CLASSIFICATION ANALYSIS: PREDICTING THE GENDER OF NYC CITI BIKE USERS

Presented by Don Macfoy

THE MODERN BIKE SHARE INDUSTRY

- Bike-sharing services offer a means for people to quickly access affordable, short term transportation in urban areas.
- Increased automation has allowed for these services to grow more robust in nature and generate more data.

NYC CITI BIKE BIKE SHARE SYSTEM

- Citi Bike is a public bicycle sharing service that operates in the New York City and Jersey City, New Jersey.
- The service works by allowing users to ride for a predetermined amount of time based on the passes or memberships purchased.
- To further understand how trip information could be used to understand consumer trends, I used the Citi Bike's data to predict gender.

EXPERIMENTAL DESIGN

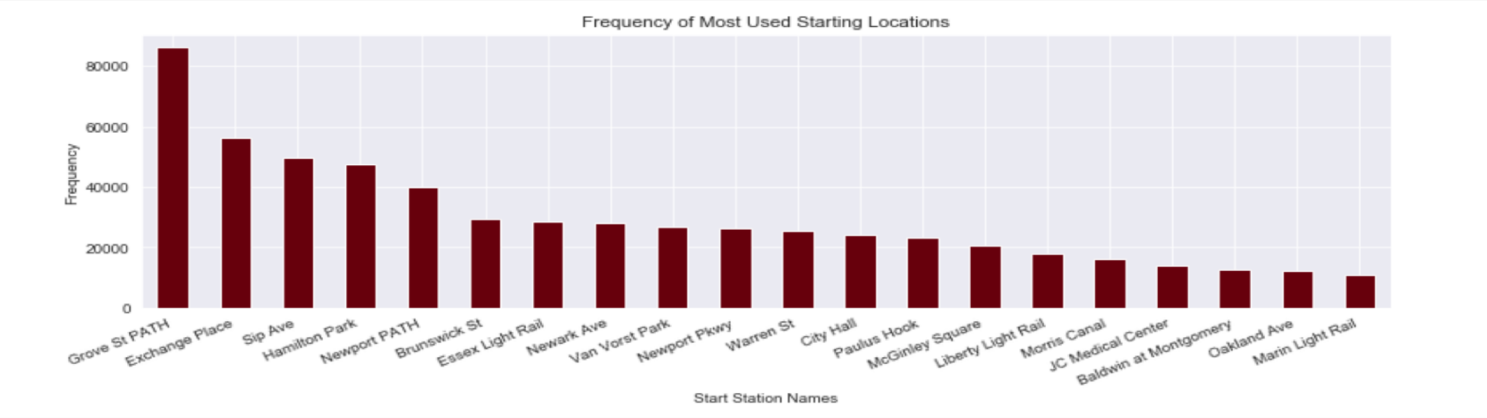
- Research Question: Which supervised learning model is best for predicting the gender of NYC Citi Bike users based on the available data?
- Gender predictions of specific users were made using multiple supervised learning model types.
- Each model type was run using 3 different types of feature selection.
- The Gradient Boosting Model using features selected by sklearn's selectKbest function was expected to be the best performing model.

