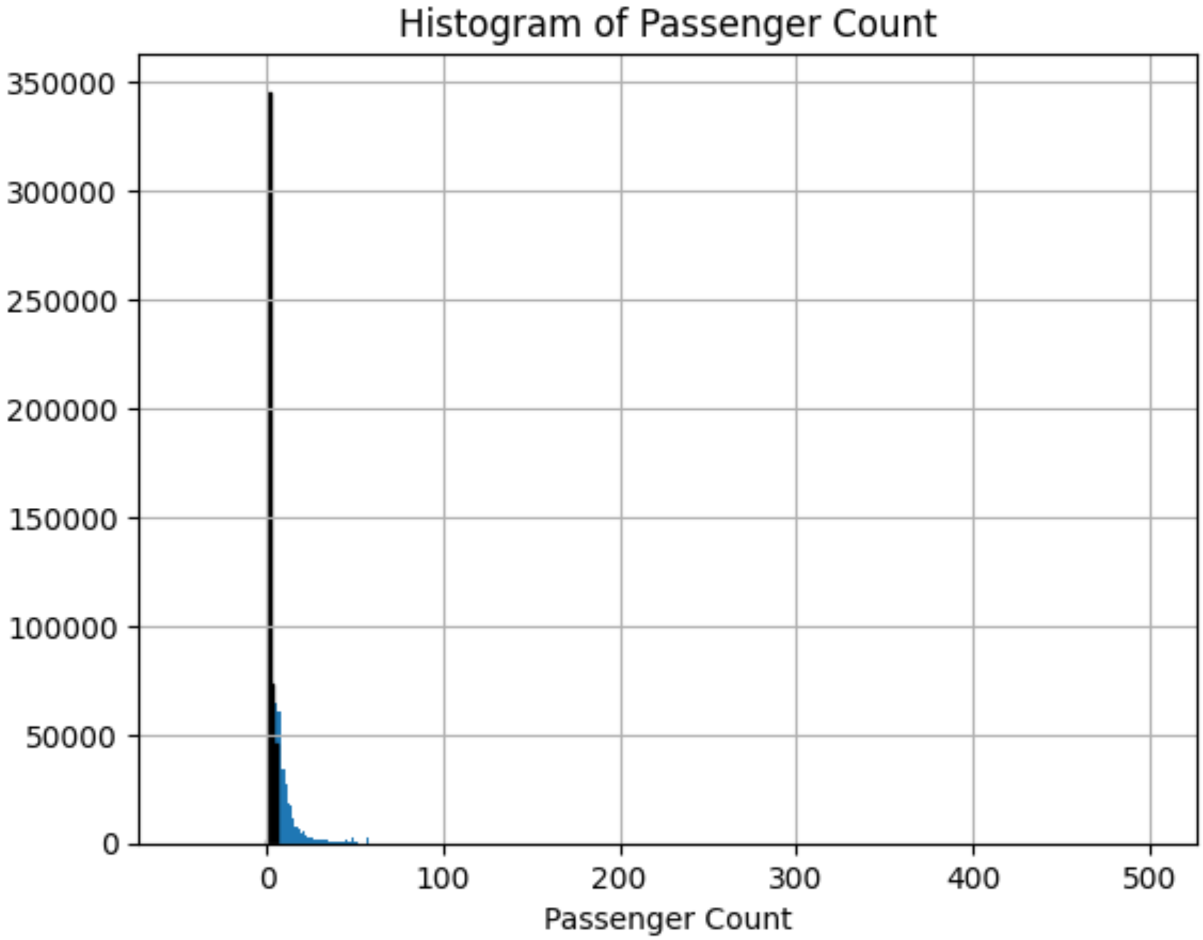
In [14]:
```
# range of longitude for NYC
nyc_min_longitude = -74.05
nyc_max_longitude = -73.75

# range of latitude for NYC
nyc_min_latitude = 40.63
nyc_max_latitude = 40.85

df2 = df.copy(deep=True)

for long in ['pickup_longitude', 'dropoff_longitude']:
    df2 = df2[(df2[long] > nyc_min_longitude) & (df2[long] <
                                                  nyc_max_longitude)]

for lat in ['pickup_latitude', 'dropoff_latitude']:
    df2 = df2[(df2[lat] > nyc_min_latitude) & (df2[lat] <
                                                nyc_max_latitude)]
```

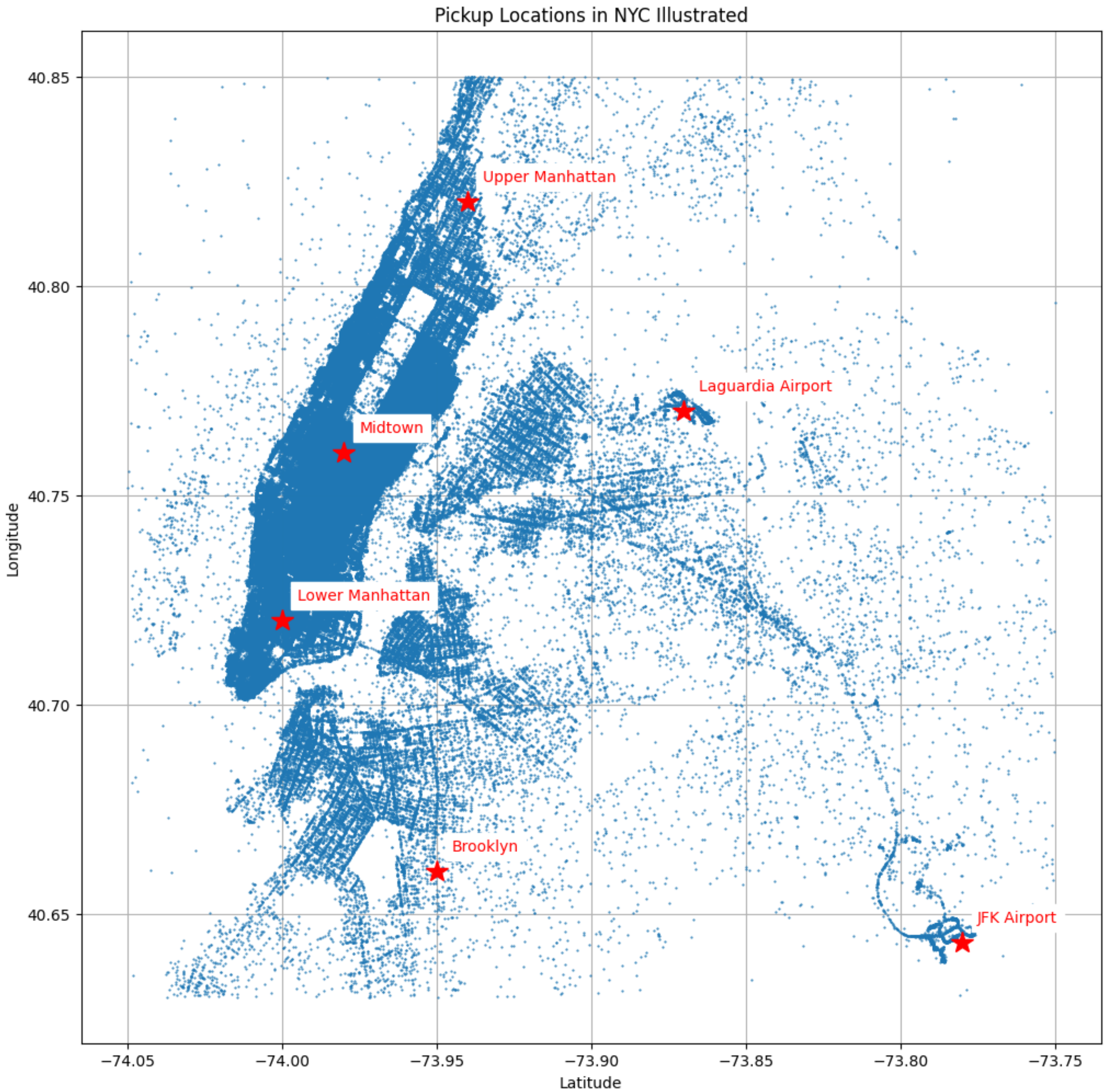
In [15]:
```
#let's define a new function that will take our DataFrame as an input, and p

landmarks = {'JFK Airport': (-73.78, 40.643),
             'Laguardia Airport': (-73.87, 40.77),
             'Midtown': (-73.98, 40.76),
             'Lower Manhattan': (-74.00, 40.72),
```

```
             'Upper Manhattan': (-73.94, 40.82),
             'Brooklyn': (-73.95, 40.66)}
```

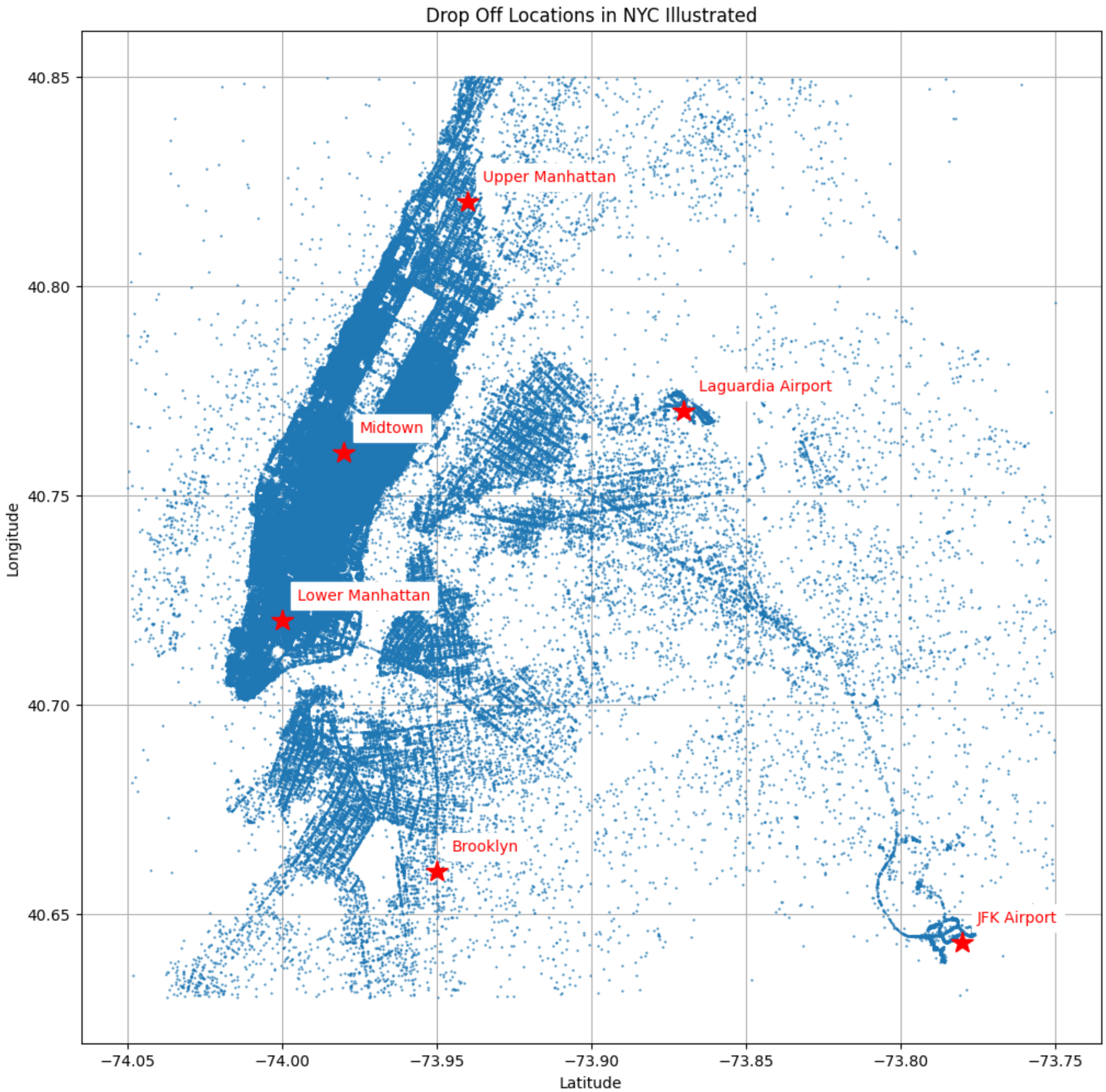
In [16]:
```
import matplotlib.pyplot as plt
%matplotlib inline
def plot_lat_long(df, landmarks, points='Pickup'):
    plt.figure(figsize = (12,12)) # set figure size
    if points == 'pickup':
        plt.plot(list(df.pickup_longitude), list(df.pickup_latitude),
                 '.', markersize=1)
    else:
        plt.plot(list(df.dropoff_longitude), list(df.dropoff_latitude),
                 '.', markersize=1)

    for landmark in landmarks:
        plt.plot(landmarks[landmark][0], landmarks[landmark][1],
                '*', markersize=15, alpha=1, color='r')
        plt.annotate(landmark, (landmarks[landmark][0]+0.005,
                     landmarks[landmark][1]+0.005), color='r',
                     backgroundcolor='w')

    plt.title("{} Locations in NYC Illustrated".format(points))
    plt.grid(None)
    plt.xlabel("Latitude")
    plt.ylabel("Longitude")
    plt.show()
```

