Predicting Bike Sharing Demand in Seoul and D.C.

Team KKL

Soyoung Kim, Pavel Kovalev

Problem Statement

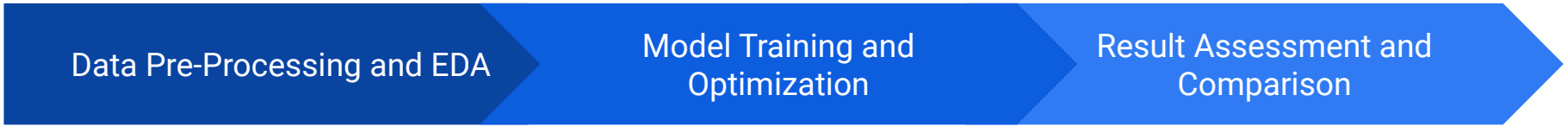
Goal: predict bike sharing demand in Seoul and D.C. based on a variety of features.

Why predict bike sharing demand?

- Accurate estimation of future bike usage is crucial for ensuring the availability of bikes when and where they are needed.
- Effective prediction models can optimize bike allocation, maintenance, and overall system efficiency.
- Developing accurate demand prediction methods can help improve the usability and accessibility of bike sharing systems.

Stakeholders: bike sharing service providers, bike users, urban planners.

Workflow



- Dataset cleaning
- Feature engineering
- EDA

- Feature and model selection
- Developing pre-processing pipelines
- Tuning hyperparameters

- Comparing final model performance on the complete test set
- Evaluating for overfitting/underfitting
- Contrasting results between Seoul and D.C. datasets

Seoul Data - Description

16 parameters/features:

- Date (1/12/17-11/30/18)
- Bike count
- Hour
- Temperature
- Humidity
- Windspeed
- Visibility
- Dew point temperature
- Solar radiation (UV Strength/Index)
- Rainfall (actual measurement of rainfall)
- Snowfall (actual measurement of snowfall)
- Holiday (total of 18 holidays, 9 of which with very low temperature)
- Functional Day (i.e. business day/whether the rental facilities are open)
- Week status (Weekday or weekend)
- Day of the week
- Season

D.C. Data - Description

14 parameters/features:

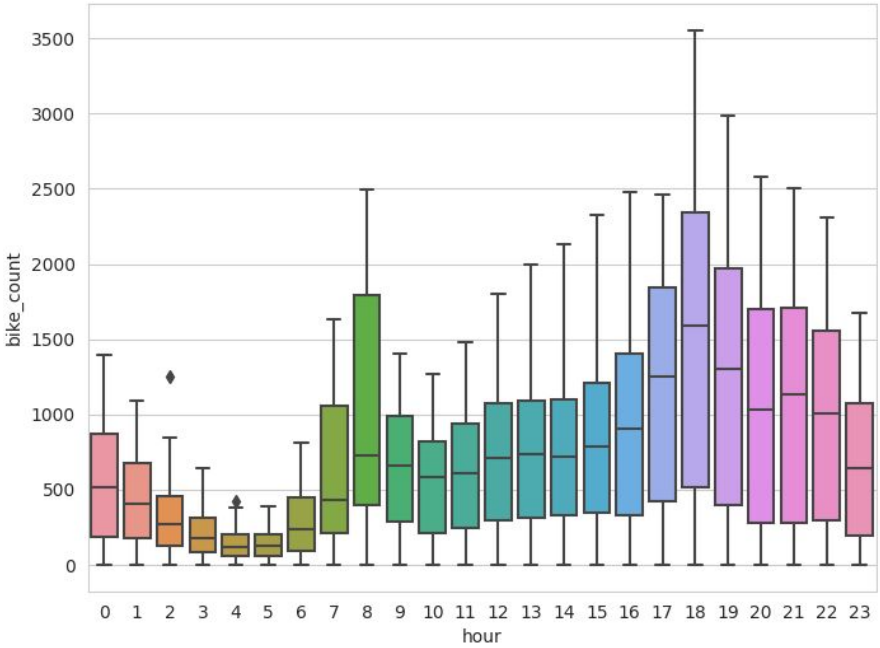
- Date
- Bike count
- season (1:winter, 2:spring, 3:summer, 4:fall)
- Year (0: 2011, 1:2012)
- Month (1 to 12)
- Hour
- Holiday
- Day of the week
- Week status/weekday (1 if weekday, 0 if weekend)
- Weather situation (4 categories)
  - 1: Clear, Few clouds, Partly cloudy, Partly cloudy
  - 2: Mist + Cloudy, Mist + Broken clouds, Mist + Few clouds, Mist
  - 3: Light Snow, Light Rain + Thunderstorm + Scattered clouds, Light Rain + Scattered clouds
  - 4: Heavy Rain + Ice Pallets + Thunderstorm + Mist, Snow + Fog
- Temperature (Normalized).
- “Feels like” temperature (also normalized)
- Humidity
- Windspeed