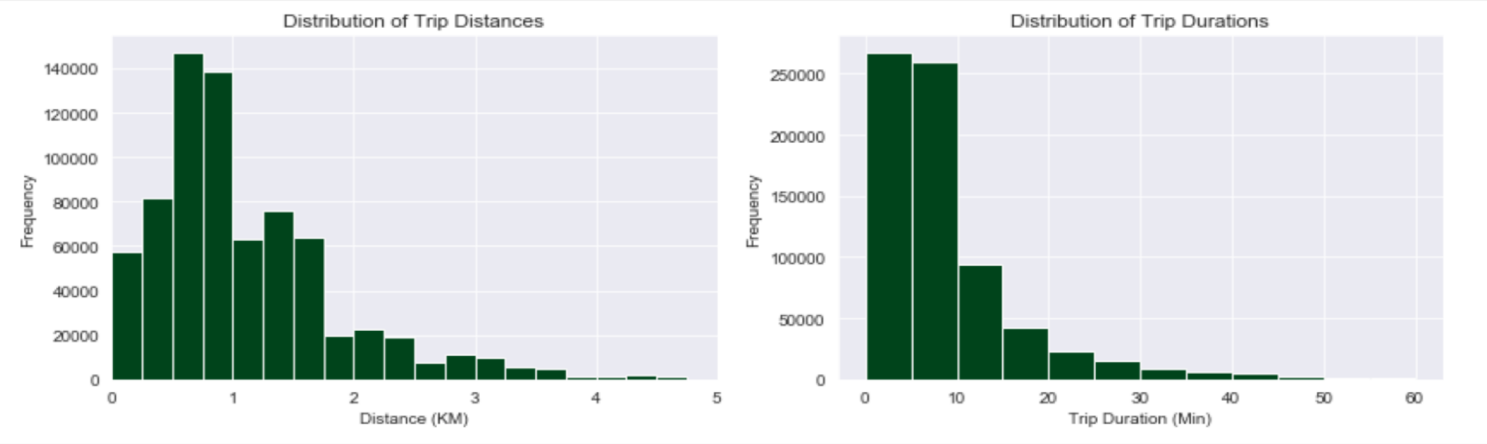
ABOUT THE DATA

- The Citi Bike trip dataset contains information about 735502 anonymized trips that took place between January 2015 and June 2017.
- Features were engineered in addition to those in the original dataset.

