ymknwvp6j

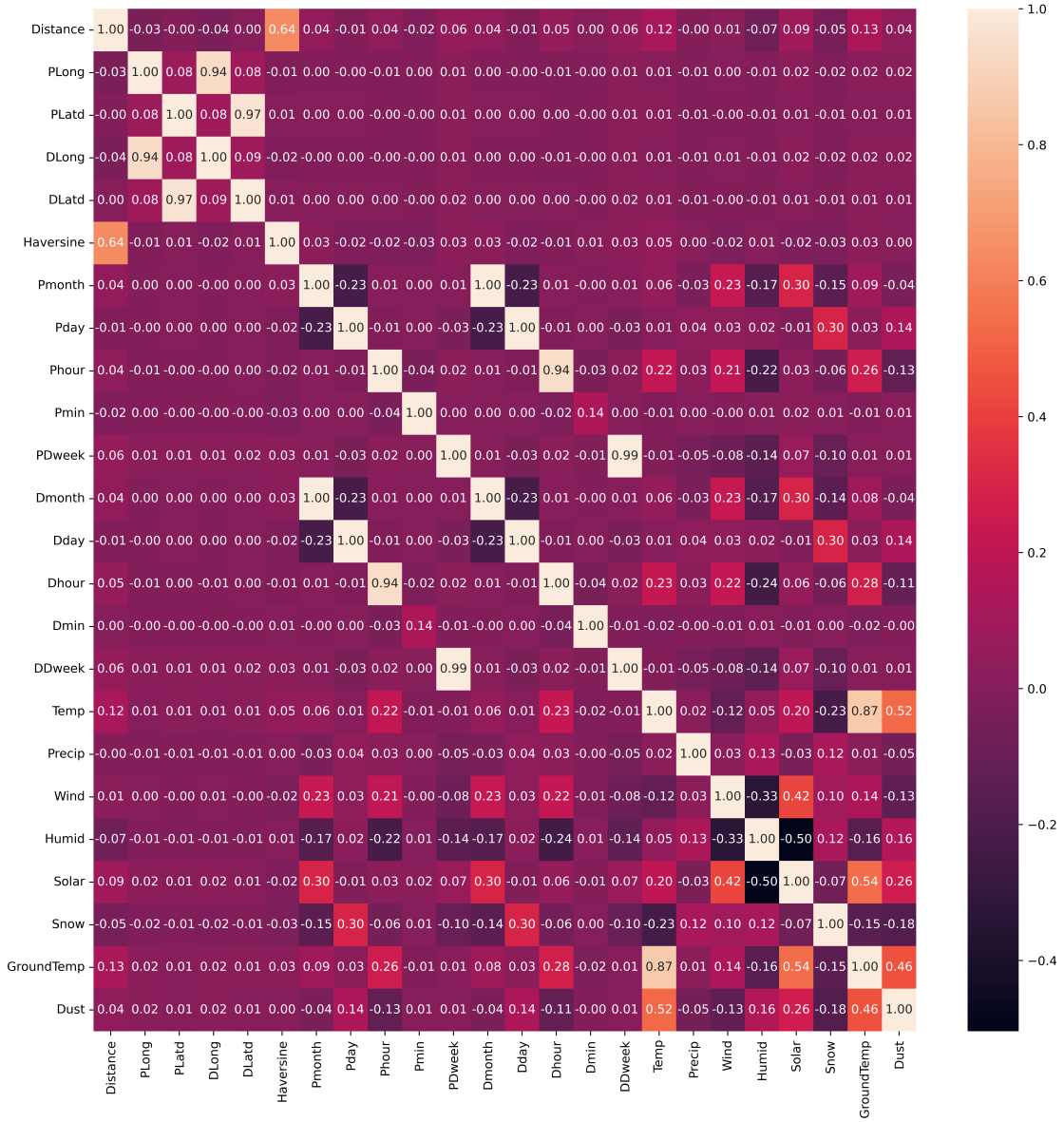
April 29, 2024

[ ]:
```
import pandas as pd
import seaborn as sns
import matplotlib
import matplotlib.pyplot as plt
matplotlib.rcParams['agg.path.chunksize'] = 1024
from xgboost import XGBRegressor
from lightgbm import LGBMRegressor,plot_importance
from sklearn.ensemble import GradientBoostingRegressor, AdaBoostRegressor
from sklearn.metrics import mean_squared_error
from sklearn.model_selection import train_test_split
import gc
import warnings
warnings.filterwarnings("ignore")
df=pd.read_csv("/content/For_modeling.csv")
df.drop("Unnamed: 0", axis = 1, inplace=True)
```