Comparison

- Compared to Seoul data, DC data is missing visibility and UV Index, which do not seem to be significant factors.
- In DC data, weathers are classified into 4 categories which are close to weather severity scores.
- In terms of the content/quality of information, not much difference
- Instead of dew point temperature, feels-like temperature is used in the DC data.
- Since time series analysis is not being performed for Seoul data, year and month features are omitted from the DC data.

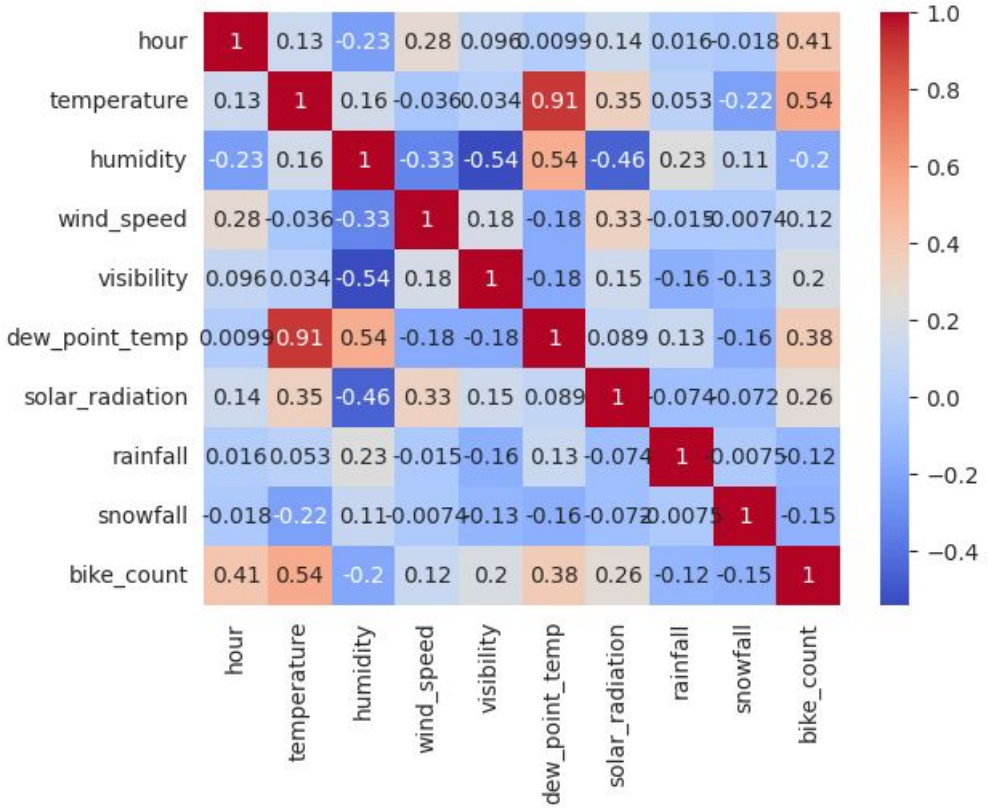
Seoul Data - EDA

- Weekday and holiday seem to be less significant, hour is

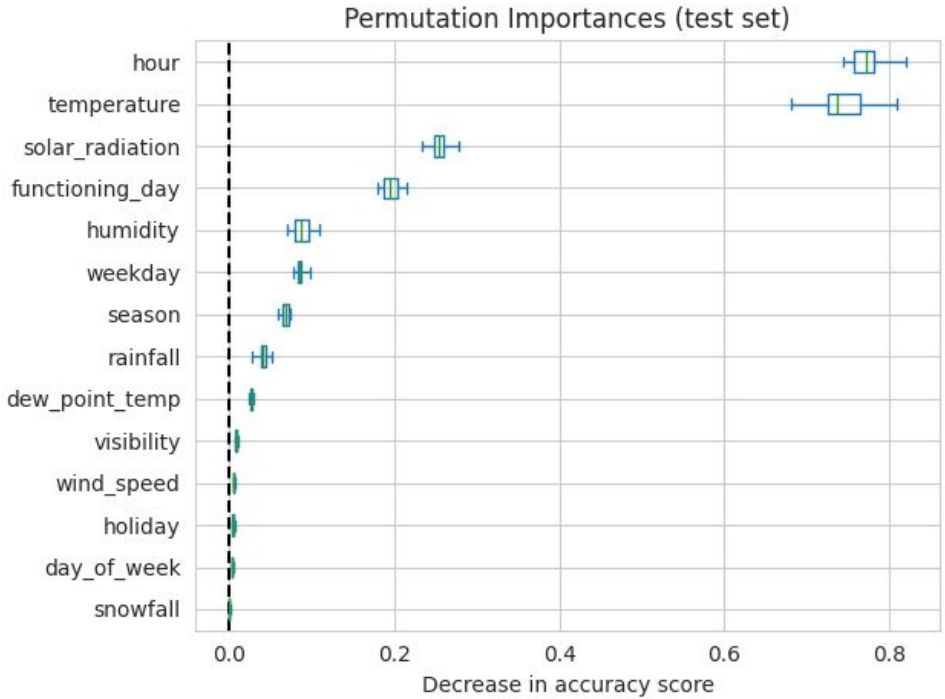


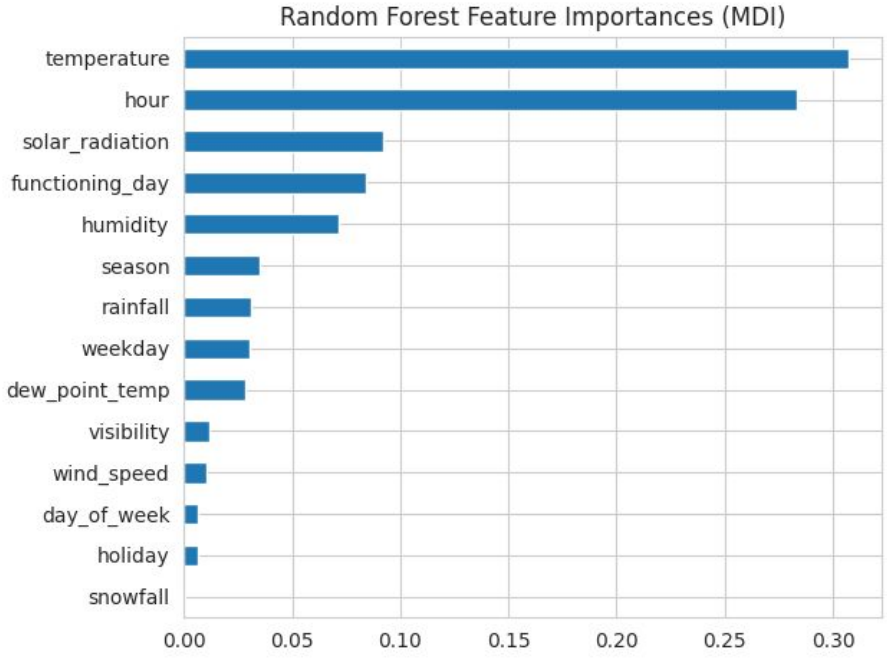
Seoul Data - EDA

- Correlation Table:



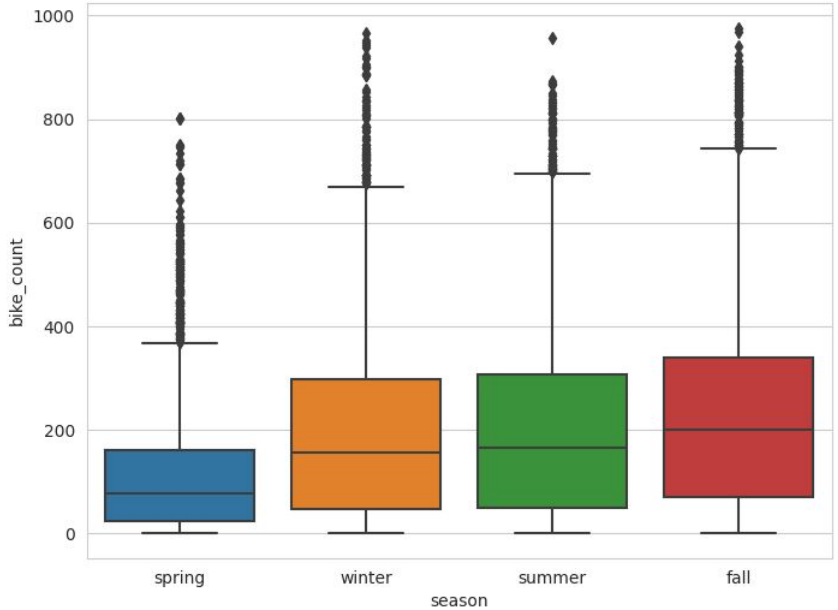
Seoul Data - EDA

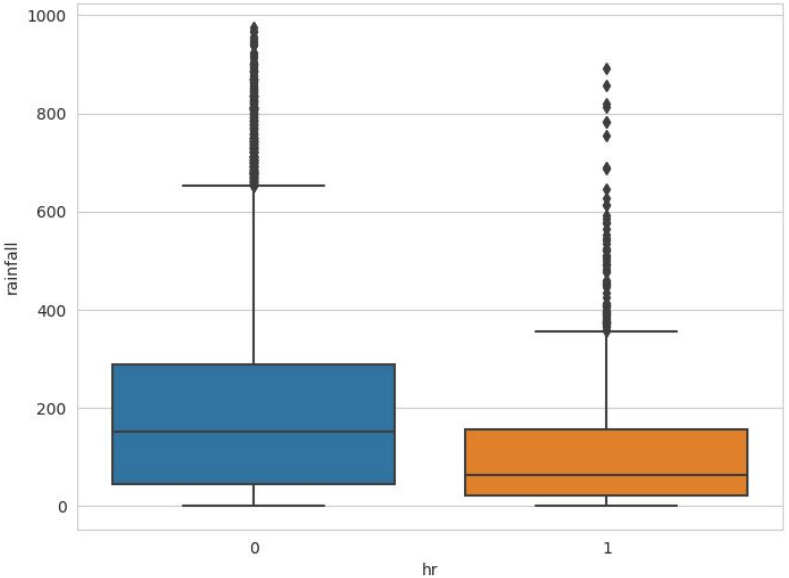




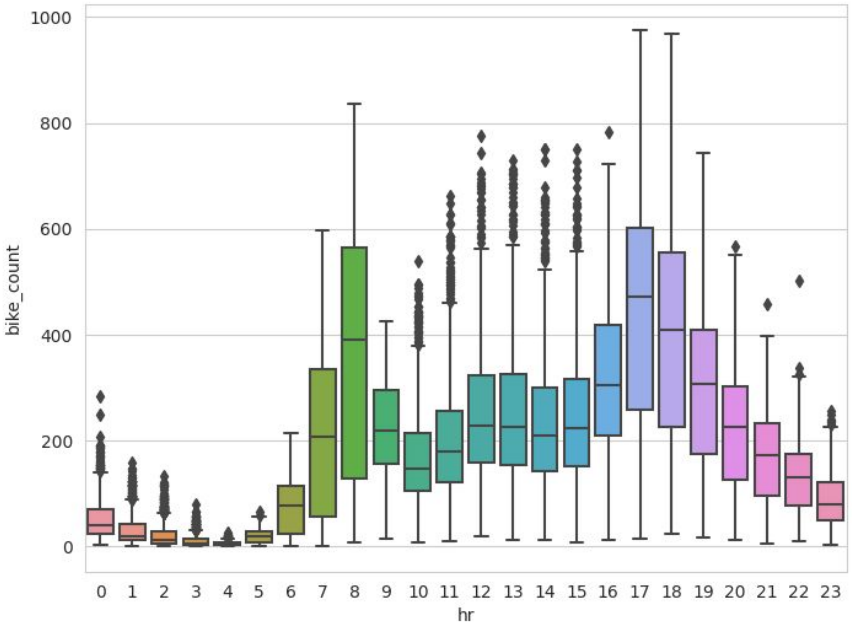