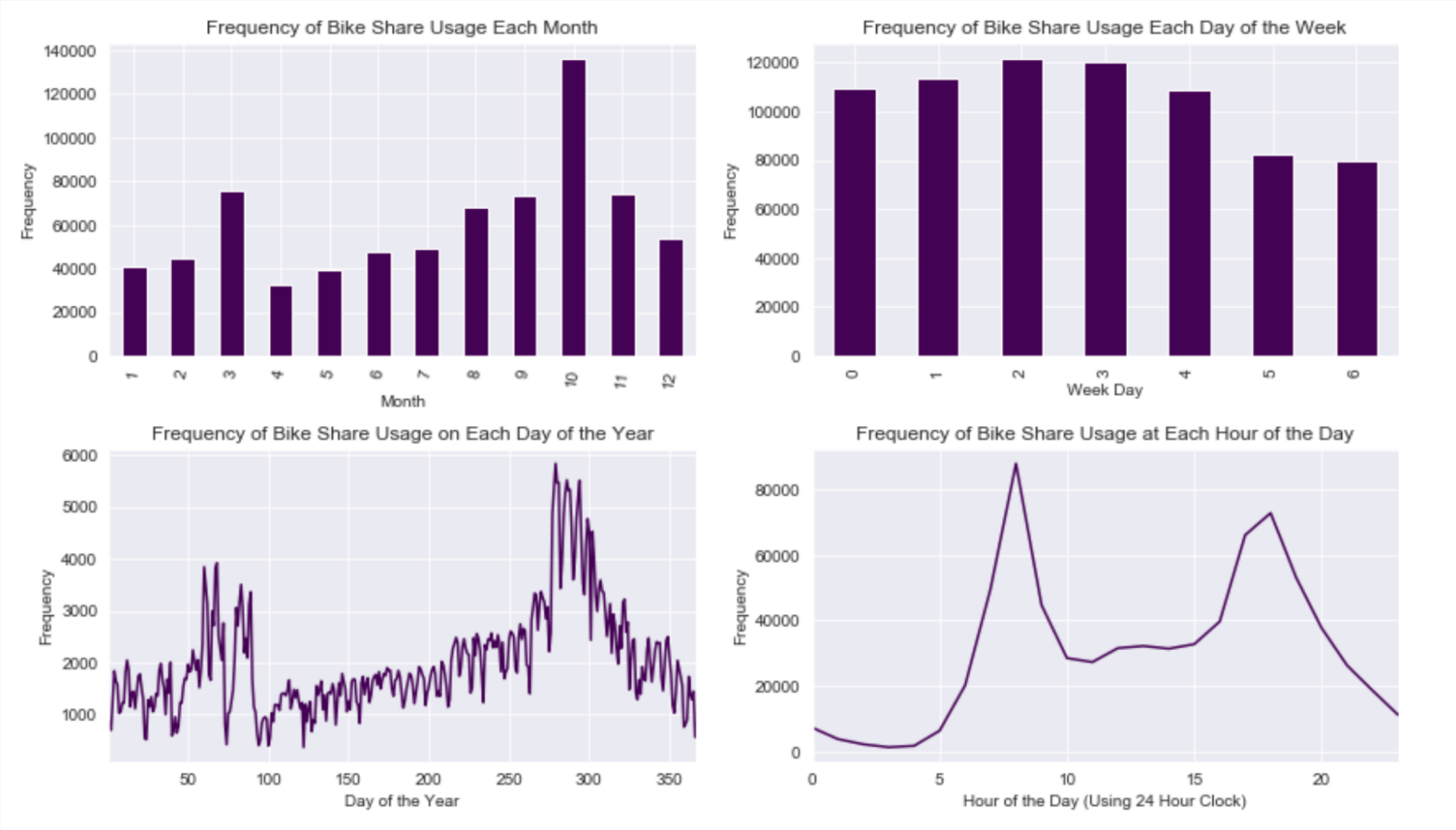
	start_station_id	start_station_name	start_station_latitude	start_station_longitude	end_station_id	end_station_name	end_station_latitude	end_station_longitude	trip_duration	start_time	stop_time	bike_id	user_type	birth_year	gender
0	3212	Christ Hospital	40.735	-74.050	3207	Oakland Ave	40.738	-74.052	376	2015-10-01 00:16:26	2015-10-01 00:22:42	3212	Subscriber	1960.000	1
1	3207	Oakland Ave	40.738	-74.052	3212	Christ Hospital	40.735	-74.050	739	2015-10-01 00:27:12	2015-10-01 00:39:32	3207	Subscriber	1960.000	1
2	3193	Lincoln Park	40.725	-74.078	3193	Lincoln Park	40.725	-74.078	2714	2015-10-01 00:32:46	2015-10-01 01:18:01	3193	Subscriber	1983.000	1
3	3199	Newport Pkwy	40.729	-74.032	3187	Warren St	40.721	-74.038	275	2015-10-01 00:34:31	2015-10-01 00:39:06	3199	Subscriber	1975.000	1
4	3183	Exchange Place	40.716	-74.033	3192	Liberty Light Rail	40.711	-74.056	561	2015-10-01 00:40:12	2015-10-01 00:49:33	3183	Customer	1984.000	0
5	3198	Heights Elevator	40.749	-74.040	3215	Central Ave	40.747	-74.049	365	2015-10-01 00:41:46	2015-10-01 00:47:51	3198	Customer	1984.000	0
6	3206	Hilltop	40.731	-74.058	3195	Sip Ave	40.731	-74.064	139	2015-10-01 00:43:44	2015-10-01 00:46:03	3206	Subscriber	1988.000	1