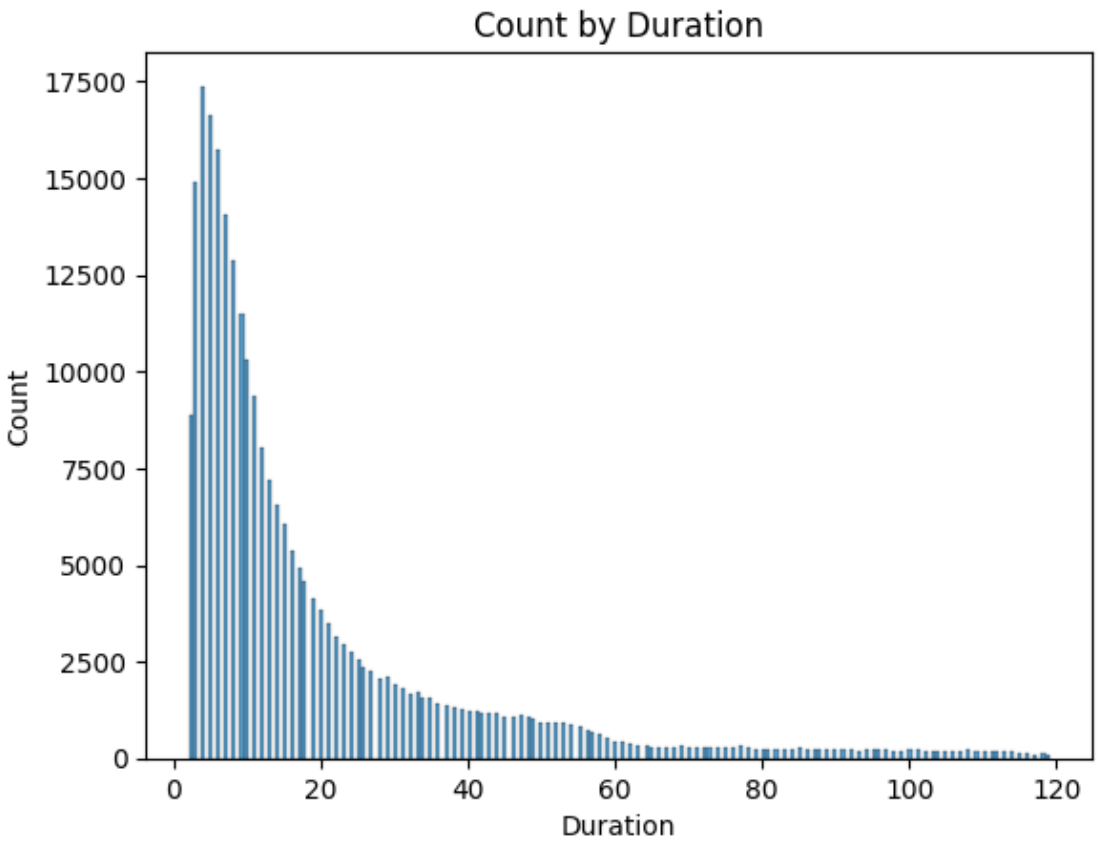
[ ]:
```
X= df.drop("Duration", axis = 1)
#Here X contains all the features except Duration
#Here y contains  Duration features
```

[ ]:
```
fig, axes = plt.subplots(nrows = 1, ncols = 1,figsize = (15, 15), dpi=800)
sns.heatmap(X.corr(), annot  = True, fmt = ".2f")
```