D.C. Data - EDA

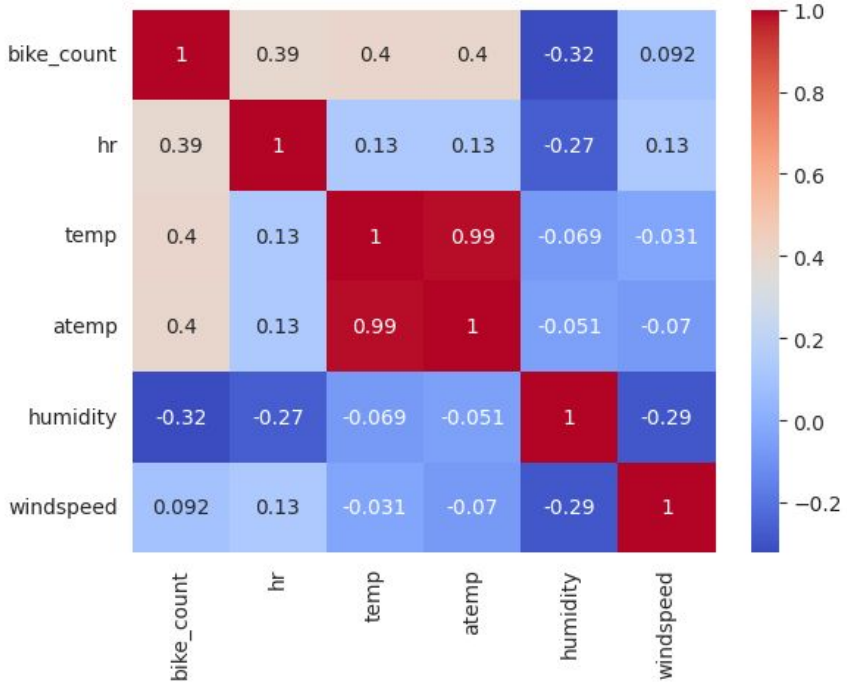
Weekday, weekend were insignificant, but season, weather condition have significance