USAGE TRENDS





USAGE TRENDS



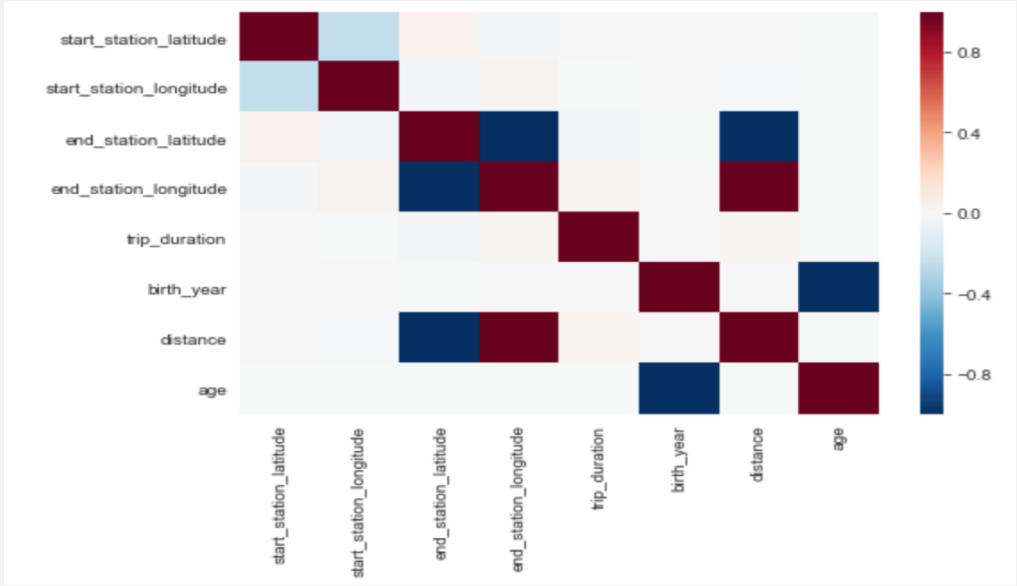
PREPARING THE DATA FOR MODELING

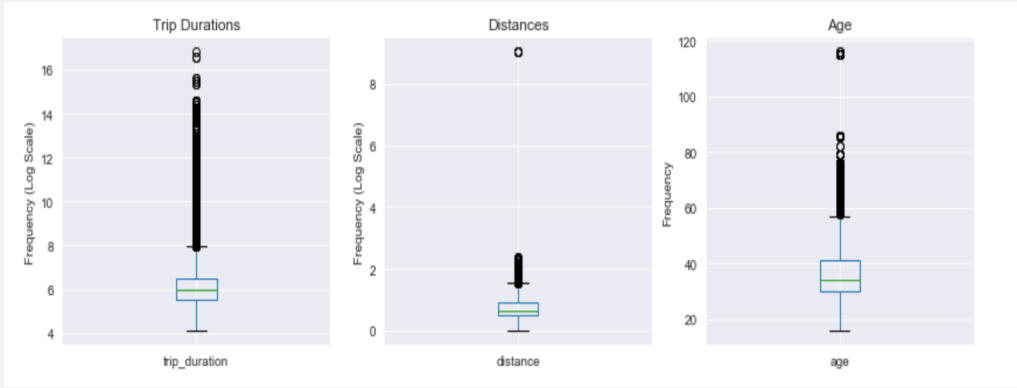
Class Balancing

Leaving Out Features that Hurt Modeling

Imputing Extreme Values

PCA and SelectKbest Transformations





MODELING PROCESS

- The 7 types of models used in this study are: Naïve Bayes, K Nearest Neighbors, Decision Tree, Random Forest, Logistic Regression, Support Vector Classifier, and Gradient Boost.
- Each model type was run using all of the dataset's useful features, PCA components, variables chosen by selectKbest.
- The models were evaluated using Cross Validation, AUC, and Confusion Matrices.