In [17]:
```
plot_lat_long(df2, landmarks, points='Pickup')
```


Histogram of Passenger Count
350000
300000
250000
200000
150000
100000
50000
0
0
100
200
300
400
500
Passenger Count

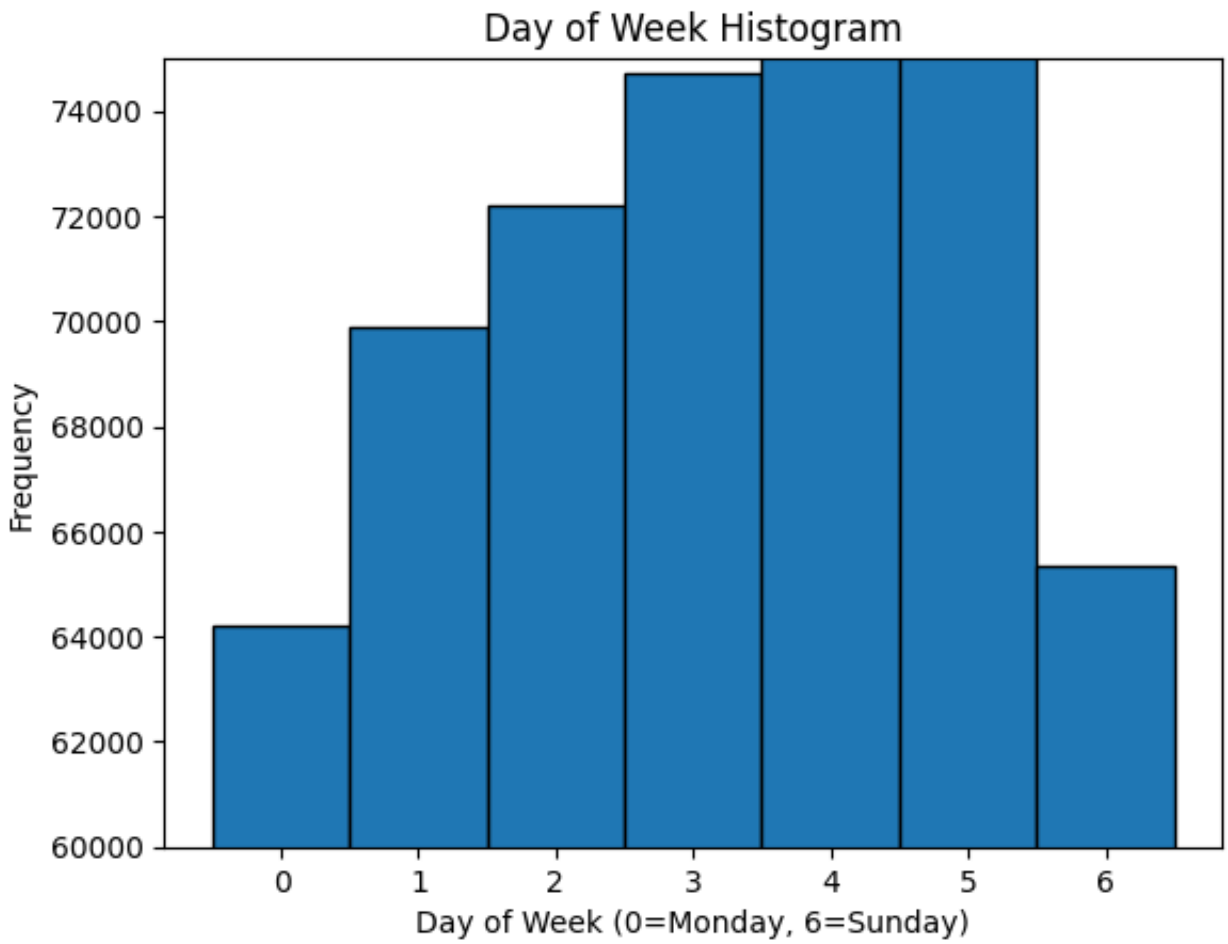



```
In [18]: plot_lat_long(df2, landmarks, points='Drop Off')
```

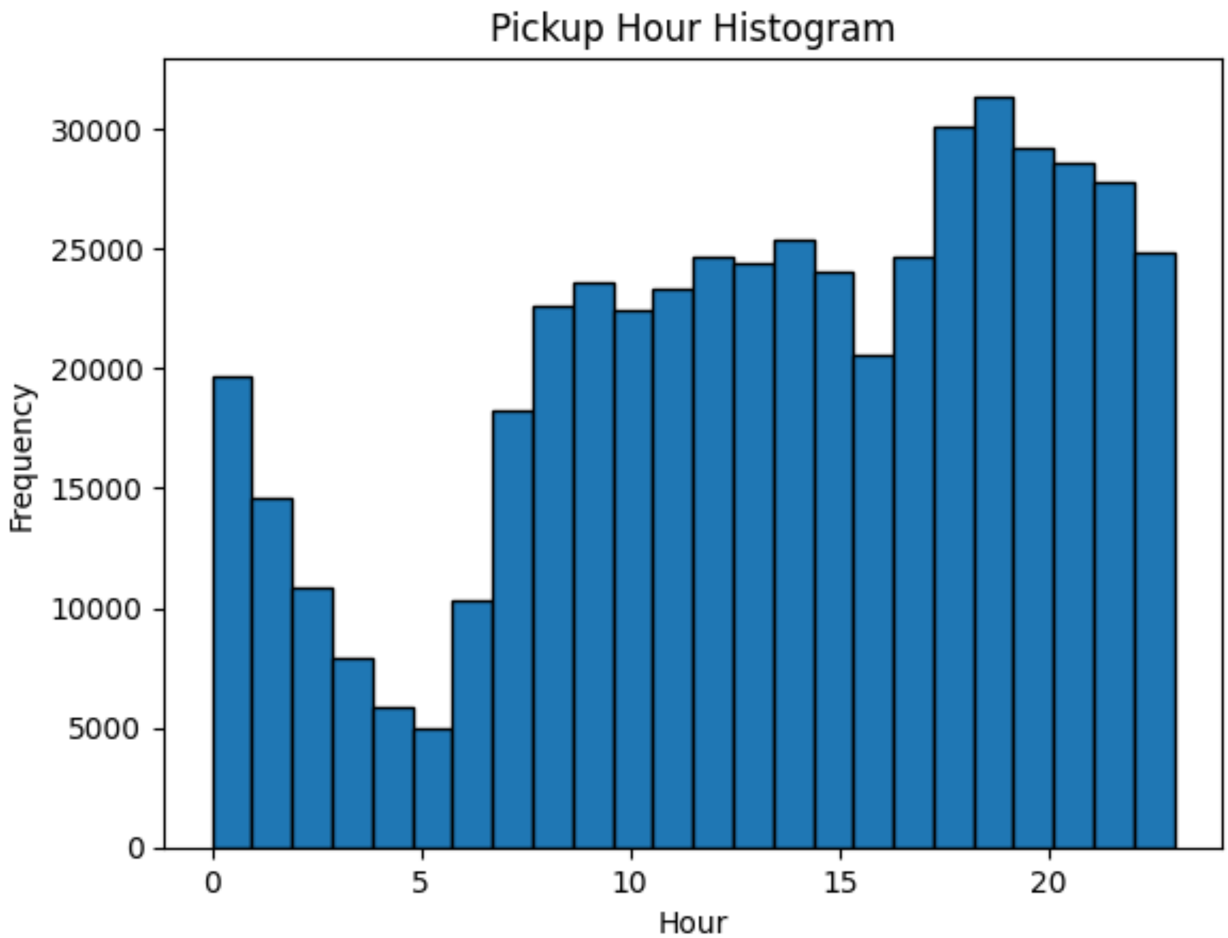


In [19]:
```
df['year'] = df['pickup_datetime'].dt.year
df['month'] = df['pickup_datetime'].dt.month
df['day'] = df['pickup_datetime'].dt.day
df['day_of_week'] = df['pickup_datetime'].dt.dayofweek
df['hour'] = df['pickup_datetime'].dt.hour
```

In [20]:
```
import numpy as np
df['day_of_week'].plot.hist(bins=np.arange(8)-0.5, ec='black',
                            ylim=(60000,75000))
plt.xlabel('Day of Week (0=Monday, 6=Sunday)')
plt.title('Day of Week Histogram')
plt.show()
```