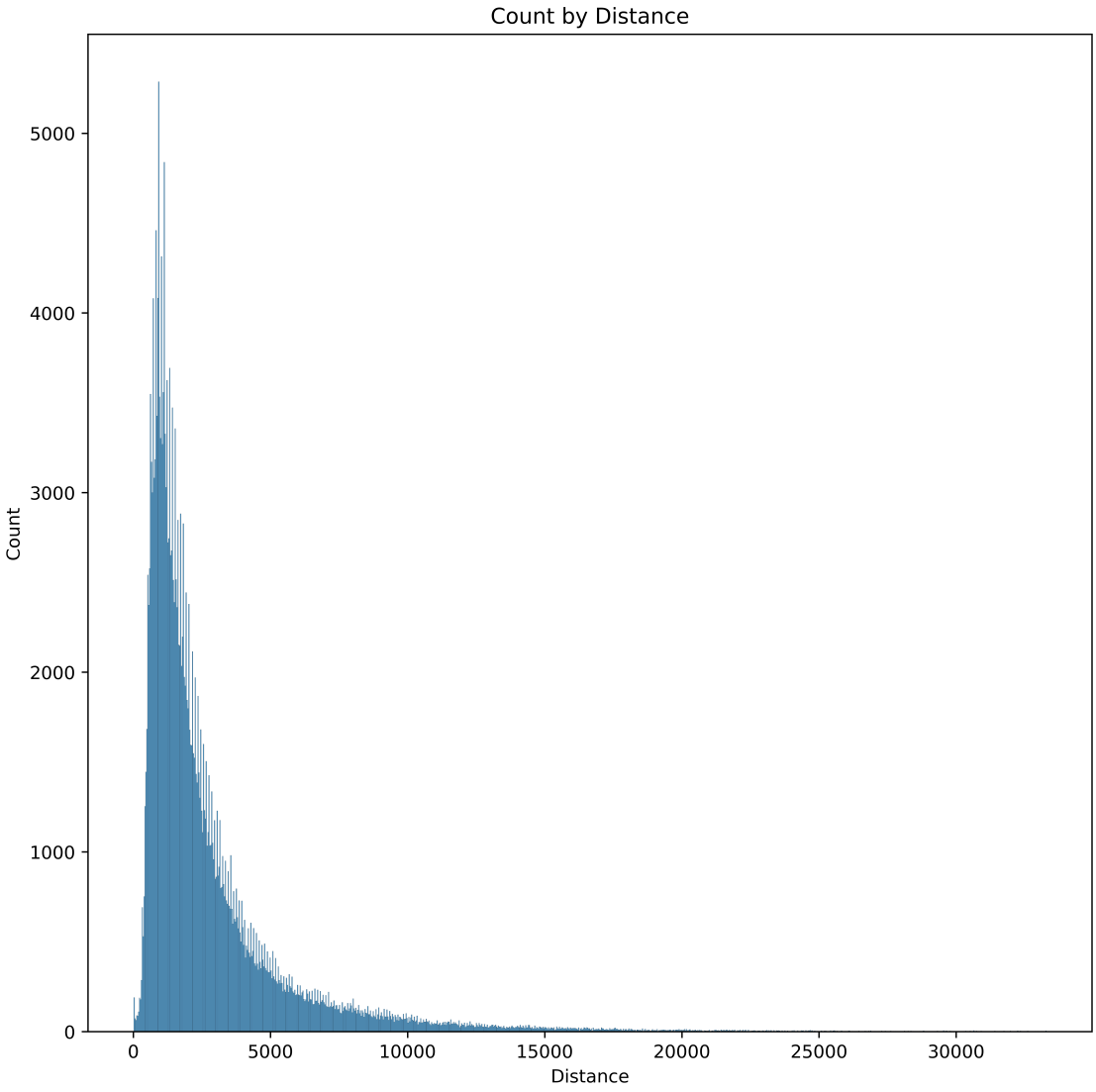
[ ]:
```
<Axes: >
```



[ ]:
```
y=df.Duration

sns.histplot(y)
plt.title('Count by Duration')
plt.show()
#X is an input feature and y is target variable
```



[ ]:
```
fig, axes = plt.subplots(nrows = 1, ncols = 1,figsize = (10, 10), dpi=800)
sns.histplot(X.Distance, bins = 1000)
plt.title('Count by Distance')
plt.show()
```



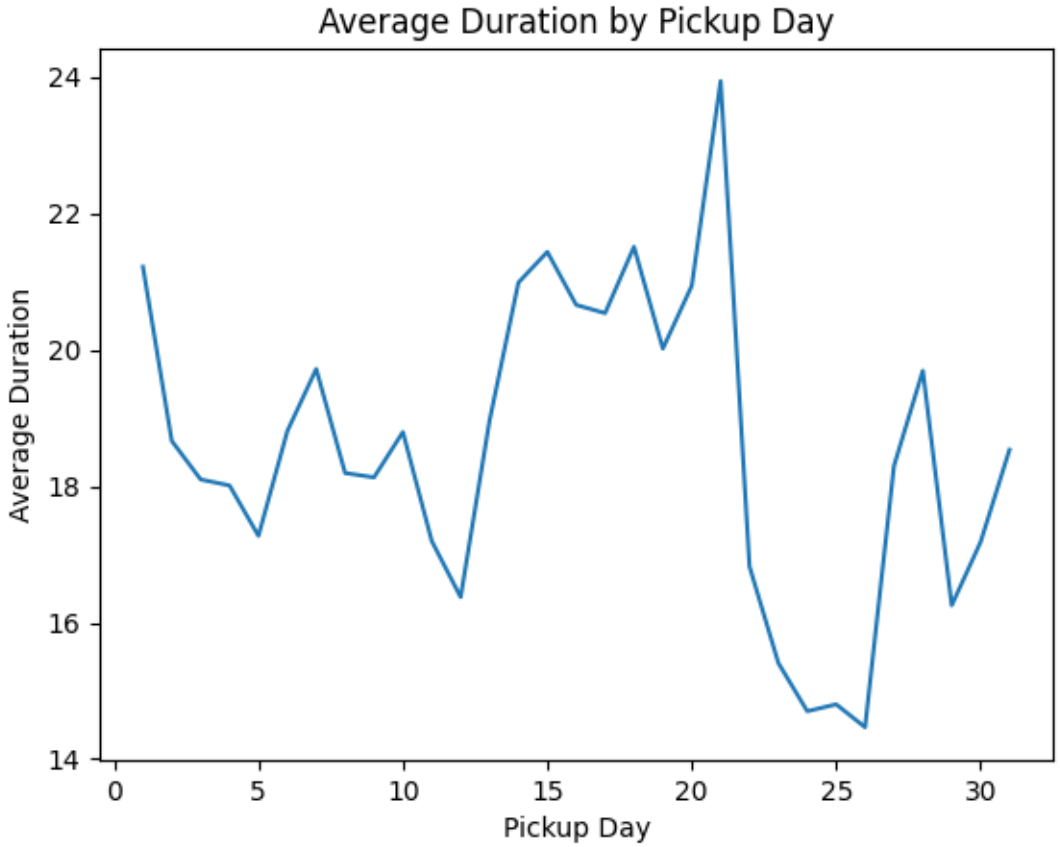
This plot tells us that short trips of upto 5kms are common