NAÏVE BAYES

- The best performing naïve bayes model used the PCA components.
- Even the best naïve bayes model performed slightly better than chance.
- The assumption on independence was violated in naïve bayes that didn't use PCA.

```
accuracy score:
0.5502142857142858

cross validation:
[0.55914286 0.54507143 0.54642857 0.54678571 0.5525    ]

cross validation with AUC:
[0.57789219 0.56450698 0.5667154  0.56039858 0.57511073]

confusion matrix:
[[19050 15935]
 [15550 19465]]

              precision    recall  f1-score   support

           1       0.55      0.54      0.55     34985
           2       0.55      0.56      0.55     35015

   micro avg       0.55      0.55      0.55     70000
   macro avg       0.55      0.55      0.55     70000
weighted avg       0.55      0.55      0.55     70000

CPU times: user 2.7 s, sys: 526 ms, total: 3.23 s
Wall time: 930 ms
```

K NEAREST NEIGHBORS

- The best performing KNN model used the PCA components.
- The KNN models were relatively low performing.
- The lack of unique trends within the different outcome variables likely resulted in meshes that weren't useful for predicting.

```
accuracy score:
0.7042571428571428

cross validation:
[0.61992857 0.60428571 0.61314286 0.61135714 0.61735714]

cross validation with AUC:
[0.66272518 0.65338118 0.6568706  0.65719946 0.6609031 ]

confusion matrix:
[[24259 10726]
 [ 9976 25039]]

              precision    recall  f1-score   support

           1       0.71      0.69      0.70     34985
           2       0.70      0.72      0.71     35015

   micro avg       0.70      0.70      0.70     70000
   macro avg       0.70      0.70      0.70     70000
weighted avg       0.70      0.70      0.70     70000

CPU times: user 2min 14s, sys: 743 ms, total: 2min 15s
Wall time: 2min 15s
```

DECISION TREE

- The best performing decision tree model used the selectKbest features.
- The random forest model using all of the dataset's useful features was the second best model in the study.
- Being able to make decisions based through the process of binary splits aided in model's success.

```
accuracy score:
0.9051857142857143

cross validation:
[0.80079994 0.8015856  0.79871429 0.79591399 0.78869919]

cross validation with AUC:
[0.80169748 0.80471628 0.80455413 0.80902564 0.81185549]

confusion matrix:
[[30260  4583]
 [ 2054 33103]]

              precision    recall  f1-score   support

           1       0.94      0.87      0.90     34843
           2       0.88      0.94      0.91     35157

   micro avg       0.91      0.91      0.91     70000
   macro avg       0.91      0.91      0.91     70000
weighted avg       0.91      0.91      0.91     70000

CPU times: user 2.37 s, sys: 58 ms, total: 2.43 s
Wall time: 2.45 s
```

RANDOM FOREST

- The best performing random forest model used all of the dataset's useful features.
- The random forest model using all of the dataset's useful features was the best model in the study.
- This model builds on the strengths of the decision tree and reduces the risk of overfitting, resulting in higher CV scores.

```
accuracy score:
0.9426571428571429

cross validation:
[0.811942   0.82264286 0.82185714 0.81464286 0.8173441 ]

cross validation with AUC:
[0.90020833 0.90318671 0.89839968 0.89670665 0.90036234]

confusion matrix:
[[32719  2417]
 [ 1597 33267]]

              precision    recall  f1-score   support

           1       0.95      0.93      0.94     35136
           2       0.93      0.95      0.94     34864

   micro avg       0.94      0.94      0.94     70000
   macro avg       0.94      0.94      0.94     70000
weighted avg       0.94      0.94      0.94     70000

CPU times: user 31min 36s, sys: 41.4 s, total: 32min 18s
Wall time: 32min 25s
```

LOGISTIC REGRESSION

- The best performing logistic regression model used all of the dataset's useful features.
- The logistic regression model was the second worst performing model (slightly better than naïve bayes).
- The model's L1 regularization, effectively acted as another form of feature selection.

```
accuracy score:
0.5649

cross validation:
[0.56731662 0.55857143 0.56678571 0.56071429 0.56525466]

cross validation with AUC:
[0.58824061 0.57685143 0.58724502 0.58079623 0.5846629 ]

confusion matrix:
[[19120 16016]
 [14441 20423]]

              precision    recall  f1-score   support

           1       0.57      0.54      0.56     35136
           2       0.56      0.59      0.57     34864

   micro avg       0.56      0.56      0.56     70000
   macro avg       0.57      0.56      0.56     70000
weighted avg       0.57      0.56      0.56     70000

CPU times: user 9min 47s, sys: 3.05 s, total: 9min 51s
Wall time: 9min 51s
```