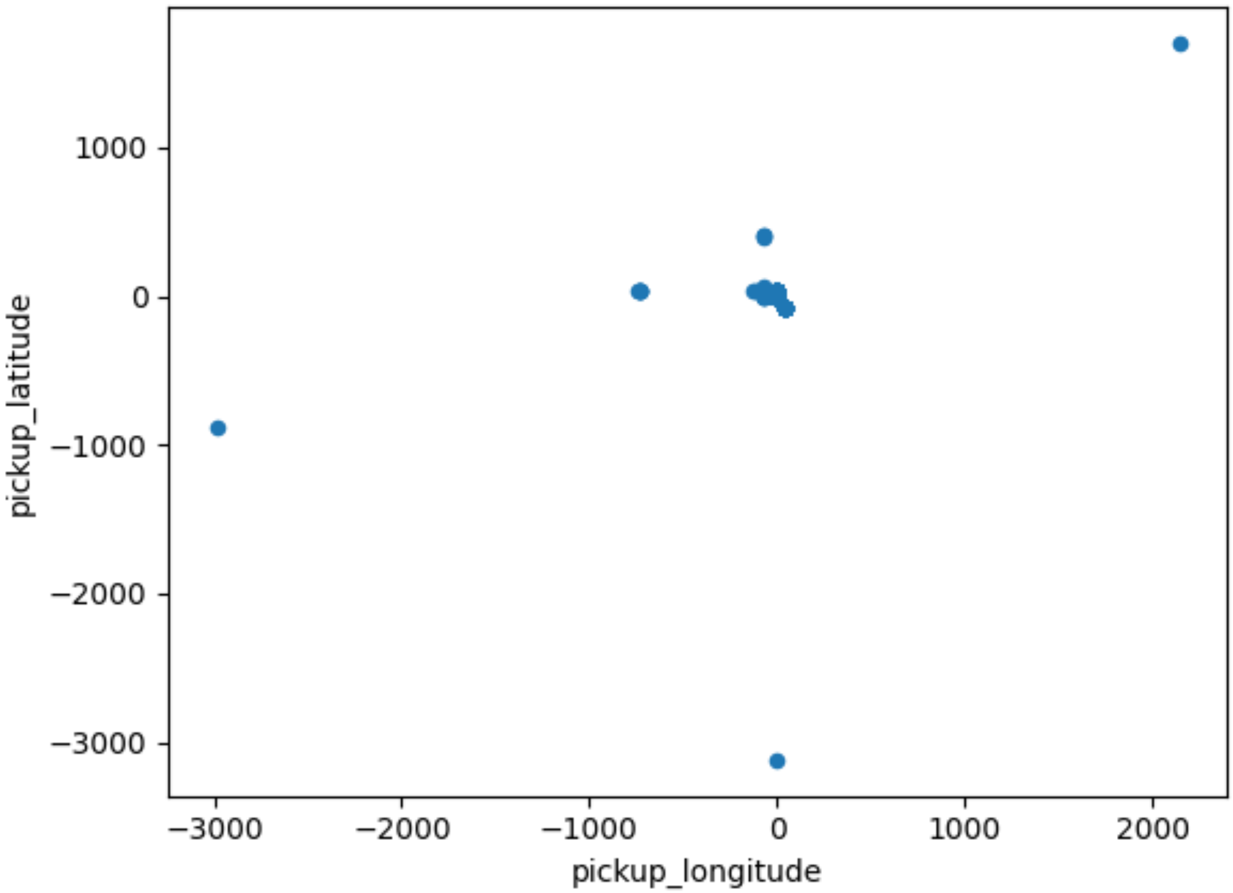


In [21]:
```
df['hour'].plot.hist(bins=24, ec='black')
plt.title('Pickup Hour Histogram')
plt.xlabel('Hour')
plt.show()
```

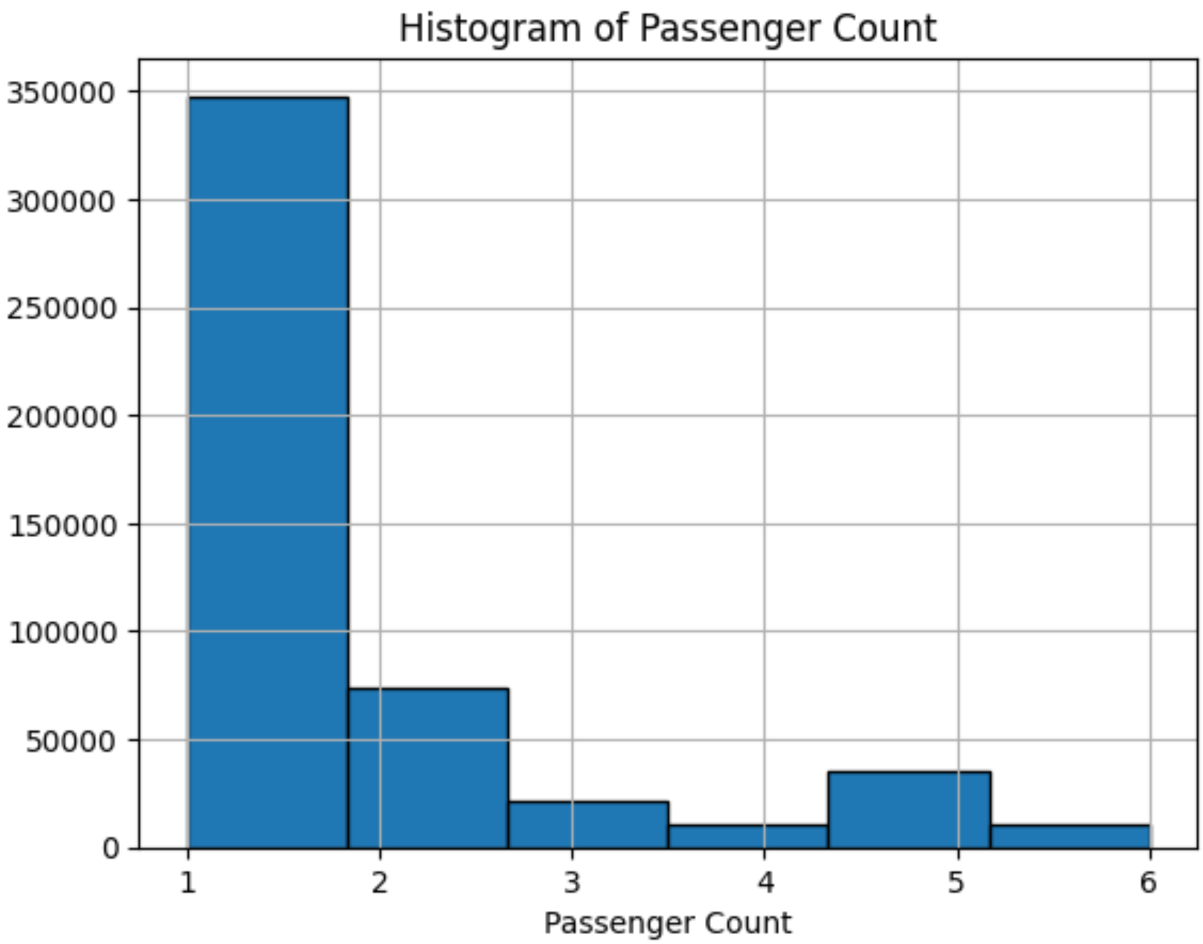


In [22]:

```
df.plot.scatter('pickup_longitude', 'pickup_latitude')
plt.show()
```



In [23]:
```
df['passenger_count'].hist(bins=6, ec='black')
plt.xlabel("Passenger Count")
plt.title("Histogram of Passenger Count")
plt.show()
```



In [24]:
```
df.loc[df['passenger_count']==0, 'passenger_count'] = 1
```

In [26]:
```
## RemovinG Outliers:

# range of longitude for NYC
nyc_min_longitude = -74.05
nyc_max_longitude = -73.75

# range of latitude for NYC
nyc_min_latitude = 40.63
nyc_max_latitude = 40.85

# only consider locations within NYC
for long in ['pickup_longitude', 'dropoff_longitude']:
    df = df[(df[long] > nyc_min_longitude) & (df[long] < nyc_max_longitude)]

for lat in ['pickup_latitude', 'dropoff_latitude']:
    df = df[(df[lat] > nyc_min_latitude) & (df[lat] <  nyc_max_latitude)]
```

In [27]:
```
# going to create functions to create a simplier project
```

```
def preprocess(df):
    # remove missing values in the dataframe
    def remove_missing_values(df):
        df = df.dropna()
        return df

    # remove outliers in fare amount
    def remove_fare_amount_outliers(df, lower_bound, upper_bound):
        df = df[(df['fare_amount'] >= lower_bound) &
                (df['fare_amount'] <= upper_bound)]
        return df

    # replace outliers in passenger count with the mode
    def replace_passenger_count_outliers(df):
        mode = df['passenger_count'].mode()
        df.loc[df['passenger_count'] == 0, 'passenger_count'] = mode
        return df
# remove outliers in latitude and longitude
    def remove_lat_long_outliers(df):
        # range of longitude for NYC
        nyc_min_longitude = -74.05
        nyc_max_longitude = -73.75
        # range of latitude for NYC
        nyc_min_latitude = 40.63
        nyc_max_latitude = 40.85
        # only consider locations within New York City
        for long in ['pickup_longitude', 'dropoff_longitude']:
            df = df[(df[long] > nyc_min_longitude) &
                    (df[long] < nyc_max_longitude)]
        for lat in ['pickup_latitude', 'dropoff_latitude']:
            df = df[(df[lat] > nyc_min_latitude) &
                    (df[lat] < nyc_max_latitude)]
        return df

    df = remove_missing_values(df)
    df = remove_fare_amount_outliers(df, lower_bound = 0,
                                     upper_bound = 100)
    df = replace_passenger_count_outliers(df)
    df = remove_lat_long_outliers(df)
    return df
```

In [ ]:

Feature Engineering:

using one's domain knowledge of the problem to create new features for the machine learning algorithm

In [28]:
```python
print(df.head()['pickup_datetime'])
```

```
0   2009-06-15 17:26:21+00:00
1   2010-01-05 16:52:16+00:00
2   2011-08-18 00:35:00+00:00
3   2012-04-21 04:30:42+00:00
4   2010-03-09 07:51:00+00:00
Name: pickup_datetime, dtype: datetime64[ns, UTC]
```

```
0   2009-06-15 17:26:21+00:00
1   2010-01-05 16:52:16+00:00
2   2011-08-18 00:35:00+00:00
3   2012-04-21 04:30:42+00:00
4   2010-03-09 07:51:00+00:00
Name: pickup_datetime, dtype: datetime64[ns, UTC]
```


In [30]:
```python
df['year'] = df['pickup_datetime'].dt.year
df['month'] = df['pickup_datetime'].dt.month
df['day'] = df['pickup_datetime'].dt.day
df['day_of_week'] = df['pickup_datetime'].dt.dayofweek
df['hour'] = df['pickup_datetime'].dt.hour
```