[ ]:
```
average_values = df.groupby('Pday')['Duration'].mean().reset_index()
average_values.head()
```

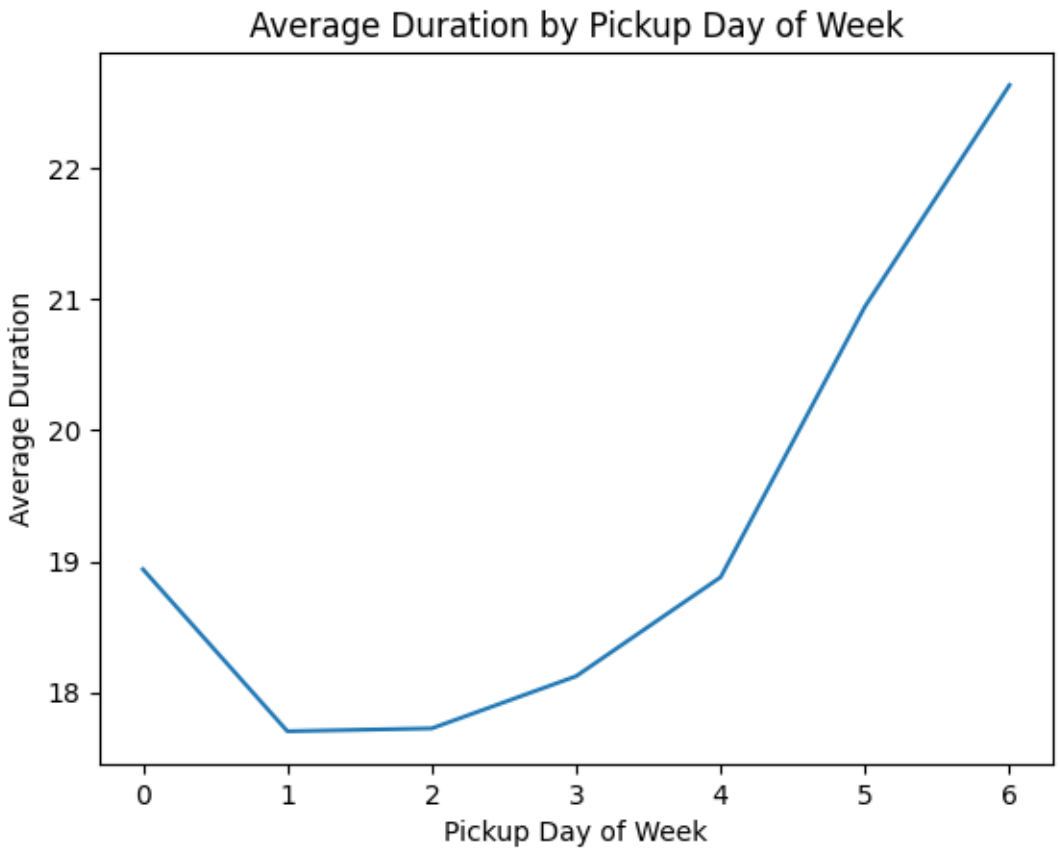
[ ]:

	Pday	Duration
0	1	21.217053
1	2	18.662414
2	3	18.098369
3	4	18.009975
4	5	17.271554

[ ]:
```
sns.lineplot(x='Pday', y='Duration', data=average_values)

plt.xlabel('Pickup Day')
plt.ylabel('Average Duration')
plt.title('Average Duration by Pickup Day')
plt.show()
```