SUPPORT VECTOR

- The best performing support vector classifier used all of the dataset's useful features.
- The support vector model had middling performance but the max_iter keyword was used to preserve computational resources.
- An unrestricted version of the model run with a much smaller sample yielded higher accuracy scores.

```
accuracy score:
0.8121571428571429

cross validation:
[0.64852511 0.6395     0.649      0.64442857 0.64261733]

cross validation with AUC:
[0.70929007 0.7017278  0.71106317 0.7048801  0.706588  ]

confusion matrix:
[[27628  7508]
 [ 5641 29223]]

              precision    recall  f1-score   support

           1       0.83      0.79      0.81     35136
           2       0.80      0.84      0.82     34864

   micro avg       0.81      0.81      0.81     70000
   macro avg       0.81      0.81      0.81     70000
weighted avg       0.81      0.81      0.81     70000

CPU times: user 1h 26min 59s, sys: 23 s, total: 1h 27min 22s
Wall time: 1h 27min 46s
```

GRADIENT BOOST

- The best performing gradient boost used all of the dataset's useful features.
- The gradient boost model had middling performance.
- This model type uses decision trees to reduce error, however this method also leaves fewer options for parameter tuning than random forest.

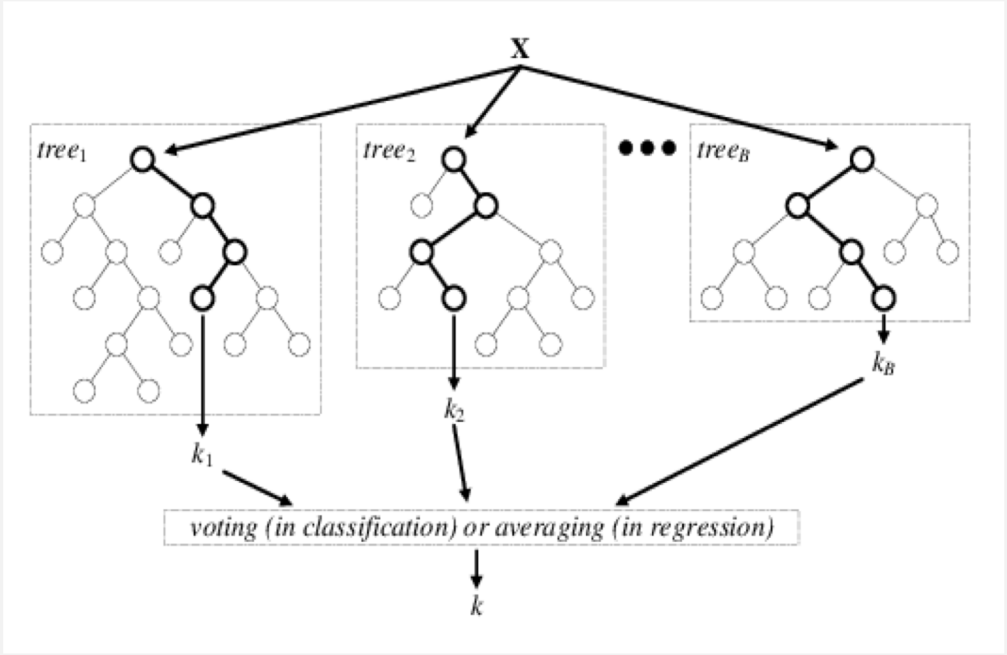
```
accuracy score:
0.8123285714285714

cross validation:
[0.79380044 0.79021429 0.79392857 0.79428571 0.79562826]

cross validation with AUC:
[0.8757069  0.87151076 0.87727955 0.87346044 0.87618924]

confusion matrix:
[[27230  7906]
 [ 5231 29633]]

              precision    recall  f1-score   support

           1       0.84      0.77      0.81     35136
           2       0.79      0.85      0.82     34864

   micro avg       0.81      0.81      0.81     70000
   macro avg       0.81      0.81      0.81     70000
weighted avg       0.81      0.81      0.81     70000

CPU times: user 2h 1min 39s, sys: 1min 38s, total: 2h 3min 17s
Wall time: 2h 4min 25s
```