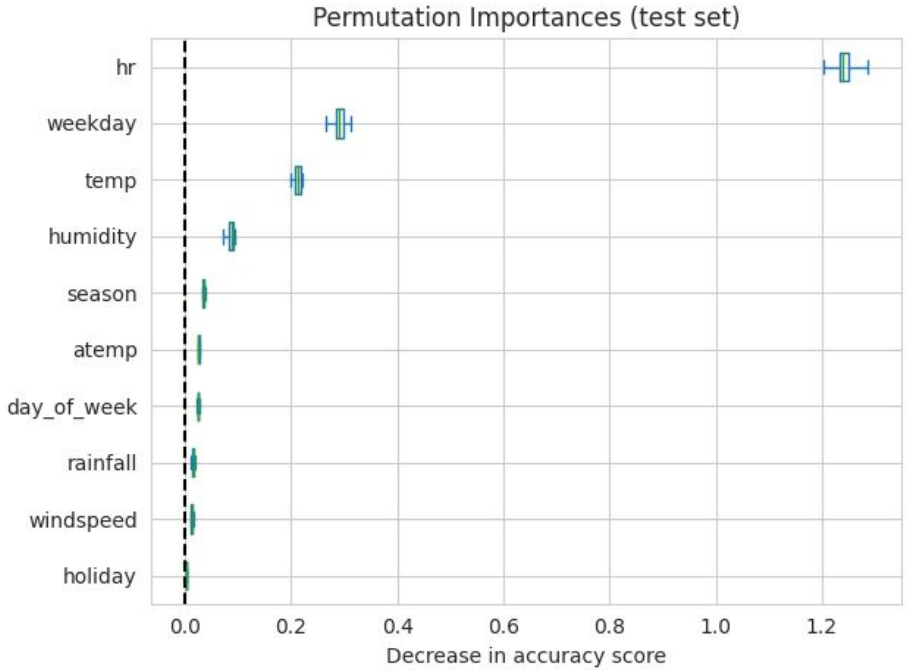


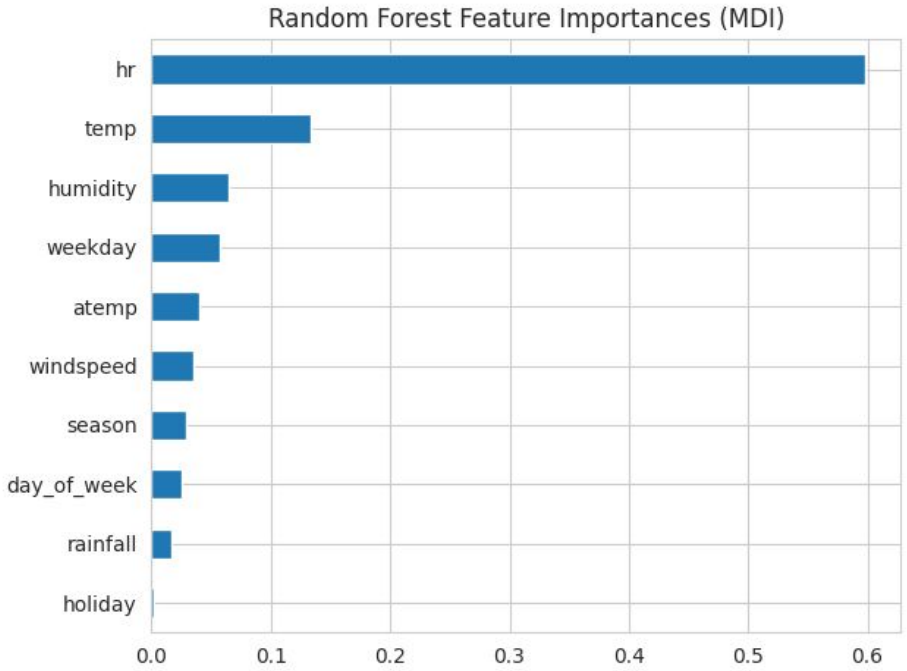
D.C. Data - EDA


bike_count
1000
800
600
400
200
0
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
hr


bike_count 1 0.39 0.4 0.4 -0.32 0.092
hr 0.39 1 0.13 0.13 -0.27 0.13
temp 0.4 0.13 1 0.99 -0.069 -0.031
atemp 0.4 0.13 0.99 1 -0.051 -0.07
humidity -0.32 -0.27 -0.069 -0.051 1 -0.29
windspeed 0.092 0.13 -0.031 -0.07 -0.29 1
bike_count
hr
temp
atemp
humidity
windspeed
1.0
0.8
0.6
0.4
0.2
0.0
−0.2

D.C. Data - EDA


Permutation Importances (test set)
hr
weekday
temp
humidity
season
atemp
day_of_week
rainfall
windspeed
holiday
0.0
0.2
0.4
0.6
0.8
1.0
1.2
Decrease in accuracy score


Random Forest Feature Importances (MDI)
hr
temp
humidity
weekday
atemp
windspeed
season
day_of_week
rainfall
holiday
0.0
0.1
0.2
0.3
0.4
0.5
0.6

Model Training

Models we tried:

- Histogram-based Gradient Boosting Regression
- XGBoosting Regression
- Extra Trees Regression
- Random Forest Regression
- Support Vector Regression
- KNN Regression
- Voter Model

Model Training

Features we tried:

- All features
- A smaller number of most important features according to EDA

There wasn’t any significant improvement in the latter case, so we ended up using all features (for both datasets) to train final models.

Model Training

Training process:

- Creating pre-processing pipeline that scales continuous features and performs ordinal encoding of categorical features
- Tuning hyperparameters with GridSearchCV (up to 4 rounds for each model, since it takes a long time)
- Assessing MSE of the tuned models on the validation set, and then on the entire test set

Model Performance Assessment - Seoul

Regressor	MSE on Validation Set	MSE on Test Set
XGBoost	23178.210901	23464.66243684614
Voter	23780.430148	24296.117394809782
HistGradientBoost	24689.983933	25708.27950378138
ExtraTrees	25194.869371	27369.80028362249
RandomForest	32022.426478	n/a
KNN	108144.778677	n/a
SVR	141993.153056	n/a

Model Performance Assessment - D.C.

Regressor	**MSE on Validation Set**	**MSE on Test Set**
XGBoost	4498.44624	4265.780799091904
HistGradientBoost	4690.298763	4483.082612993278
Voter	4696.552088	4447.998083074991
ExtraTrees	5195.108605	n/a
RandomForest	5291.292368	n/a
KNN	12478.160786	n/a
SVR	16798.227407	n/a

Conclusion

- For both datasets, XGBoost demonstrates the best results, while Voter and HistGradientBoost perform closely behind.
- KNN and SVR consistently perform the worst.
- Our models work better with the DC Data (probably because of differences in feature distribution and in the general nature of the datasets)