In [31]:
```python
print(df.loc[:5,['pickup_datetime', 'year', 'month',
                 'day', 'day_of_week', 'hour']])
```

```
            pickup_datetime  year  month  day  day_of_week  hour
0 2009-06-15 17:26:21+00:00  2009      6   15            0    17
1 2010-01-05 16:52:16+00:00  2010      1    5            1    16
2 2011-08-18 00:35:00+00:00  2011      8   18            3     0
3 2012-04-21 04:30:42+00:00  2012      4   21            5     4
4 2010-03-09 07:51:00+00:00  2010      3    9            1     7
5 2011-01-06 09:50:45+00:00  2011      1    6            3     9
```

```
            pickup_datetime  year  month  day  day_of_week  hour
0 2009-06-15 17:26:21+00:00  2009      6   15            0    17
1 2010-01-05 16:52:16+00:00  2010      1    5            1    16
2 2011-08-18 00:35:00+00:00  2011      8   18            3     0
3 2012-04-21 04:30:42+00:00  2012      4   21            5     4
4 2010-03-09 07:51:00+00:00  2010      3    9            1     7
5 2011-01-06 09:50:45+00:00  2011      1    6            3     9
```


In [32]:
```python
df = df.drop(['pickup_datetime'], axis=1)
```

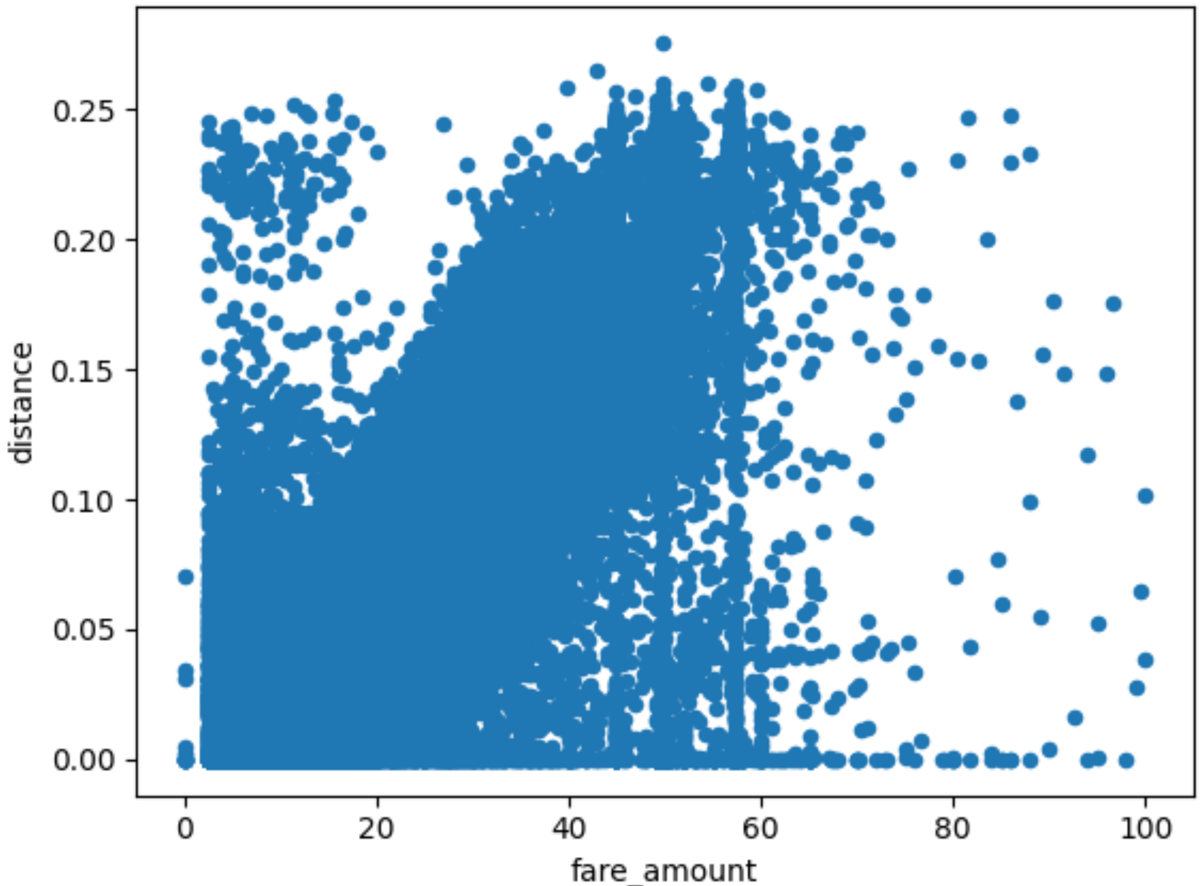
In [33]:
```python
def euc_distance(lat1, long1, lat2, long2):
    return(((lat1-lat2)**2 + (long1-long2)**2)**0.5)
```

In [34]:
```python
df['distance'] = euc_distance(df['pickup_latitude'],
                              df['pickup_longitude'],
                              df['dropoff_latitude'],
                              df['dropoff_longitude'])
```

Hypothesis:

trip fare is closely correlated to the distance traveled

In [35]:
```
df.plot.scatter('fare_amount', 'distance')
plt.show()
```



In [36]:
```
airports = {'JFK_Airport': (-73.78,40.643),
            'Laguardia_Airport': (-73.87, 40.77),
            'Newark_Airport' : (-74.18, 40.69)}

for airport in airports:
    df['pickup_dist_' + airport] = euc_distance(df['pickup_latitude'],
                                                df['pickup_longitude'],
                                                airports[airport][1],
                                                airports[airport][0])
    df['dropoff_dist_' + airport] = euc_distance(df['dropoff_latitude'],
                                                 df['dropoff_longitude'],
                                                 airports[airport][1],
                                                 airports[airport][0])
```

In [37]:
```
print(df[['key', 'pickup_longitude', 'pickup_latitude',
```

```
          'dropoff_longitude', 'dropoff_latitude',
          'pickup_dist_JFK_Airport',
          'dropoff_dist_JFK_Airport']].head())
```

```
                             key  pickup_longitude  pickup_latitude  dropoff
_longitude  dropoff_latitude  \
0    2009-06-15 17:26:21.0000001        -73.844311        40.721319
-73.841610         40.712278
1    2010-01-05 16:52:16.0000002        -74.016048        40.711303
-73.979268         40.782004
2   2011-08-18 00:35:00.00000049        -73.982738        40.761270
-73.991242         40.750562
3    2012-04-21 04:30:42.0000001        -73.987130        40.733143
-73.991567         40.758092
4  2010-03-09 07:51:00.000000135        -73.968095        40.768008
-73.956655         40.783762

   pickup_dist_JFK_Airport  dropoff_dist_JFK_Airport
0                 0.101340                  0.092710
1                 0.245731                  0.242961
2                 0.234714                  0.237050
3                 0.225895                  0.240846
4                 0.225847                  0.225878
                             key  pickup_longitude  pickup_latitude  dropoff
_longitude  dropoff_latitude  \
0    2009-06-15 17:26:21.0000001        -73.844311        40.721319
-73.841610         40.712278
1    2010-01-05 16:52:16.0000002        -74.016048        40.711303
-73.979268         40.782004
2   2011-08-18 00:35:00.00000049        -73.982738        40.761270
-73.991242         40.750562
3    2012-04-21 04:30:42.0000001        -73.987130        40.733143
-73.991567         40.758092
4  2010-03-09 07:51:00.000000135        -73.968095        40.768008
-73.956655         40.783762

   pickup_dist_JFK_Airport  dropoff_dist_JFK_Airport
0                 0.101340                  0.092710
1                 0.245731                  0.242961
2                 0.234714                  0.237050
3                 0.225895                  0.240846
4                 0.225847                  0.225878
```

In [38]:

```
df = df.drop(['key'], axis=1)

## drop 'key' because its irrelevant
```

In [39]:

```
def feature_engineer(df):
    # create new columns for year, month, day, day of week and hour
    def create_time_features(df):
```

```
        df['year'] = df['pickup_datetime'].dt.year
        df['month'] = df['pickup_datetime'].dt.month
        df['day'] = df['pickup_datetime'].dt.day
        df['day_of_week'] = df['pickup_datetime'].dt.dayofweek
        df['hour'] = df['pickup_datetime'].dt.hour
        df = df.drop(['pickup_datetime'], axis=1)
        return df

    # function to calculate euclidean distance
    def euc_distance(lat1, long1, lat2, long2):
        return(((lat1-lat2)**2 + (long1-long2)**2)**0.5)

    # create new column for the distance travelled
    def create_pickup_dropoff_dist_features(df):
        df['travel_distance'] = euc_distance(df['pickup_latitude'],
                                             df['pickup_longitude'],
                                             df['dropoff_latitude'],
                                             df['dropoff_longitude'])
        return df

    # create new column for the distance away from airports
    def create_airport_dist_features(df):
        airports = {'JFK_Airport': (-73.78,40.643),
                    'Laguardia_Airport': (-73.87, 40.77),
                    'Newark_Airport' : (-74.18, 40.69)}
        for k in airports:
            df['pickup_dist_'+k]=euc_distance(df['pickup_latitude'],
                                              df['pickup_longitude'],
                                              airports[k][1],
                                              airports[k][0])
            df['dropoff_dist_'+k]=euc_distance(df['dropoff_latitude'],
                                               df['dropoff_longitude'],
                                               airports[k][1],
                                               airports[k][0])
        return df

    df = create_time_features(df)
    df = create_pickup_dropoff_dist_features(df)
    df = create_airport_dist_features(df)
    df = df.drop(['key'], axis=1)
    return df
```

Feature Scaling

Making sure each feature has uniform scale

In [40]:
```
df_prescaled = df.copy()
```

In [41]:
```
df_scaled = df.drop(['fare_amount'], axis=1)
```

In [42]:
```
from sklearn.preprocessing import scale

df_scaled = scale(df_scaled)
```

In [43]:
```
cols = df.columns.tolist()
cols.remove('fare_amount')
df_scaled = pd.DataFrame(df_scaled, columns=cols, index=df.index)
df_scaled = pd.concat([df_scaled, df['fare_amount']], axis=1)
df = df_scaled.copy()
```

Splitting The Data: Keras

In [44]:
```
X = df.loc[:, df.columns != 'fare_amount']
y = df.loc[:, 'fare_amount']
```

In [45]:
```
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2)
```

In [46]:
```
from keras.models import Sequential
from keras.layers import Dense

model = Sequential()
model.add(Dense(128, activation= 'relu', input_dim=X_train.shape[1]))
model.add(Dense(64, activation= 'relu'))
model.add(Dense(32, activation= 'relu'))
model.add(Dense(8, activation= 'relu'))
model.add(Dense(1))
```