MODELING OVERVIEW

- The random forest was the best performing model type. The support vector classifier has potential to yield very high accuracy scores as well.
- More complex model types that relied on all of the dataset's useful features generally performed better.
- The fact that full featured models generally performed better implies that model types using selectKbest may have removed well performing features in the process.

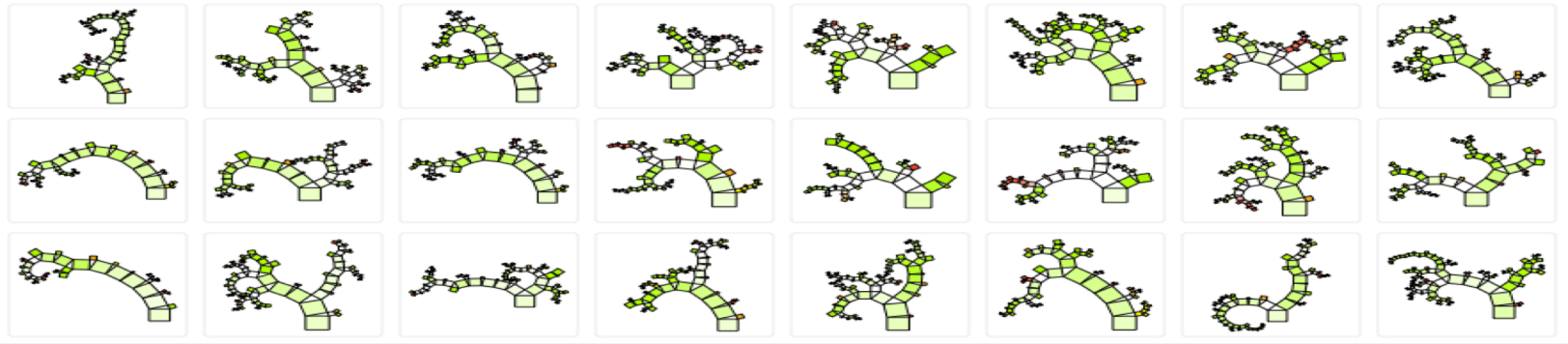
THE RANDOM FOREST

- The random forest model that used all of the dataset's useful features was the most accurate when it came to predicting gender.
- It had slightly better accuracy scores than the random forest that used selectKbest.
- The random forest was also capable of handling larger amounts of training data than the gradient boost, resulting in faster run times.



THE RANDOM FOREST

- The random forest tends to favor using features with larger numbers of classes in the decision making process, making it more difficult to discern the actual importance of individual features.
- Despite the 'black box' nature of the random forest, the model yielded the best results when it came to the bike share data through tuning of parameters and the limited range of the datapoints.



CONCLUSIONS OF THE STUDY

- This study established the best supervised modeling technique and feature classification pairing for the gender of the Citi bike users.
- Understanding how to better utilize supervised modeling techniques to predict gender, will give insight as to what kind of people are using the bike share service and particular habits different types of customers share.

FUTURE DIRECTION

- The next step in using this data to discern the demographics of the users based on their usage of the service would be to collect more types of data and go more in depth into which features have a greater impact on the likelihood of a user being a particular gender.
- Afterwards the study can be expanded to include different types of demographical classes as outcomes such as age.

THANK YOU!