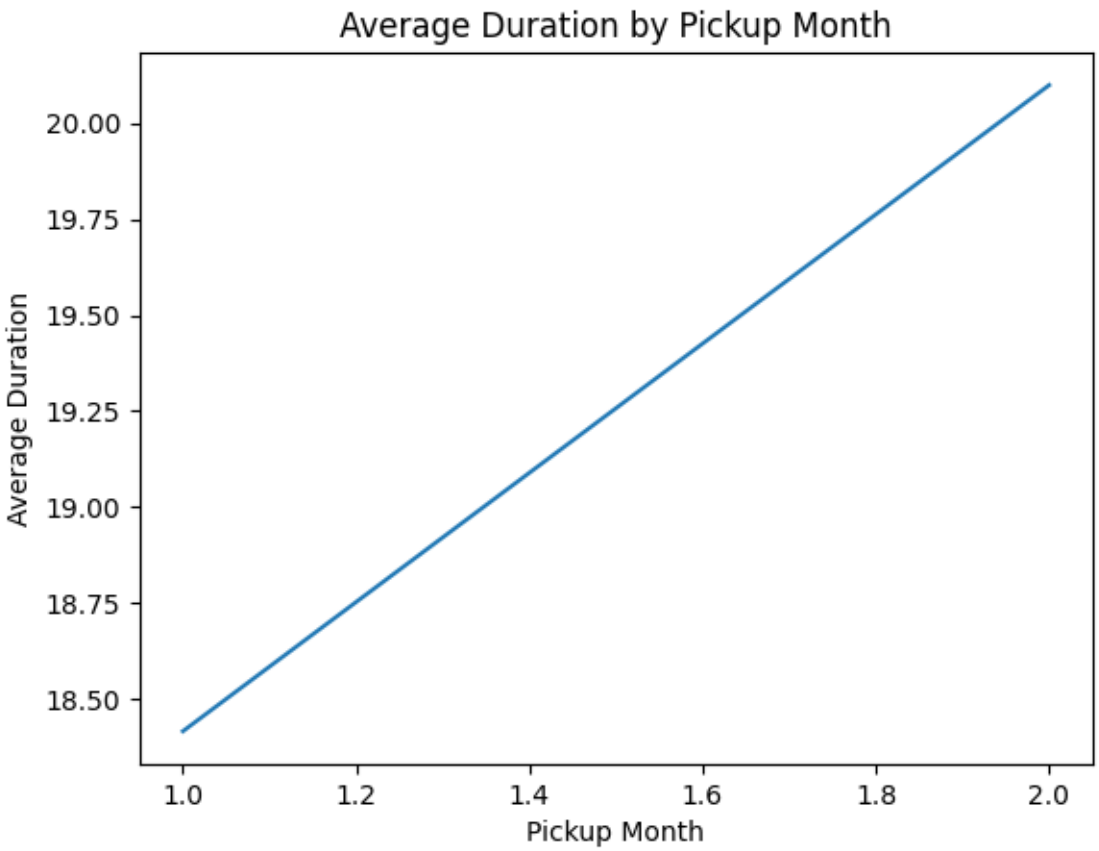


[ ]:
```
average_values = df.groupby('PDweek')['Duration'].mean().reset_index()

sns.lineplot(x='PDweek', y='Duration', data=average_values)

plt.xlabel('Pickup Day of Week')
plt.ylabel('Average Duration')
plt.title('Average Duration by Pickup Day of Week')
plt.show()
```



Weekends are seen as the most busy days in the week where maximum people do for cycling

[ ]:
```
average_values = df.groupby('Pmonth')['Duration'].mean().reset_index()

sns.lineplot(x='Pmonth', y='Duration', data=average_values)

plt.xlabel('Pickup Month')
plt.ylabel('Average Duration')
plt.title('Average Duration by Pickup Month')
plt.show()
```



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

1 Gradient Boosting

[ ]:
```
GBR = GradientBoostingRegressor(n_estimators = 50, learning_rate = 0.1,␣
↪max_depth = 8, subsample = 0.8)
```

[ ]:
```
GBR.fit(X_train, y_train)
gbr_pred = GBR.predict(X_test)
mse_gbr = mean_squared_error(y_test, gbr_pred)
print("Mean squared error of Gradient Boost: ",mse_gbr)
```

```
Mean squared error of Gradient Boost:  72.91531183225224
```

2 Xtreme Gradient Booting

[ ]:
```
XGB = XGBRegressor(n_estimators = 50, learning_rate = 0.1, max_depth = 8,␣
↪subsample = 0.8)
```

[ ]:
```
XGB.fit(X_train, y_train)
xgb_pred = XGB.predict(X_test)
mse_xgb = mean_squared_error(y_test, xgb_pred)
print("Mean squared error of Xtreme Gradient Boost: ",mse_xgb)
```

```
Mean squared error of Xtreme Gradient Boost:  72.1602086554309
```