In [47]:
```
model.summary()
```

Model: "sequential"
Model: "sequential"

```
┏━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━┳━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━┳━━━━━━━━━━━━━━━━━━┓
┃ Layer (type)
Param # ┃
┡━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━╇━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━╇━━━━━━━━━━━━━━━━━━┩
│ dense (Dense)                          │ (None, 128)                   │
2,304 │
├────────────────────────────────────────┼──────────────────────────────┼──────────────────┤
│ dense_1 (Dense)                        │ (None, 64)                    │
```

```
8,256 │
├─────────────────────────────────────────┼───────────────────────────────┼──────
────────────┤
│ dense_2 (Dense)                         │ (None, 32)                    │
2,080 │
├─────────────────────────────────────────┼───────────────────────────────┼──────
────────────┤
│ dense_3 (Dense)                         │ (None, 8)                     │
264 │
├─────────────────────────────────────────┼───────────────────────────────┼──────
────────────┤
│ dense_4 (Dense)                         │ (None, 1)                     │
9 │
└─────────────────────────────────────────┴───────────────────────────────┴──────
────────────┘
┏━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━┳━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━┳━━━━━━
━━━━━━━━━━━━┓
┃ Layer (type)                            ┃ Output Shape                  ┃
Param # ┃
┡━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━╇━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━╇━━━━━━
━━━━━━━━━━━━┩
│ dense (Dense)                           │ (None, 128)                   │
2,304 │
├─────────────────────────────────────────┼───────────────────────────────┼──────
────────────┤
│ dense_1 (Dense)                         │ (None, 64)                    │
8,256 │
├─────────────────────────────────────────┼───────────────────────────────┼──────
────────────┤
│ dense_2 (Dense)                         │ (None, 32)                    │
2,080 │
├─────────────────────────────────────────┼───────────────────────────────┼──────
────────────┤
│ dense_3 (Dense)                         │ (None, 8)                     │
264 │
├─────────────────────────────────────────┼───────────────────────────────┼──────
────────────┤
│ dense_4 (Dense)                         │ (None, 1)                     │
9 │
└─────────────────────────────────────────┴───────────────────────────────┴──────
────────────┘
 Total params: 12,913 (50.44 KB)
 Total params: 12,913 (50.44 KB)
 Trainable params: 12,913 (50.44 KB)
 Trainable params: 12,913 (50.44 KB)
 Non-trainable params: 0 (0.00 B)
 Non-trainable params: 0 (0.00 B)
┏━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━┳━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━┳━━━━━━
━━━━━━━━━━━━┓
┃ Layer (type)                            ┃ Output Shape                  ┃
```

Param #

dense (Dense)	(None, 128)	2,304
dense_1 (Dense)	(None, 64)	8,256
dense_2 (Dense)	(None, 32)	2,080
dense_3 (Dense)	(None, 8)	264
dense_4 (Dense)	(None, 1)	9

Layer (type)	Output Shape	Param #
dense (Dense)	(None, 128)	2,304
dense_1 (Dense)	(None, 64)	8,256
dense_2 (Dense)	(None, 32)	2,080
dense_3 (Dense)	(None, 8)	264
dense_4 (Dense)	(None, 1)	9

```
Total params: 12,913 (50.44 KB)
```


```
Total params: 12,913 (50.44 KB)
```


```
Trainable params: 12,913 (50.44 KB)
```


```
Trainable params: 12,913 (50.44 KB)
```


```
Non-trainable params: 0 (0.00 B)
```


```
Non-trainable params: 0 (0.00 B)
```


In [48]:
```
model.compile(loss='mse', optimizer='adam', metrics=['mse'])
model.fit(X_train, y_train, epochs=1)
```

```
12086/12086 ━━━━━━━━━━━━━━━━━━━━ 5s 360us/step - loss: 17.0973 - mse: 17.097
3
12086/12086 ━━━━━━━━━━━━━━━━━━━━ 5s 360us/step - loss: 17.0973 - mse: 17.097
3
```

