Predicting Profit of Bike-Sharing Services

A Comparative Analysis of Machine Learning Algorithms

Introduction

In this report, we will attempt to address what variables are most important in predicting profit for a bike-sharing service. We want to compute a predictive model that will help the company predict profit or loss on a given day. The most important features in predicting profit through the various means of analysis were:

- `weekend`: Whether a day is on the weekend or not
- `humidity`: Humidity on a given day
- `feels_like`: The temperature it feels like on a given day
- `temperature`: Temperature on a given day
- `wind_speed`: Wind speed on a given day

The variables not listed above, but were a part of our original dataset, include `date`, the continuous variable `N_bikes` (the number of bikes used on a given day), and the categorical variables `holiday` and `season`. We use some combination of these predictors to address the goals of the analysis. Some methods placed more importance on certain variables than others, which is addressed in the **Results** section.

Methods

To construct the criteria for whether a day was profitable or not, we used the variable `N_bikes`. If `N_bikes` was greater than $20,000$, then the day is considered profitable, and if it was less, then it is not profitable. We examined the dataset and identified that the continuous variables `N_bikes` and `humidity` and the categorical variable `holiday` had missing values. This posed an interesting problem, as we were using `N_bikes` to construct our response variable `Profit`. To address the missing values, we imputed the missing variables two different ways: mean imputation and iterative regression imputation.

For mean imputation, we began by imputing the continuous variable `humidity` with its mean. Since we have missing values in `N_bikes`, we cannot construct the response variable. Imputing this with the mean and then constructing the response variable would be misleading, as the mean turned out to be greater than $20,000$ and there would be more profitable days reported than there truly are. So to handle this, we remove all missing values for `N_bikes`. We handle the missing categorical values for `holiday` by removing them from the dataset as well.

For iterative regression imputation, we began by imputing `N_bikes` and `humidity` with simple random imputation. We then impute `N_bikes` and `humidity` by constructing regression functions

using all other available variables in the dataset and the imputed versions of `N_bikes` and `humidity`. To deal with the missing categorical values in `holiday`, we created a new variable `holiday2` for which we replaced all the missing values with a category "missing", and proceeded with our analysis using `holiday2`.

From here on, the analysis and building of predictive models follow the same general process for both datasets. We split the datasets into a training and test set using the validation set method with a 70% training and 30% test set. Then, we conducted our analysis using three methods: logistic regression, K-nearest neighbors, and decision tree classification with random forests.

Logistic regression was used since it predicts a binary variable given other independent covariates with a logit link function on our linear predictor. We also have strong assumptions for this method, such as no multicollinearity, no outliers, and independence in errors.

When tasked with choosing between K-nearest neighbors (KNN), linear discriminant analysis (LDA), or quadratic discriminant analysis (QDA), we chose KNN. This is because of key assumptions that must be met when using LDA and QDA. Both of those methods require assumption of jointly Normal predictors, and since we include categorical predictors in our analysis, the assumption is violated, leaving KNN as the only reasonable choice. KNN is a nonparametric classifier that predicts based on the closest K points to the observation we want to predict. We chose the number of neighbors K with cross validation.

Decision trees are a nonparametric regression and classification method that creates regions corresponding to specific values of the predictors, then assigns the maximum class probability of the response, `Profit`. We also implement a random forest algorithm, which constructs B bootstrapped training datasets and uses $m = \sqrt{p}$ predictors (where p is the total number of predictors), which aids in providing decorrelated trees and reduces the variance of our prediction.

For each method, we first constructed our model on the training set. We then predicted our model on the test set and classified our posterior probabilities using Bayes rule, and then calculated our test error rate by comparing our predicted classifications to our test set observations. We also constructed ROC curves to show how sensitivity and specificity vary for different classification thresholds and calculated the area under the curve (AUC).

Results

In this section, we will address the methodology and relevant outputs for logistic regression, K-nearest neighbors, and the tree-based methods for both the mean imputed data and the iterative regression imputed data separately.