3 AdaBoost (Adaptive Boosting

[ ]:
```
from sklearn.ensemble import AdaBoostClassifier
AdaBoost = AdaBoostClassifier(n_estimators=50, learning_rate=1)
AdaBoost.fit(X_train, y_train)
Ada_pred = AdaBoost.predict(X_test)
mse_Adb = mean_squared_error(y_test, Ada_pred)
print("Mean squared error of Xtreme Gradient Boost: ",mse_Adb)
```

```
Mean squared error of Xtreme Gradient Boost:  249.31859394888866
```

4 CatBoost with pooling

[ ]:
```
!pip install catboost
```

```
Collecting catboost
  Downloading catboost-1.2.3-cp310-cp310-manylinux2014_x86_64.whl (98.5 MB)
                           98.5/98.5 MB
4.3 MB/s eta 0:00:00
Requirement already satisfied: graphviz in /usr/local/lib/python3.10/dist-
packages (from catboost) (0.20.3)
Requirement already satisfied: matplotlib in /usr/local/lib/python3.10/dist-
packages (from catboost) (3.7.1)
Requirement already satisfied: numpy>=1.16.0 in /usr/local/lib/python3.10/dist-
packages (from catboost) (1.25.2)
Requirement already satisfied: pandas>=0.24 in /usr/local/lib/python3.10/dist-
packages (from catboost) (2.0.3)
Requirement already satisfied: scipy in /usr/local/lib/python3.10/dist-packages
(from catboost) (1.11.4)
Requirement already satisfied: plotly in /usr/local/lib/python3.10/dist-packages
(from catboost) (5.15.0)
Requirement already satisfied: six in /usr/local/lib/python3.10/dist-packages
(from catboost) (1.16.0)
Requirement already satisfied: python-dateutil>=2.8.2 in
```

```
/usr/local/lib/python3.10/dist-packages (from pandas>=0.24->catboost) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.10/dist-
packages (from pandas>=0.24->catboost) (2023.4)
Requirement already satisfied: tzdata>=2022.1 in /usr/local/lib/python3.10/dist-
packages (from pandas>=0.24->catboost) (2024.1)
Requirement already satisfied: contourpy>=1.0.1 in
/usr/local/lib/python3.10/dist-packages (from matplotlib->catboost) (1.2.0)
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.10/dist-
packages (from matplotlib->catboost) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in
/usr/local/lib/python3.10/dist-packages (from matplotlib->catboost) (4.50.0)
Requirement already satisfied: kiwisolver>=1.0.1 in
/usr/local/lib/python3.10/dist-packages (from matplotlib->catboost) (1.4.5)
Requirement already satisfied: packaging>=20.0 in
/usr/local/lib/python3.10/dist-packages (from matplotlib->catboost) (24.0)
Requirement already satisfied: pillow>=6.2.0 in /usr/local/lib/python3.10/dist-
packages (from matplotlib->catboost) (9.4.0)
Requirement already satisfied: pyparsing>=2.3.1 in
/usr/local/lib/python3.10/dist-packages (from matplotlib->catboost) (3.1.2)
Requirement already satisfied: tenacity>=6.2.0 in
/usr/local/lib/python3.10/dist-packages (from plotly->catboost) (8.2.3)
Installing collected packages: catboost
Successfully installed catboost-1.2.3
```

[ ]:
```
from catboost import CatBoostRegressor, Pool
CatB = CatBoostRegressor(n_estimators = 50, learning_rate = 0.1, max_depth = 8,␣
 ↪subsample = 0.8)
```