Out[48]: `<keras.src.callbacks.history.History at 0x17b0d6fd0>`

```
  276/12086 ━━━━━━━━━━━━━━━━━━━━ 4s 365us/step - loss: 74.2244 - mse: 74.224
4
  276/12086 ━━━━━━━━━━━━━━━━━━━━ 4s 365us/step - loss: 74.2244 - mse: 74.224
4
  422/12086 ━━━━━━━━━━━━━━━━━━━━ 4s 357us/step - loss: 59.6179 - mse: 59.617
9
  422/12086 ━━━━━━━━━━━━━━━━━━━━ 4s 357us/step - loss: 59.6179 - mse: 59.617
9
  566/12086 ━━━━━━━━━━━━━━━━━━━━ 4s 355us/step - loss: 51.4886 - mse: 51.488
6
  566/12086 ━━━━━━━━━━━━━━━━━━━━ 4s 355us/step - loss: 51.4886 - mse: 51.488
6
  714/12086 ━━━━━━━━━━━━━━━━━━━━ 4s 352us/step - loss: 45.9170 - mse: 45.917
0
  714/12086 ━━━━━━━━━━━━━━━━━━━━ 4s 352us/step - loss: 45.9170 - mse: 45.917
0
  860/12086 ━━━━━━━━━━━━━━━━━━━━ 3s 350us/step - loss: 41.8904 - mse: 41.890
4
  860/12086 ━━━━━━━━━━━━━━━━━━━━ 3s 350us/step - loss: 41.8904 - mse: 41.890
4
```

```
 1008/12086 ━━━━━━━━━━━━━━━━━━━━ 3s 349us/step - loss: 38.8350 - mse: 38.835
0
 1008/12086 ━━━━━━━━━━━━━━━━━━━━ 3s 349us/step - loss: 38.8350 - mse: 38.835
0
 1154/12086 ━━━━━━━━━━━━━━━━━━━━ 3s 348us/step - loss: 36.5074 - mse: 36.507
4
 1154/12086 ━━━━━━━━━━━━━━━━━━━━ 3s 348us/step - loss: 36.5074 - mse: 36.507
4
 1302/12086 ━━━━━━━━━━━━━━━━━━━━ 3s 347us/step - loss: 34.6076 - mse: 34.607
6
 1302/12086 ━━━━━━━━━━━━━━━━━━━━ 3s 347us/step - loss: 34.6076 - mse: 34.607
6
 1449/12086 ━━━━━━━━━━━━━━━━━━━━ 3s 346us/step - loss: 33.0514 - mse: 33.051
4
 1449/12086 ━━━━━━━━━━━━━━━━━━━━ 3s 346us/step - loss: 33.0514 - mse: 33.051
4
 1598/12086 ━━━━━━━━━━━━━━━━━━━━ 3s 345us/step - loss: 31.7110 - mse: 31.711
0
 1598/12086 ━━━━━━━━━━━━━━━━━━━━ 3s 345us/step - loss: 31.7110 - mse: 31.711
0
 1744/12086 ━━━━━━━━━━━━━━━━━━━━ 3s 345us/step - loss: 30.5807 - mse: 30.580
7
 1744/12086 ━━━━━━━━━━━━━━━━━━━━ 3s 345us/step - loss: 30.5807 - mse: 30.580
7
 1893/12086 ━━━━━━━━━━━━━━━━━━━━ 3s 345us/step - loss: 29.5860 - mse: 29.586
0
 1893/12086 ━━━━━━━━━━━━━━━━━━━━ 3s 345us/step - loss: 29.5860 - mse: 29.586
0
 2040/12086 ━━━━━━━━━━━━━━━━━━━━ 3s 344us/step - loss: 28.7281 - mse: 28.728
1
 2040/12086 ━━━━━━━━━━━━━━━━━━━━ 3s 344us/step - loss: 28.7281 - mse: 28.728
1
```

```
2189/12086 ━━━━━━━━━━━━━━━━━━━━ 3s 344us/step - loss: 27.9623 - mse: 27.9623
2189/12086 ━━━━━━━━━━━━━━━━━━━━ 3s 344us/step - loss: 27.9623 - mse: 27.9623
2337/12086 ━━━━━━━━━━━━━━━━━━━━ 3s 344us/step - loss: 27.2804 - mse: 27.2804
2337/12086 ━━━━━━━━━━━━━━━━━━━━ 3s 344us/step - loss: 27.2804 - mse: 27.2804
2431/12086 ━━━━━━━━━━━━━━━━━━━━ 3s 354us/step - loss: 26.8864 - mse: 26.8864
2431/12086 ━━━━━━━━━━━━━━━━━━━━ 3s 354us/step - loss: 26.8864 - mse: 26.8864
2561/12086 ━━━━━━━━━━━━━━━━━━━━ 3s 356us/step - loss: 26.3833 - mse: 26.3833
2561/12086 ━━━━━━━━━━━━━━━━━━━━ 3s 356us/step - loss: 26.3833 - mse: 26.3833
2707/12086 ━━━━━━━━━━━━━━━━━━━━ 3s 355us/step - loss: 25.8637 - mse: 25.8637
2707/12086 ━━━━━━━━━━━━━━━━━━━━ 3s 355us/step - loss: 25.8637 - mse: 25.8637
2855/12086 ━━━━━━━━━━━━━━━━━━━━ 3s 354us/step - loss: 25.3792 - mse: 25.3792
2855/12086 ━━━━━━━━━━━━━━━━━━━━ 3s 354us/step - loss: 25.3792 - mse: 25.3792
3005/12086 ━━━━━━━━━━━━━━━━━━━━ 3s 354us/step - loss: 24.9261 - mse: 24.9261
3005/12086 ━━━━━━━━━━━━━━━━━━━━ 3s 354us/step - loss: 24.9261 - mse: 24.9261
3156/12086 ━━━━━━━━━━━━━━━━━━━━ 3s 353us/step - loss: 24.5048 - mse: 24.5048
3156/12086 ━━━━━━━━━━━━━━━━━━━━ 3s 353us/step - loss: 24.5048 - mse: 24.5048
```

```
□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
 3305/12086 ━━━━━━━━━━━━━━━━━━━━ 3s 352us/step - loss: 24.1225 - mse: 24.122
5□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
□
 3305/12086 ━━━━━━━━━━━━━━━━━━━━ 3s 352us/step - loss: 24.1225 - mse: 24.122
5
□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
 3454/12086 ━━━━━━━━━━━━━━━━━━━━ 3s 351us/step - loss: 23.7704 - mse: 23.770
4□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
□
 3454/12086 ━━━━━━━━━━━━━━━━━━━━ 3s 351us/step - loss: 23.7704 - mse: 23.770
4
□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
 3604/12086 ━━━━━━━━━━━━━━━━━━━━ 2s 351us/step - loss: 23.4406 - mse: 23.440
6□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
□
 3604/12086 ━━━━━━━━━━━━━━━━━━━━ 2s 351us/step - loss: 23.4406 - mse: 23.440
6
□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
 3752/12086 ━━━━━━━━━━━━━━━━━━━━ 2s 350us/step - loss: 23.1371 - mse: 23.137
1□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
□
 3752/12086 ━━━━━━━━━━━━━━━━━━━━ 2s 350us/step - loss: 23.1371 - mse: 23.137
1
□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
 3901/12086 ━━━━━━━━━━━━━━━━━━━━ 2s 350us/step - loss: 22.8475 - mse: 22.847
5□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
□
 3901/12086 ━━━━━━━━━━━━━━━━━━━━ 2s 350us/step - loss: 22.8475 - mse: 22.847
5
□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
 4047/12086 ━━━━━━━━━━━━━━━━━━━━ 2s 349us/step - loss: 22.5794 - mse: 22.579
4□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
□
 4047/12086 ━━━━━━━━━━━━━━━━━━━━ 2s 349us/step - loss: 22.5794 - mse: 22.579
4
□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
 4196/12086 ━━━━━━━━━━━━━━━━━━━━ 2s 349us/step - loss: 22.3190 - mse: 22.319
0□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
□
 4196/12086 ━━━━━━━━━━━━━━━━━━━━ 2s 349us/step - loss: 22.3190 - mse: 22.319
0
□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
 4342/12086 ━━━━━━━━━━━━━━━━━━━━ 2s 349us/step - loss: 22.0771 - mse: 22.077
1□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
□
 4342/12086 ━━━━━━━━━━━━━━━━━━━━ 2s 349us/step - loss: 22.0771 - mse: 22.077
1
```

```
                                                                            
 4489/12086 ━━━━━━━━━━━━━━━━━━━━ 2s 348us/step - loss: 21.8444 - mse: 21.844
4                                                                            

 4489/12086 ━━━━━━━━━━━━━━━━━━━━ 2s 348us/step - loss: 21.8444 - mse: 21.844
4
                                                                            
 4637/12086 ━━━━━━━━━━━━━━━━━━━━ 2s 348us/step - loss: 21.6222 - mse: 21.622
2                                                                            

 4637/12086 ━━━━━━━━━━━━━━━━━━━━ 2s 348us/step - loss: 21.6222 - mse: 21.622
2
                                                                            
 4784/12086 ━━━━━━━━━━━━━━━━━━━━ 2s 348us/step - loss: 21.4146 - mse: 21.414
6                                                                            

 4784/12086 ━━━━━━━━━━━━━━━━━━━━ 2s 348us/step - loss: 21.4146 - mse: 21.414
6
                                                                            
 4932/12086 ━━━━━━━━━━━━━━━━━━━━ 2s 348us/step - loss: 21.2161 - mse: 21.216
1                                                                            

 4932/12086 ━━━━━━━━━━━━━━━━━━━━ 2s 348us/step - loss: 21.2161 - mse: 21.216
1
                                                                            
 5083/12086 ━━━━━━━━━━━━━━━━━━━━ 2s 347us/step - loss: 21.0234 - mse: 21.023
4                                                                            

 5083/12086 ━━━━━━━━━━━━━━━━━━━━ 2s 347us/step - loss: 21.0234 - mse: 21.023
4
                                                                            
 5230/12086 ━━━━━━━━━━━━━━━━━━━━ 2s 347us/step - loss: 20.8473 - mse: 20.847
3                                                                            

 5230/12086 ━━━━━━━━━━━━━━━━━━━━ 2s 347us/step - loss: 20.8473 - mse: 20.847
3
                                                                            
 5377/12086 ━━━━━━━━━━━━━━━━━━━━ 2s 347us/step - loss: 20.6814 - mse: 20.681
4                                                                            

 5377/12086 ━━━━━━━━━━━━━━━━━━━━ 2s 347us/step - loss: 20.6814 - mse: 20.681
4
                                                                            
 5526/12086 ━━━━━━━━━━━━━━━━━━━━ 2s 347us/step - loss: 20.5214 - mse: 20.521
4                                                                            

 5526/12086 ━━━━━━━━━━━━━━━━━━━━ 2s 347us/step - loss: 20.5214 - mse: 20.521
4
```

```
□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
 5592/12086 ━━━━━━━━━━━━━━━━━━━━ 2s 352us/step - loss: 20.4525 - mse: 20.452
5□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
□
 5592/12086 ━━━━━━━━━━━━━━━━━━━━ 2s 352us/step - loss: 20.4525 - mse: 20.452
5
□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
 5729/12086 ━━━━━━━━━━━━━━━━━━━━ 2s 352us/step - loss: 20.3130 - mse: 20.313
0□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
□
 5729/12086 ━━━━━━━━━━━━━━━━━━━━ 2s 352us/step - loss: 20.3130 - mse: 20.313
0
□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
 5877/12086 ━━━━━━━━━━━━━━━━━━━━ 2s 352us/step - loss: 20.1681 - mse: 20.168
1□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
□
 5877/12086 ━━━━━━━━━━━━━━━━━━━━ 2s 352us/step - loss: 20.1681 - mse: 20.168
1
□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
 6024/12086 ━━━━━━━━━━━━━━━━━━━━ 2s 351us/step - loss: 20.0320 - mse: 20.032
0□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
□
 6024/12086 ━━━━━━━━━━━━━━━━━━━━ 2s 351us/step - loss: 20.0320 - mse: 20.032
0
□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
 6169/12086 ━━━━━━━━━━━━━━━━━━━━ 2s 351us/step - loss: 19.9045 - mse: 19.904
5□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
□
 6169/12086 ━━━━━━━━━━━━━━━━━━━━ 2s 351us/step - loss: 19.9045 - mse: 19.904
5
□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
 6304/12086 ━━━━━━━━━━━━━━━━━━━━ 2s 352us/step - loss: 19.7899 - mse: 19.789
9□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
□
 6304/12086 ━━━━━━━━━━━━━━━━━━━━ 2s 352us/step - loss: 19.7899 - mse: 19.789
9
□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
 6441/12086 ━━━━━━━━━━━━━━━━━━━━ 1s 352us/step - loss: 19.6776 - mse: 19.677
6□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
□
 6441/12086 ━━━━━━━━━━━━━━━━━━━━ 1s 352us/step - loss: 19.6776 - mse: 19.677
6
□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
 6576/12086 ━━━━━━━━━━━━━━━━━━━━ 1s 352us/step - loss: 19.5714 - mse: 19.571
4□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
□
 6576/12086 ━━━━━━━━━━━━━━━━━━━━ 1s 352us/step - loss: 19.5714 - mse: 19.571
4
```

```
▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯
 6715/12086 ━━━━━━━━━━━━━━━━━━━━ 1s 353us/step - loss: 19.4663 - mse: 19.466
3▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯
▯
 6715/12086 ━━━━━━━━━━━━━━━━━━━━ 1s 353us/step - loss: 19.4663 - mse: 19.466
3
▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯
 6857/12086 ━━━━━━━━━━━━━━━━━━━━ 1s 353us/step - loss: 19.3627 - mse: 19.362
7▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯
▯
 6857/12086 ━━━━━━━━━━━━━━━━━━━━ 1s 353us/step - loss: 19.3627 - mse: 19.362
7
▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯
 7001/12086 ━━━━━━━━━━━━━━━━━━━━ 1s 353us/step - loss: 19.2611 - mse: 19.261
1▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯
▯
 7001/12086 ━━━━━━━━━━━━━━━━━━━━ 1s 353us/step - loss: 19.2611 - mse: 19.261
1
▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯
 7138/12086 ━━━━━━━━━━━━━━━━━━━━ 1s 353us/step - loss: 19.1688 - mse: 19.168
8▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯
▯
 7138/12086 ━━━━━━━━━━━━━━━━━━━━ 1s 353us/step - loss: 19.1688 - mse: 19.168
8
▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯
 7279/12086 ━━━━━━━━━━━━━━━━━━━━ 1s 353us/step - loss: 19.0776 - mse: 19.077
6▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯
▯
 7279/12086 ━━━━━━━━━━━━━━━━━━━━ 1s 353us/step - loss: 19.0776 - mse: 19.077
6
▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯
 7422/12086 ━━━━━━━━━━━━━━━━━━━━ 1s 353us/step - loss: 18.9890 - mse: 18.989
0▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯
▯
 7422/12086 ━━━━━━━━━━━━━━━━━━━━ 1s 353us/step - loss: 18.9890 - mse: 18.989
0
▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯
 7561/12086 ━━━━━━━━━━━━━━━━━━━━ 1s 353us/step - loss: 18.9053 - mse: 18.905
3▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯
▯
 7561/12086 ━━━━━━━━━━━━━━━━━━━━ 1s 353us/step - loss: 18.9053 - mse: 18.905
3
▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯
 7698/12086 ━━━━━━━━━━━━━━━━━━━━ 1s 353us/step - loss: 18.8246 - mse: 18.824
6▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯
▯
 7698/12086 ━━━━━━━━━━━━━━━━━━━━ 1s 353us/step - loss: 18.8246 - mse: 18.824
6
```

```
 7839/12086 ━━━━━━━━━━━━━━━━━━━━ 1s 353us/step - loss: 18.7434 - mse: 18.743
4
 7839/12086 ━━━━━━━━━━━━━━━━━━━━ 1s 353us/step - loss: 18.7434 - mse: 18.743
4
 7936/12086 ━━━━━━━━━━━━━━━━━━━━ 1s 356us/step - loss: 18.6892 - mse: 18.689
2
 7936/12086 ━━━━━━━━━━━━━━━━━━━━ 1s 356us/step - loss: 18.6892 - mse: 18.689
2
 8065/12086 ━━━━━━━━━━━━━━━━━━━━ 1s 357us/step - loss: 18.6192 - mse: 18.619
2
 8065/12086 ━━━━━━━━━━━━━━━━━━━━ 1s 357us/step - loss: 18.6192 - mse: 18.619
2
 8211/12086 ━━━━━━━━━━━━━━━━━━━━ 1s 357us/step - loss: 18.5428 - mse: 18.542
8
 8211/12086 ━━━━━━━━━━━━━━━━━━━━ 1s 357us/step - loss: 18.5428 - mse: 18.542
8
 8354/12086 ━━━━━━━━━━━━━━━━━━━━ 1s 357us/step - loss: 18.4710 - mse: 18.471
0
 8354/12086 ━━━━━━━━━━━━━━━━━━━━ 1s 357us/step - loss: 18.4710 - mse: 18.471
0
 8500/12086 ━━━━━━━━━━━━━━━━━━━━ 1s 356us/step - loss: 18.4002 - mse: 18.400
2
 8500/12086 ━━━━━━━━━━━━━━━━━━━━ 1s 356us/step - loss: 18.4002 - mse: 18.400
2
 8642/12086 ━━━━━━━━━━━━━━━━━━━━ 1s 356us/step - loss: 18.3327 - mse: 18.332
7
 8642/12086 ━━━━━━━━━━━━━━━━━━━━ 1s 356us/step - loss: 18.3327 - mse: 18.332
7
 8785/12086 ━━━━━━━━━━━━━━━━━━━━ 1s 356us/step - loss: 18.2667 - mse: 18.266
7
 8785/12086 ━━━━━━━━━━━━━━━━━━━━ 1s 356us/step - loss: 18.2667 - mse: 18.266
7
```

```
 8928/12086 ━━━━━━━━━━━━━━━━━━━━ 1s 356us/step - loss: 18.2026 - mse: 18.2026
 8928/12086 ━━━━━━━━━━━━━━━━━━━━ 1s 356us/step - loss: 18.2026 - mse: 18.2026
 9045/12086 ━━━━━━━━━━━━━━━━━━━━ 1s 357us/step - loss: 18.1517 - mse: 18.1517
 9045/12086 ━━━━━━━━━━━━━━━━━━━━ 1s 357us/step - loss: 18.1517 - mse: 18.1517
 9187/12086 ━━━━━━━━━━━━━━━━━━━━ 1s 357us/step - loss: 18.0911 - mse: 18.0911
 9187/12086 ━━━━━━━━━━━━━━━━━━━━ 1s 357us/step - loss: 18.0911 - mse: 18.0911
 9330/12086 ━━━━━━━━━━━━━━━━━━━━ 0s 357us/step - loss: 18.0311 - mse: 18.0311
 9330/12086 ━━━━━━━━━━━━━━━━━━━━ 0s 357us/step - loss: 18.0311 - mse: 18.0311
 9474/12086 ━━━━━━━━━━━━━━━━━━━━ 0s 357us/step - loss: 17.9720 - mse: 17.9720
 9474/12086 ━━━━━━━━━━━━━━━━━━━━ 0s 357us/step - loss: 17.9720 - mse: 17.9720
 9618/12086 ━━━━━━━━━━━━━━━━━━━━ 0s 357us/step - loss: 17.9148 - mse: 17.9148
 9618/12086 ━━━━━━━━━━━━━━━━━━━━ 0s 357us/step - loss: 17.9148 - mse: 17.9148
 9762/12086 ━━━━━━━━━━━━━━━━━━━━ 0s 357us/step - loss: 17.8584 - mse: 17.8584
 9762/12086 ━━━━━━━━━━━━━━━━━━━━ 0s 357us/step - loss: 17.8584 - mse: 17.8584
 9906/12086 ━━━━━━━━━━━━━━━━━━━━ 0s 356us/step - loss: 17.8035 - mse: 17.8035
 9906/12086 ━━━━━━━━━━━━━━━━━━━━ 0s 356us/step - loss: 17.8035 - mse: 17.8035
```

```
10043/12086 0s 357us/step - loss: 17.7520 - mse: 17.7520
10043/12086 0s 357us/step - loss: 17.7520 - mse: 17.7520
10183/12086 0s 357us/step - loss: 17.7001 - mse: 17.7001
10183/12086 0s 357us/step - loss: 17.7001 - mse: 17.7001
10318/12086 0s 357us/step - loss: 17.6508 - mse: 17.6508
10318/12086 0s 357us/step - loss: 17.6508 - mse: 17.6508
10449/12086 0s 357us/step - loss: 17.6040 - mse: 17.6040
10449/12086 0s 357us/step - loss: 17.6040 - mse: 17.6040
10596/12086 0s 357us/step - loss: 17.5529 - mse: 17.5529
10596/12086 0s 357us/step - loss: 17.5529 - mse: 17.5529
10742/12086 0s 357us/step - loss: 17.5032 - mse: 17.5032
10742/12086 0s 357us/step - loss: 17.5032 - mse: 17.5032
10889/12086 0s 357us/step - loss: 17.4544 - mse: 17.4544
10889/12086 0s 357us/step - loss: 17.4544 - mse: 17.4544
10946/12086 0s 359us/step - loss: 17.4357 - mse: 17.4357
10946/12086 0s 359us/step - loss: 17.4357 - mse: 17.4357
```

```
▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯
11085/12086 ━━━━━━━━━━━━━━━━━━━━ 0s 359us/step - loss: 17.3910 - mse: 17.391
0▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯
▯
11085/12086 ━━━━━━━━━━━━━━━━━━━━ 0s 359us/step - loss: 17.3910 - mse: 17.391
0
▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯
11233/12086 ━━━━━━━━━━━━━━━━━━━━ 0s 359us/step - loss: 17.3449 - mse: 17.344
9▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯
▯
11233/12086 ━━━━━━━━━━━━━━━━━━━━ 0s 359us/step - loss: 17.3449 - mse: 17.344
9
▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯
11380/12086 ━━━━━━━━━━━━━━━━━━━━ 0s 359us/step - loss: 17.3005 - mse: 17.300
5▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯
▯
11380/12086 ━━━━━━━━━━━━━━━━━━━━ 0s 359us/step - loss: 17.3005 - mse: 17.300
5
▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯
11470/12086 ━━━━━━━━━━━━━━━━━━━━ 0s 360us/step - loss: 17.2736 - mse: 17.273
6▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯
▯
11470/12086 ━━━━━━━━━━━━━━━━━━━━ 0s 360us/step - loss: 17.2736 - mse: 17.273
6
▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯
11611/12086 ━━━━━━━━━━━━━━━━━━━━ 0s 360us/step - loss: 17.2321 - mse: 17.232
1▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯
▯
11611/12086 ━━━━━━━━━━━━━━━━━━━━ 0s 360us/step - loss: 17.2321 - mse: 17.232
1
▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯
11758/12086 ━━━━━━━━━━━━━━━━━━━━ 0s 360us/step - loss: 17.1893 - mse: 17.189
3▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯
▯
11758/12086 ━━━━━━━━━━━━━━━━━━━━ 0s 360us/step - loss: 17.1893 - mse: 17.189
3
▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯
11904/12086 ━━━━━━━━━━━━━━━━━━━━ 0s 360us/step - loss: 17.1480 - mse: 17.148
0▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯
▯
11904/12086 ━━━━━━━━━━━━━━━━━━━━ 0s 360us/step - loss: 17.1480 - mse: 17.148
0
▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯
12052/12086 ━━━━━━━━━━━━━━━━━━━━ 0s 360us/step - loss: 17.1069 - mse: 17.106
9▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯
▯
12052/12086 ━━━━━━━━━━━━━━━━━━━━ 0s 360us/step - loss: 17.1069 - mse: 17.106
9
```

```
▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯
12086/12086 ━━━━━━━━━━━━━━━━━━━━ 5s 360us/step - loss: 17.0973 - mse: 17.097
3
▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯
12086/12086 ━━━━━━━━━━━━━━━━━━━━ 5s 360us/step - loss: 17.0973 - mse: 17.097
3
```

