Cameron Lee

Assignment 2

Professor Rohit Kate

COMPSCI-411

***Note: This dataset is large, and my program took upwards of an hour to run completely

Dataset – Bike Sharing Demand (ID Number: 42713)

Overview: This dataset is a compilation of a two-year historical log of bicycle rental data from the Capital Bikeshare system located in Washington, D.C.

Task: The task of this dataset is to determine/predict the number of bikes that are being rented during a specific environmental and seasonal setting. This can be useful when identifying when there are inclement weather conditions or when it is summer/ a tourist season. Additionally, it can be useful when determining the reduction in traffic conditions and carbon footprint. [The data was extracted on a two hourly and daily basis.]

Target: The target of this dataset is a specific number (int64) of bikes that are being used by casual and registered riders. [Example: 64 bikes currently being used]

Features: There were two nominal features and 8 numerical features. The nominal features were converted into numerical features by using one-hot encoding. So, the seasons were split up into float64 by season [summer, fall, winter spring] and the weather was split into different types of precipitation [clear, misty, rain, heavy_rain]. After one hot encoding, the new data was put into a dataframe and there ended up being 18 individual features.

```
Data columns (total 18 columns):
 #   Column                      Non-Null Count  Dtype
---  ------                      --------------  -----
 0   encoder__season_fall        17379 non-null  float64
 1   encoder__season_spring      17379 non-null  float64
 2   encoder__season_summer      17379 non-null  float64
 3   encoder__season_winter      17379 non-null  float64
 4   encoder__weather_clear      17379 non-null  float64
 5   encoder__weather_heavy_rain 17379 non-null  float64
 6   encoder__weather_misty      17379 non-null  float64
 7   encoder__weather_rain       17379 non-null  float64
 8   remainder__year             17379 non-null  float64
 9   remainder__month            17379 non-null  float64
 10  remainder__hour             17379 non-null  float64
 11  remainder__holiday          17379 non-null  float64
 12  remainder__weekday          17379 non-null  float64
 13  remainder__workingday       17379 non-null  float64
 14  remainder__temp             17379 non-null  float64
 15  remainder__feel_temp        17379 non-null  float64
 16  remainder__humidity         17379 non-null  float64
 17  remainder__windspeed        17379 non-null  float64
dtypes: float64(18)
memory usage: 2.4 MB
```

Subtask 1: Bagged vs Base Regressors RMSE

Base Regressors:

```
Linear Regression Base Model: Finding RMSE
138.86674356023786

Decision Tree Regressor Base Model: Finding RMSE
75.00104458746662

K Nearest Neighbors Regressor Base Model: Finding RMSE
125.2951469888339

Support-Vector Machines Regressor Base Model: Finding RMSE
180.22318729325204
```

Bagged Regressors:

```
Linear Bagged Regressor: RMSE
I'm not really sure why this number is so big. I copied the powerpoint slide... sorry!
[-4.31296064e+11 -1.05697809e+02 -1.28667155e+02 -1.19298655e+02
 -1.02292191e+02 -1.38805547e+11 -1.73191414e+02 -1.82600157e+02
 -2.05666927e+02 -1.58472910e+02]
57010161189.699974

Decision Tree Bagged Regressor: RMSE
63.06255691303966

K Neighbors Bagged Regressor: RMSE
127.11239050353916

Support-Vector Machine Bagged Regressor: RMSE
180.05988287982603
```

Statistical Significance:

```
Linear Regression: Bagged vs Base
TtestResult(statistic=-1.3011419042720838, pvalue=0.22553178546384484, df=9)

DT Regression: Bagged vs Base
TtestResult(statistic=6.194388818458034, pvalue=0.00015993991602683238, df=9)

KN Regression: Bagged vs Base
TtestResult(statistic=-1.1996372564001188, pvalue=0.2609072579850982, df=9)

SVM Regression: Bagged vs Base
TtestResult(statistic=1.6131347602320762, pvalue=0.14117345863824338, df=9)
```

Table of Base vs Bagged RMSE values:

Method	Linear	Decision Tree	K Nearest	Support-Vector
Base	**138.867**	75.001	**125.295**	180.223
Bagged	57010161189	**63.063***	127.112	**180.060**

Table: Results (RMSE) shown in bold were better when compared in the same the same column. The results marked with * were found to be statistically significant at $p < 0.05$ level when using the two-tailed paired t-test when compared with the result in the same column.

Ultimately, the best model to use for predicting the number of bicycles in use is the decision tree bagged model. A close second was the Decision tree-based model. The decision tree RMSE's were statistically significant/statistically different from each other.

Note on error: Highlighted in red is the linear bagged model RMSE that was given. I am not sure how or why it is that large and that incorrect. I have added my code here (which is the exact same as the professors) in hopes of finding a correct solution.

```
#Linear Bagged
print("Linear Bagged Regressor: RMSE")
bagged_lr = BaggingRegressor(estimator=LinearRegression())
scores = model_selection.cross_validate(bagged_lr, bike_new_data, bike.target,
                                        cv = 10, scoring = "neg_root_mean_squared
print("I'm not really sure why this number is so big. I copied the powerpoint sli
print(scores["test_score"])
#rmseLBagged = RMSE Linear Bagged
rmseLBagged = 0 - scores["test_score"]
print(rmseLBagged.mean())
print()
```