[ ]:
```
p_train = Pool(X_train, y_train, cat_features = ['Pmonth', 'Pday', 'Phour',␣
 ↪'PDweek', 'Dmonth', 'Dday', 'Dhour', 'DDweek'])
p_test = Pool(X_test, y_test, cat_features = ['Pmonth', 'Pday', 'Phour',␣
 ↪'PDweek', 'Dmonth', 'Dday', 'Dhour', 'DDweek'])
CatB.fit(p_train)
catb_pred = CatB.predict(p_test)
mse_cat1 = mean_squared_error(y_test, catb_pred)
print("Mean squared error of Cat Boost with pool: ",mse_cat1)
```

```
0:      learn: 19.7612394       total: 405ms    remaining: 19.8s
1:      learn: 18.7534004       total: 680ms    remaining: 16.3s
2:      learn: 17.8940476       total: 1.09s    remaining: 17s
3:      learn: 17.1551706       total: 1.36s    remaining: 15.7s
4:      learn: 16.5091069       total: 1.62s    remaining: 14.6s
5:      learn: 15.9659869       total: 1.9s     remaining: 14s
6:      learn: 15.4977533       total: 2.13s    remaining: 13.1s
7:      learn: 15.0961702       total: 2.39s    remaining: 12.6s
8:      learn: 14.7551969       total: 2.67s    remaining: 12.2s
9:      learn: 14.4761545       total: 3s       remaining: 12s
10:     learn: 14.2363743       total: 3.3s     remaining: 11.7s
```

```
11:     learn: 14.0293282       total: 3.83s    remaining: 12.1s
12:     learn: 13.8580288       total: 4.33s    remaining: 12.3s
13:     learn: 13.7158931       total: 4.88s    remaining: 12.5s
14:     learn: 13.5949207       total: 5.41s    remaining: 12.6s
15:     learn: 13.4917783       total: 5.94s    remaining: 12.6s
16:     learn: 13.4042889       total: 6.33s    remaining: 12.3s
17:     learn: 13.3247697       total: 6.67s    remaining: 11.9s
18:     learn: 13.2594118       total: 6.99s    remaining: 11.4s
19:     learn: 13.2082827       total: 7.36s    remaining: 11s
20:     learn: 13.1649230       total: 7.71s    remaining: 10.7s
21:     learn: 13.1306393       total: 8s       remaining: 10.2s
22:     learn: 13.0954461       total: 8.28s    remaining: 9.71s
23:     learn: 13.0580864       total: 8.61s    remaining: 9.33s
24:     learn: 13.0311797       total: 8.97s    remaining: 8.97s
25:     learn: 13.0031121       total: 9.26s    remaining: 8.54s
26:     learn: 12.9805589       total: 9.58s    remaining: 8.16s
27:     learn: 12.9606946       total: 9.91s    remaining: 7.79s
28:     learn: 12.8738188       total: 10.3s    remaining: 7.45s
29:     learn: 12.8219820       total: 10.6s    remaining: 7.05s
30:     learn: 12.7490232       total: 10.9s    remaining: 6.68s
31:     learn: 12.6885097       total: 11.4s    remaining: 6.39s
32:     learn: 12.6190092       total: 11.8s    remaining: 6.09s
33:     learn: 12.5557238       total: 12.2s    remaining: 5.76s
34:     learn: 12.4778871       total: 12.7s    remaining: 5.45s
35:     learn: 12.3730740       total: 13.2s    remaining: 5.13s
36:     learn: 12.2723052       total: 13.6s    remaining: 4.76s
37:     learn: 12.2570089       total: 13.9s    remaining: 4.39s
38:     learn: 12.1683085       total: 14.3s    remaining: 4.03s
39:     learn: 12.1217285       total: 14.6s    remaining: 3.64s
40:     learn: 12.0281739       total: 14.9s    remaining: 3.28s
41:     learn: 12.0081414       total: 15.3s    remaining: 2.91s
42:     learn: 11.9031966       total: 15.7s    remaining: 2.55s
43:     learn: 11.8658375       total: 16.1s    remaining: 2.19s
44:     learn: 11.8443958       total: 16.4s    remaining: 1.82s
45:     learn: 11.8213789       total: 16.8s    remaining: 1.46s
46:     learn: 11.7565361       total: 17.4s    remaining: 1.11s
47:     learn: 11.7446443       total: 18.1s    remaining: 755ms
48:     learn: 11.7248969       total: 18.6s    remaining: 379ms
49:     learn: 11.6000826       total: 19.1s    remaining: 0us
Mean squared error of Cat Boost with pool:  130.9364943527845
```