Out[48]: <keras.src.callbacks.history.History at 0x17b0d6fd0>

In [49]:

```python
def predict_random(df_prescaled, X_test, model):
    sample = X_test.sample(n=1, random_state=np.random.randint(low=0,
                                                              high=10000))
    idx = sample.index[0]

    actual_fare = df_prescaled.loc[idx,'fare_amount']
    day_names = ['Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday',
                 'Saturday', 'Sunday']
    day_of_week = day_names[df_prescaled.loc[idx,'day_of_week']]
    hour = df_prescaled.loc[idx,'hour']
    predicted_fare = model.predict(sample)[0][0]
    rmse = np.sqrt(np.square(predicted_fare-actual_fare))

    print("Trip Details: {}, {}:00hrs".format(day_of_week, hour))
    print("Actual fare: ${:0.2f}".format(actual_fare))
    print("Predicted fare: ${:0.2f}".format(predicted_fare))
    print("RMSE: ${:0.2f}".format(rmse))
```

In [50]:

```python
predict_random(df_prescaled, X_test, model)
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 102ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 102ms/step
Trip Details: Thursday, 0:00hrs
Actual fare: $4.90
Predicted fare: $5.48
RMSE: $0.58
Trip Details: Thursday, 0:00hrs
Actual fare: $4.90
Predicted fare: $5.48
RMSE: $0.58
▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 102ms/step
▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 102ms/step
```

```
Trip Details: Thursday, 0:00hrs
Actual fare: $4.90
Predicted fare: $5.48
RMSE: $0.58
Trip Details: Thursday, 0:00hrs
Actual fare: $4.90
Predicted fare: $5.48
RMSE: $0.58
```

In [51]:
```
predict_random(df_prescaled, X_test, model)
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 10ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 10ms/step
Trip Details: Monday, 2:00hrs
Actual fare: $7.50
Predicted fare: $6.83
RMSE: $0.67
Trip Details: Monday, 2:00hrs
Actual fare: $7.50
Predicted fare: $6.83
RMSE: $0.67
▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 10ms/step
▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 10ms/step
Trip Details: Monday, 2:00hrs
Actual fare: $7.50
Predicted fare: $6.83
RMSE: $0.67
Trip Details: Monday, 2:00hrs
Actual fare: $7.50
Predicted fare: $6.83
RMSE: $0.67
```

In [52]:
```
predict_random(df_prescaled, X_test, model)
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 10ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 10ms/step
Trip Details: Friday, 14:00hrs
Actual fare: $5.70
Predicted fare: $7.53
RMSE: $1.83
Trip Details: Friday, 14:00hrs
Actual fare: $5.70
Predicted fare: $7.53
RMSE: $1.83
▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 10ms/step
▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 10ms/step
```

```
Trip Details: Friday, 14:00hrs
Actual fare: $5.70
Predicted fare: $7.53
RMSE: $1.83
Trip Details: Friday, 14:00hrs
Actual fare: $5.70
Predicted fare: $7.53
RMSE: $1.83
```

In [53]:
```python
predict_random(df_prescaled, X_test, model)
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 9ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 9ms/step
Trip Details: Thursday, 5:00hrs
Actual fare: $15.50
Predicted fare: $17.92
RMSE: $2.42
Trip Details: Thursday, 5:00hrs
Actual fare: $15.50
Predicted fare: $17.92
RMSE: $2.42
□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 9ms/step
□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 9ms/step
Trip Details: Thursday, 5:00hrs
Actual fare: $15.50
Predicted fare: $17.92
RMSE: $2.42
Trip Details: Thursday, 5:00hrs
Actual fare: $15.50
Predicted fare: $17.92
RMSE: $2.42
```

In [54]:
```python
predict_random(df_prescaled, X_test, model)
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 10ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 10ms/step
Trip Details: Saturday, 14:00hrs
Actual fare: $8.50
Predicted fare: $8.21
RMSE: $0.29
Trip Details: Saturday, 14:00hrs
Actual fare: $8.50
Predicted fare: $8.21
RMSE: $0.29
□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 10ms/step
□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 10ms/step
```

```
Trip Details: Saturday, 14:00hrs
Actual fare: $8.50
Predicted fare: $8.21
RMSE: $0.29
Trip Details: Saturday, 14:00hrs
Actual fare: $8.50
Predicted fare: $8.21
RMSE: $0.29
```

In [55]:
```python
predict_random(df_prescaled, X_test, model)
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 11ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 11ms/step
Trip Details: Tuesday, 21:00hrs
Actual fare: $5.30
Predicted fare: $5.57
RMSE: $0.27
Trip Details: Tuesday, 21:00hrs
Actual fare: $5.30
Predicted fare: $5.57
RMSE: $0.27
□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 11ms/step
□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 11ms/step
Trip Details: Tuesday, 21:00hrs
Actual fare: $5.30
Predicted fare: $5.57
RMSE: $0.27
Trip Details: Tuesday, 21:00hrs
Actual fare: $5.30
Predicted fare: $5.57
RMSE: $0.27
```