Mean imputation data

Logistic Regression

For logistic regression, we first built a model with all available predictors, then used best subset selection to determine which were most important. For mean imputation, the model obtained with

best subset selection is below. $\hat{\pi}_i$ is the probability of a day being profitable.

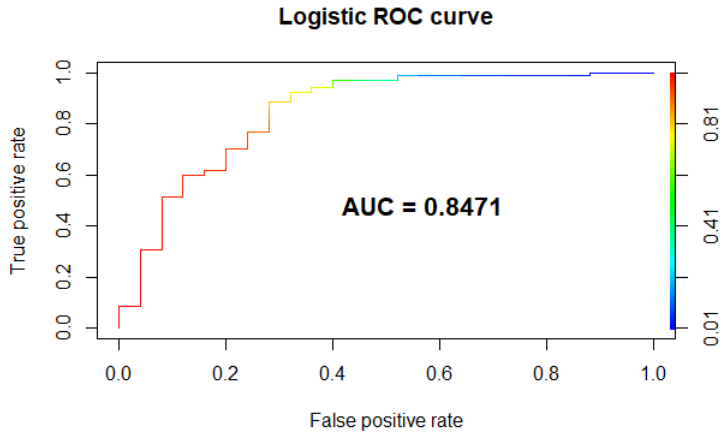
$$\log(\frac{\hat{\pi}_i}{1-\hat{\pi}_i}) = 19.096 + 0.585 * \texttt{temperature}_i - 0.165 * \texttt{feels\_like}_i - 0.225 * \texttt{humidity}_i$$
$$- 0.199 * \texttt{wind\_speed}_i - 3.568 * \texttt{holidayYes}_i - 4.371 * \texttt{weekendYes}_i + \epsilon$$

After building this model, we conducted a deviance test on it to see if it performed as good as the saturated model with all predictors. Our hypothesis test is as follows:

$$H_0 : l_{\text{saturated}} = l_{\text{fitted}}$$
$$H_a : l_{\text{saturated}} \neq l_{\text{fitted}}$$

We obtained a very high p-value of 1, which may be a result of overfitting the data. A p-value greater than $\alpha = 0.05$ tells us that we fail to reject the null hypothesis H_0, indicating that our fitted model is as good as the saturated model. As peculiar as the high p-value is, we have no better alternative to fit the data, so we continue our analysis with the model above.

We then used the model to predict on our test dataset. We obtained our posterior probabilities $\hat{\pi}_i$, and implemented Bayes classification rule for each observation i in the test set. That is, for $\hat{\pi}_i > 0.5$, the day is classified as profitable (i.e. `Profit = 1`), otherwise it is not profitable (i.e. `Profit = 0`). With this model, we obtained a test error rate of 0.098, which is an indication of a fairly good classifier. We construct the ROC curve shown below.



The ROC curve does not approach a true positive rate of 1 very quickly. This is reflected in the obtained value for the AUC as well.

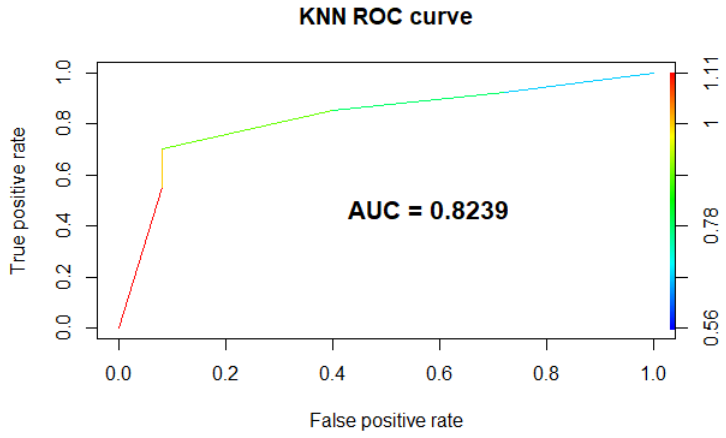
For this method, we obtained a fairly low test error rate for our data, which is desirable. However, we also face a low AUC and a slow progression to a high true positive rate for our ROC curve. This indicates that we are more likely to report that a day will be profitable when in fact it will not be. We will now assess whether other methods perform better in this regard.