5 CatBoost without pooling

[ ]:
```
p_train = Pool(X_train, y_train)
p_test = Pool(X_test, y_test)
CatB.fit(p_train)
catb2_pred = CatB.predict(p_test)
```

```
mse_cat2 = mean_squared_error(y_test, catb2_pred)
print("Mean squared error of Cat Boost with pool: ",mse_cat2)
```

```
0:      learn: 19.7486593       total: 119ms    remaining: 5.84s
1:      learn: 18.7376528       total: 196ms    remaining: 4.7s
2:      learn: 17.8729434       total: 309ms    remaining: 4.84s
3:      learn: 17.1377454       total: 424ms    remaining: 4.88s
4:      learn: 16.4964221       total: 524ms    remaining: 4.72s
5:      learn: 15.9399925       total: 633ms    remaining: 4.64s
6:      learn: 15.4713492       total: 726ms    remaining: 4.46s
7:      learn: 15.0663015       total: 779ms    remaining: 4.09s
8:      learn: 14.7257197       total: 827ms    remaining: 3.77s
9:      learn: 14.4421889       total: 880ms    remaining: 3.52s
10:     learn: 14.1947970       total: 929ms    remaining: 3.29s
11:     learn: 13.9796013       total: 974ms    remaining: 3.08s
12:     learn: 13.7980767       total: 1.02s    remaining: 2.91s
13:     learn: 13.6348112       total: 1.07s    remaining: 2.76s
14:     learn: 13.4823013       total: 1.13s    remaining: 2.64s
15:     learn: 13.3431539       total: 1.19s    remaining: 2.52s
16:     learn: 13.2437789       total: 1.24s    remaining: 2.41s
17:     learn: 13.1258088       total: 1.29s    remaining: 2.29s
18:     learn: 13.0416220       total: 1.34s    remaining: 2.19s
19:     learn: 12.9325069       total: 1.41s    remaining: 2.12s
20:     learn: 12.8025180       total: 1.47s    remaining: 2.03s
21:     learn: 12.6978969       total: 1.52s    remaining: 1.94s
22:     learn: 12.5598454       total: 1.58s    remaining: 1.85s
23:     learn: 12.4636737       total: 1.64s    remaining: 1.78s
24:     learn: 12.3836853       total: 1.69s    remaining: 1.69s
25:     learn: 12.3324924       total: 1.74s    remaining: 1.6s
26:     learn: 12.2558165       total: 1.79s    remaining: 1.52s
27:     learn: 12.2307518       total: 1.83s    remaining: 1.44s
28:     learn: 12.1598558       total: 1.89s    remaining: 1.37s
29:     learn: 12.1361139       total: 1.94s    remaining: 1.29s
30:     learn: 12.0935438       total: 1.98s    remaining: 1.22s
31:     learn: 12.0031567       total: 2.07s    remaining: 1.16s
32:     learn: 11.9272123       total: 2.16s    remaining: 1.11s
33:     learn: 11.8912329       total: 2.28s    remaining: 1.07s
34:     learn: 11.8381586       total: 2.4s     remaining: 1.03s
35:     learn: 11.8011910       total: 2.51s    remaining: 977ms
36:     learn: 11.7226906       total: 2.63s    remaining: 922ms
37:     learn: 11.7063223       total: 2.7s     remaining: 854ms
38:     learn: 11.6410179       total: 2.81s    remaining: 792ms
39:     learn: 11.4704843       total: 2.92s    remaining: 731ms
40:     learn: 11.4423845       total: 3.03s    remaining: 666ms
41:     learn: 11.3819677       total: 3.13s    remaining: 597ms
42:     learn: 11.3260746       total: 3.24s    remaining: 527ms
43:     learn: 11.2284533       total: 3.35s    remaining: 457ms
```

```
44:     learn: 11.1181355       total: 3.5s     remaining: 389ms
45:     learn: 11.0781470       total: 3.64s    remaining: 316ms
46:     learn: 11.0153945       total: 3.8s     remaining: 242ms
47:     learn: 10.9611117       total: 3.91s    remaining: 163ms
48:     learn: 10.8601292       total: 4.03s    remaining: 82.2ms
49:     learn: 10.8474142       total: 4.13s    remaining: 0us
Mean squared error of Cat Boost with pool:  115.12235129338926
```

Accuracy Comparisons

[ ]:
```
from sklearn.metrics import accuracy_score
accuracyBG = accuracy_score(y_test, gbr_pred.round())
print(f"Accuracy of Gradient Boost: {accuracyBG:.4f}")
accuracyXGB = accuracy_score(y_test, xgb_pred.round())
print(f"Accuracy of XG boost: {accuracyXGB:.4f}")
accuracyCATB = accuracy_score(y_test, catb_pred.round())
print(f"Accuracy of CATBoost (pooling): {accuracyCATB:.4f}")
accuracyCATB2 = accuracy_score(y_test, catb2_pred.round())
print(f"Accuracy of CATBoost (non-pooling): {accuracyCATB2:.4f}")
accuracyCAD = accuracy_score(y_test, Ada_pred.round())
print(f"Accuracy of CAD Boost: {accuracyCAD:.4f}")
```

```
Accuracy of Gradient Boost: 0.1472
Accuracy of XG boost: 0.1474
Accuracy of CATBoost (pooling): 0.1068
Accuracy of CATBoost (non-pooling): 0.1068
Accuracy of CAD Boost: 0.1287
```

[ ]:
```
acc = {
    "MSE Gradient Boost" : mse_gbr,
    "MSE Xtreme GB" : mse_xgb,
    "MSE AdaBoost" : mse_Adb,
    "MSE CatBoost with pool" : mse_cat1,
    "MSE CatBoost without pool" : mse_cat2
}
```

[ ]:
```
acc = dict(sorted(acc.items()))
for i in acc:
    print(i, " : ", acc[i])
```

```
MSE AdaBoost  :  249.31859394888866
MSE CatBoost with pool  :  130.9364943527845
MSE CatBoost without pool  :  115.12235129338926
MSE Gradient Boost  :  72.91531183225224
MSE Xtreme GB  :  72.1602086554309
```

[ ]:
```
# fig, axes = plt.subplots(nrows = 1, ncols = 1,figsize = (7,7), dpi=800)
plt.bar(list(acc.keys()), list(acc.values()))
```

```
plt.title("MSE of all models")
plt.xticks(rotation=20)
plt.show()
```