In [56]:
```python
from sklearn.metrics import mean_squared_error

train_pred = model.predict(X_train)
train_rmse = np.sqrt(mean_squared_error(y_train, train_pred))

test_pred = model.predict(X_test)
test_rmse = np.sqrt(mean_squared_error(y_test, test_pred))

print("Train RMSE: {:0.2f}".format(train_rmse))
print("Test RMSE: {:0.2f}".format(test_rmse))
```

```
12086/12086 ━━━━━━━━━━━━━━━━━━━━ 3s 271us/step
12086/12086 ━━━━━━━━━━━━━━━━━━━━ 3s 271us/step
3022/3022 ━━━━━━━━━━━━━━━━━━━━ 1s 232us/step
3022/3022 ━━━━━━━━━━━━━━━━━━━━ 1s 232us/step
Train RMSE: 3.47
Test RMSE: 3.56
Train RMSE: 3.47
Test RMSE: 3.56

  215/12086 ━━━━━━━━━━━━━━━━━━━━ 2s 235us/step

  215/12086 ━━━━━━━━━━━━━━━━━━━━ 2s 235us/step

  425/12086 ━━━━━━━━━━━━━━━━━━━━ 2s 237us/step

  425/12086 ━━━━━━━━━━━━━━━━━━━━ 2s 237us/step

  633/12086 ━━━━━━━━━━━━━━━━━━━━ 2s 238us/step

  633/12086 ━━━━━━━━━━━━━━━━━━━━ 2s 238us/step

  847/12086 ━━━━━━━━━━━━━━━━━━━━ 2s 237us/step

  847/12086 ━━━━━━━━━━━━━━━━━━━━ 2s 237us/step

 1056/12086 ━━━━━━━━━━━━━━━━━━━━ 2s 238us/step

 1056/12086 ━━━━━━━━━━━━━━━━━━━━ 2s 238us/step

 1260/12086 ━━━━━━━━━━━━━━━━━━━━ 2s 239us/step

 1260/12086 ━━━━━━━━━━━━━━━━━━━━ 2s 239us/step

 1470/12086 ━━━━━━━━━━━━━━━━━━━━ 2s 239us/step

 1470/12086 ━━━━━━━━━━━━━━━━━━━━ 2s 239us/step

 1683/12086 ━━━━━━━━━━━━━━━━━━━━ 2s 239us/step

 1683/12086 ━━━━━━━━━━━━━━━━━━━━ 2s 239us/step

 1893/12086 ━━━━━━━━━━━━━━━━━━━━ 2s 239us/step

 1893/12086 ━━━━━━━━━━━━━━━━━━━━ 2s 239us/step

 2106/12086 ━━━━━━━━━━━━━━━━━━━━ 2s 238us/step

 2106/12086 ━━━━━━━━━━━━━━━━━━━━ 2s 238us/step
```

```
2318/12086 ━━━━━━━━━━━━━━━━━━━━ 2s 238us/step
2318/12086 ━━━━━━━━━━━━━━━━━━━━ 2s 238us/step
2533/12086 ━━━━━━━━━━━━━━━━━━━━ 2s 238us/step
2533/12086 ━━━━━━━━━━━━━━━━━━━━ 2s 238us/step
2746/12086 ━━━━━━━━━━━━━━━━━━━━ 2s 237us/step
2746/12086 ━━━━━━━━━━━━━━━━━━━━ 2s 237us/step
2955/12086 ━━━━━━━━━━━━━━━━━━━━ 2s 238us/step
2955/12086 ━━━━━━━━━━━━━━━━━━━━ 2s 238us/step
3166/12086 ━━━━━━━━━━━━━━━━━━━━ 2s 238us/step
3166/12086 ━━━━━━━━━━━━━━━━━━━━ 2s 238us/step
3311/12086 ━━━━━━━━━━━━━━━━━━━━ 3s 347us/step
3311/12086 ━━━━━━━━━━━━━━━━━━━━ 3s 347us/step
3528/12086 ━━━━━━━━━━━━━━━━━━━━ 2s 340us/step
3528/12086 ━━━━━━━━━━━━━━━━━━━━ 2s 340us/step
3744/12086 ━━━━━━━━━━━━━━━━━━━━ 2s 333us/step
3744/12086 ━━━━━━━━━━━━━━━━━━━━ 2s 333us/step
3969/12086 ━━━━━━━━━━━━━━━━━━━━ 2s 327us/step
3969/12086 ━━━━━━━━━━━━━━━━━━━━ 2s 327us/step
4052/12086 ━━━━━━━━━━━━━━━━━━━━ 2s 334us/step
4052/12086 ━━━━━━━━━━━━━━━━━━━━ 2s 334us/step
4226/12086 ━━━━━━━━━━━━━━━━━━━━ 2s 332us/step
4226/12086 ━━━━━━━━━━━━━━━━━━━━ 2s 332us/step
4441/12086 ━━━━━━━━━━━━━━━━━━━━ 2s 327us/step
4441/12086 ━━━━━━━━━━━━━━━━━━━━ 2s 327us/step
```

```
 4668/12086 ━━━━━━━━━━━━━━━━━━━━ 2s 322us/step
 4668/12086 ━━━━━━━━━━━━━━━━━━━━ 2s 322us/step
 4900/12086 ━━━━━━━━━━━━━━━━━━━━ 2s 317us/step
 4900/12086 ━━━━━━━━━━━━━━━━━━━━ 2s 317us/step
 5087/12086 ━━━━━━━━━━━━━━━━━━━━ 2s 315us/step
 5087/12086 ━━━━━━━━━━━━━━━━━━━━ 2s 315us/step
 5295/12086 ━━━━━━━━━━━━━━━━━━━━ 2s 312us/step
 5295/12086 ━━━━━━━━━━━━━━━━━━━━ 2s 312us/step
 5507/12086 ━━━━━━━━━━━━━━━━━━━━ 2s 309us/step
 5507/12086 ━━━━━━━━━━━━━━━━━━━━ 2s 309us/step
 5721/12086 ━━━━━━━━━━━━━━━━━━━━ 1s 306us/step
 5721/12086 ━━━━━━━━━━━━━━━━━━━━ 1s 306us/step
 5936/12086 ━━━━━━━━━━━━━━━━━━━━ 1s 304us/step
 5936/12086 ━━━━━━━━━━━━━━━━━━━━ 1s 304us/step
 6150/12086 ━━━━━━━━━━━━━━━━━━━━ 1s 301us/step
 6150/12086 ━━━━━━━━━━━━━━━━━━━━ 1s 301us/step
 6366/12086 ━━━━━━━━━━━━━━━━━━━━ 1s 299us/step
 6366/12086 ━━━━━━━━━━━━━━━━━━━━ 1s 299us/step
 6578/12086 ━━━━━━━━━━━━━━━━━━━━ 1s 297us/step
 6578/12086 ━━━━━━━━━━━━━━━━━━━━ 1s 297us/step
 6793/12086 ━━━━━━━━━━━━━━━━━━━━ 1s 295us/step
 6793/12086 ━━━━━━━━━━━━━━━━━━━━ 1s 295us/step
 7009/12086 ━━━━━━━━━━━━━━━━━━━━ 1s 293us/step
 7009/12086 ━━━━━━━━━━━━━━━━━━━━ 1s 293us/step
```

```
7181/12086 1s 293us/step
7181/12086 1s 293us/step
7393/12086 1s 291us/step
7393/12086 1s 291us/step
7602/12086 1s 294us/step
7602/12086 1s 294us/step
7794/12086 1s 293us/step
7794/12086 1s 293us/step
8011/12086 1s 291us/step
8011/12086 1s 291us/step
8227/12086 1s 290us/step
8227/12086 1s 290us/step
8443/12086 1s 288us/step
8443/12086 1s 288us/step
8658/12086 0s 287us/step
8658/12086 0s 287us/step
8874/12086 0s 285us/step
8874/12086 0s 285us/step
9092/12086 0s 284us/step
9092/12086 0s 284us/step
9306/12086 0s 283us/step
9306/12086 0s 283us/step
9526/12086 0s 282us/step
9526/12086 0s 282us/step
```

```
 9743/12086 0s 281us/step
 9743/12086 0s 281us/step
 9959/12086 0s 280us/step
 9959/12086 0s 280us/step
10162/12086 0s 279us/step
10162/12086 0s 279us/step
10377/12086 0s 278us/step
10377/12086 0s 278us/step
10598/12086 0s 277us/step
10598/12086 0s 277us/step
10815/12086 0s 276us/step
10815/12086 0s 276us/step
11025/12086 0s 275us/step
11025/12086 0s 275us/step
11242/12086 0s 274us/step
11242/12086 0s 274us/step
11445/12086 0s 274us/step
11445/12086 0s 274us/step
11668/12086 0s 273us/step
11668/12086 0s 273us/step
11890/12086 0s 272us/step
11890/12086 0s 272us/step
12086/12086 3s 271us/step
12086/12086 3s 271us/step
   1/3022 32s 11ms/step
   1/3022 32s 11ms/step
```

```
 209/3022 ━━━━━━━━━━━━━━━━━━━━ 0s 241us/step
 209/3022 ━━━━━━━━━━━━━━━━━━━━ 0s 241us/step
 423/3022 ━━━━━━━━━━━━━━━━━━━━ 0s 238us/step
 423/3022 ━━━━━━━━━━━━━━━━━━━━ 0s 238us/step
 638/3022 ━━━━━━━━━━━━━━━━━━━━ 0s 236us/step
 638/3022 ━━━━━━━━━━━━━━━━━━━━ 0s 236us/step
 856/3022 ━━━━━━━━━━━━━━━━━━━━ 0s 235us/step
 856/3022 ━━━━━━━━━━━━━━━━━━━━ 0s 235us/step
1072/3022 ━━━━━━━━━━━━━━━━━━━━ 0s 234us/step
1072/3022 ━━━━━━━━━━━━━━━━━━━━ 0s 234us/step
1288/3022 ━━━━━━━━━━━━━━━━━━━━ 0s 234us/step
1288/3022 ━━━━━━━━━━━━━━━━━━━━ 0s 234us/step
1503/3022 ━━━━━━━━━━━━━━━━━━━━ 0s 234us/step
1503/3022 ━━━━━━━━━━━━━━━━━━━━ 0s 234us/step
1723/3022 ━━━━━━━━━━━━━━━━━━━━ 0s 233us/step
1723/3022 ━━━━━━━━━━━━━━━━━━━━ 0s 233us/step
1939/3022 ━━━━━━━━━━━━━━━━━━━━ 0s 233us/step
1939/3022 ━━━━━━━━━━━━━━━━━━━━ 0s 233us/step
2155/3022 ━━━━━━━━━━━━━━━━━━━━ 0s 233us/step
2155/3022 ━━━━━━━━━━━━━━━━━━━━ 0s 233us/step
2372/3022 ━━━━━━━━━━━━━━━━━━━━ 0s 233us/step
2372/3022 ━━━━━━━━━━━━━━━━━━━━ 0s 233us/step
2588/3022 ━━━━━━━━━━━━━━━━━━━━ 0s 232us/step
2588/3022 ━━━━━━━━━━━━━━━━━━━━ 0s 232us/step
```

```
2806/3022 ━━━━━━━━━━━━━━━━━━━━ 0s 232us/step
2806/3022 ━━━━━━━━━━━━━━━━━━━━ 0s 232us/step
3022/3022 ━━━━━━━━━━━━━━━━━━━━ 1s 232us/step
3022/3022 ━━━━━━━━━━━━━━━━━━━━ 1s 232us/step
Train RMSE: 3.47
Test RMSE: 3.56
Train RMSE: 3.47
Test RMSE: 3.56
```

In [ ]:

```
```

References

We report here relevant references: Loy, J. (2019). Neural network projects with Python : the ultimate guide to using Python to explore the true power of neural networks through six projects. http://cds.cern.ch/record/2671438