K-nearest Neighbors

For this section, we began by choosing the number of neighbors K to consider for each observation we want to predict. We choose this K via cross-validation with 10 folds, and it was determined that $K = 9$ was the best choice to retain high accuracy of prediction. For deeper insight into the analysis, we also ran the KNN analysis for various K values surrounding $K = 9$ to see how it affected error rates and flexibility. The results are in the table below.

K	Test ER
3	0.1742
5	0.1742
7	0.1818
9	**0.1667**
11	0.1667
13	0.1742

As K increases, flexibility decreases, so bias increases and would lead to higher error in our prediction. Since we have predicted on our test data, we now assess the sensitivity and specificity via the ROC curve, shown below.



The ROC curve does not approach a true positive rate of 1 very quickly. This is reflected in the obtained value for the AUC as well.

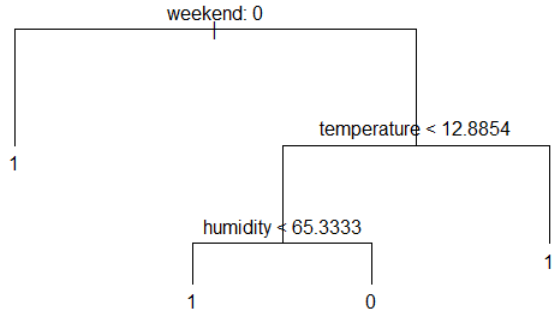
For this method, we obtained a relatively high test error rate for our data compared to logistic regression above and random forest below. Additionally, we face a low AUC and a slow progression to a high true positive rate for our ROC curve. This indicates that we are more likely to report that a day will be profitable when in fact it will not be. KNN does not perform very well with this data at all compared to the other methods. This may be since the model is nonparametric, there are no assumptions on linearity. Considering how much worse KNN performs compared to logistic regression, it may be reasonable to believe that the decision boundary is linear, which would not be accounted for in KNN classification.

Classification Tree / Random Forest

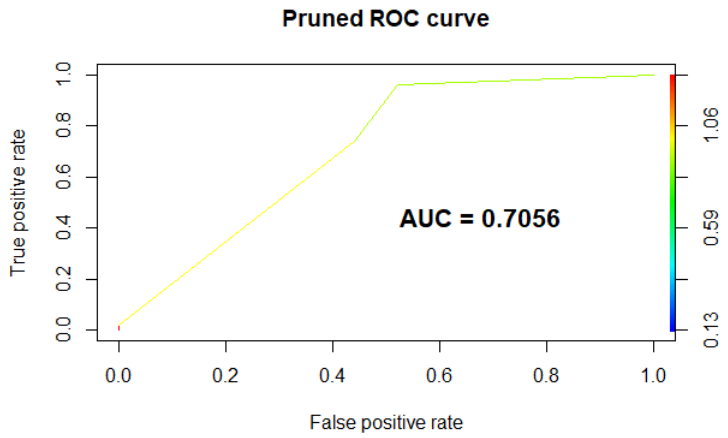
For this section, we first construct a classification tree with all available predictors, then prune it back and select the subtree that leads to the largest reduction in the Gini index. We then calculate the test error rate of the unpruned tree and compare it to the pruned tree. The results are in the table below.

Tree type	Test ER
Unpruned	0.1667
Pruned	0.1288

As we would expect, the pruned tree consists of better predictive performance. This is because we are not at risk of overfitting the data with too many predictors as we did with the unpruned tree. The pruned tree is displayed graphically below.