Subtask 2: Boosted vs Base Regressors RMSE

Base Regressors:

```
Linear Regression Base Model: Finding RMSE
138.86674356023786

Decision Tree Regressor Base Model: Finding RMSE
75.00104458746662

K Nearest Neighbors Regressor Base Model: Finding RMSE
125.2951469888339

Support-Vector Machines Regressor Base Model: Finding RMSE
180.22318729325204
```

Boosted Regressors:

```
Linear Boosted Regressor: RMSE
852270524403.109

Decision Tree Boosted Regressor: RMSE
62.07870658256985

K Neighbors Boosted Regressor: RMSE
126.62004786423043

Support-Vector Machine Boosted Regressor: RMSE
180.26494593786157
```

Statistical Significance:

```
Subtask2 - Boosted vs Base Regressors RMSE

Linear Regression: Boosted vs Base
TtestResult(statistic=-1.4989938475697513, pvalue=0.16810669487498395, df=9)

DT Regression: Boosted vs Base

KN Regression: Boosted vs Base
TtestResult(statistic=-1.0493134642837905, pvalue=0.32139309545705563, df=9)

SVM Regression: Boosted vs Base
TtestResult(statistic=-0.96571960432128, pvalue=0.3594156065473799, df=9)
```

Missed DT By Accident:

```
DT Regression: Boosted vs Base
TtestResult(statistic=4.869595846112389, pvalue=0.0008840107590216821, df=9)
```

Table of Base vs Boosted Regressor RMSE:

Method	Linear	Decision Tree	K Nearest	Support-Vector
Base	**138.867**	75.001	**125.295**	**180.223**
Boosted	852270524403	**62.079***	126.620	180.265

Table: Results (RMSE) shown in bold were better when compared to those in the same column. The results marked with * were found to be statistically significant at the $p < 0.5$ level using the two-tailed paired t-test when compared to with the result in the same column.

Again, the best model to use for predicting the number of bicycles in use is the decision tree boosted model. A close second was the Decision tree-based model. The decision tree RMSE's were statistically significant/statistically different from each other.

Note on error: Highlighted in red is the linear boosted model RMSE that was given. I am not sure how or why it is that large and that incorrect. I have added my code here (which is the exact same as the professors) in hopes of finding a correct solution.

```
#Linear Boosted
print("Linear Boosted Regressor: RMSE")
boosted_lr = AdaBoostRegressor(estimator=LinearRegression())
scores = model_selection.cross_validate(boosted_lr, bike_new_data, bike.target,
                                        cv = 10, scoring = "neg_root_mean_squared
scores["test_score"]
#rmseLBoosted = RMSE Linear Regression Boosted
rmseLBoosted= 0 - scores["test_score"]
print(rmseLBoosted.mean())
print()
```

Subtask 3 VotingRegressor vs Base Methods RMSE:

Base Models:

```
Linear Regression Base Model: Finding RMSE
138.86674356023786

Decision Tree Regressor Base Model: Finding RMSE
75.00104458746662

K Nearest Neighbors Regressor Base Model: Finding RMSE
125.2951469888339

Support-Vector Machines Regressor Base Model: Finding RMSE
180.22318729325204
```

All base methods Voting Regressor & Statistical Significance:

```
Voting Regressor: RMSE of ALL base methods
101.9665497163192
Subtask3 - VotingRegressor vs Base Regressors RMSE

Linear Regression: DT vs Base
TtestResult(statistic=5.187969972942325, pvalue=0.0005732024056651599, df=9)

DT Regression: VotingRegressor vs DT
TtestResult(statistic=-2.571435016269495, pvalue=0.03011681345876433, df=9)

KN Regression: DT vs Base
TtestResult(statistic=6.048433957003847, pvalue=0.000190850678098803, df=9)

SVM Regression: DT vs Base
TtestResult(statistic=5.2595770041381735, pvalue=0.0005210264515704163, df=9)
```

Table of VotingRegressor vs Base Models:

	Voted	Linear	Decision Tree	K Nearest	Support-Vec
RMSE	**101.943**	**138.867**	***75.001***	**125.295**	**180.223**

Table: The best result [Decision Tree] is shown in bold and italics. According to the two-tailed paired t-test ran on each of the other Regression Models RMSE's, each method was statistically significantly different than the best result. Each of the results are bolded because of this.

Conclusion:

Overall, the best performing model was the Decision Tree. With an average RMSE score of **66.714** when comparing all forms of the model (base, bagged, and boosted) it stands out from the rest of the field. The best out of the Decision Tree regression models was the Boosted Regressor with an RMSE of **62.079.** The worst model overall was the Linear Regression model with an average RMSE score of **303093561900**. This is a ridiculously high number and it is very incorrect. I am unsure if there is something wrong with my code. The absolute worst was the

Linear Regression Boosted Model in which the RMSE score was **852270524403**. Again, I believe that there is an error somewhere in my code. This is not even possible as there are only about 17,000 instances of data and very likely that there are not that many bikes.

In conclusion, the best performing model was the Decision Tree Boosted Regressor model. It made fairly accurate predictions with an error of about 62 bikes not accounted for. Even though I ran into some linear regression errors, seeing all of the models compared side by side really does show that the best regression model depends on the dataset being used. In my case, a decision tree model was best used and I believe that can be inferred because of the weather conditions that are dependent on the number of bikes being out. It will be easier to use a decision tree to determine when larger numbers of bikes are out versus smaller numbers due to inclement weather.