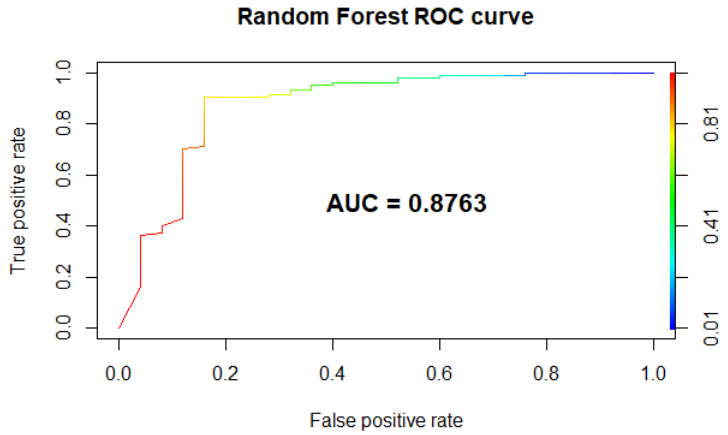


The pruned tree is only dependent on three predictors. Through cross-validation, it was determined that this tree led to the largest reduction of the Gini index and is the optimal tree for classification. We observe the AUC of this classification tree below.



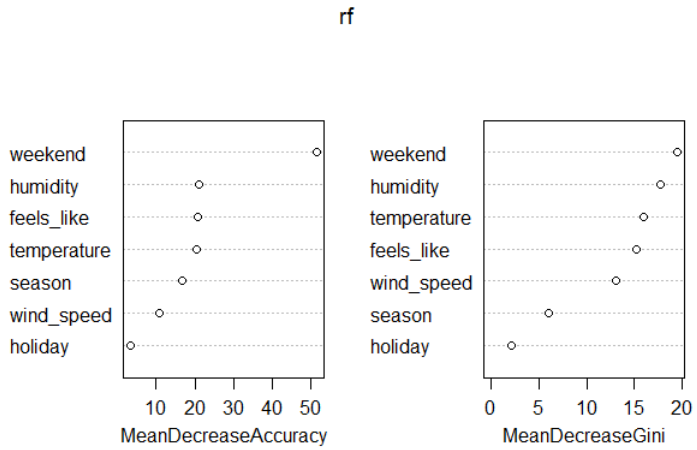
Similar to the previous methods, we suffer from a low AUC and a slow approach to a true positive rate of 1, which makes classification unreliable. In this case, the AUC is even lower than that of logistic regression and KNN.

Thankfully, there is a better way to use classification trees for prediction, and that is with the implementation of a random forest algorithm. With random forest, we construct $B = 500$ bootstrapped trees with a subset $m = 3$ of our predictors p. This will help lower variance in our classification and improve predictive power. Using a random forest approach should lead to a reduction in the test error rate and a higher AUC. For the random forest implementation, we obtained a test error rate of 0.1212. The ROC curve and AUC are below.



From above, we can see that the AUC for random forest is considerably higher than the AUC for the pruned tree, and our error rate is much lower. This is what we expected to happen with this implementation for the reasons detailed above.

Another advantage of bootstrapping over many trees is that we can construct a plot of variable importance to show which predictors are the most important in modeling. This is measured by the decrease of the Gini index induced by each split over a given predictor, and is averaged over the B trees. The variable importance plot is given below.



The left hand plot details the variables that contribute the most to a lower error rate. The right hand plot details the variables that are most important in constructing the tree. For the most part the same variables are important in both, specifically `weekend`, `humidity`, `temperature` and `feels_like`. Knowing this is helpful and gives an at-a-glance look at which variables contribute most to predicting `Profit` successfully.

Before moving on to iterative regression, our summary of the mean imputed methods are below.

(n=438)	Logistic Regression	KNN (k=9)	Unpruned Tree	Pruned Tree	Random Forest
Test ER	0.098	0.1667	0.1667	0.1288	0.1212
AUC	0.8471	0.8239	N/A	0.7056	0.8763

Iterative regression imputation data

Logistic Regression

Similar to before, we first built a model with all available predictors, then used best subset selection to determine which were most important. The model obtained with best subset selection is below with our iteratively imputed dataset. $\hat{\pi}_i$ is the probability of a day being profitable.

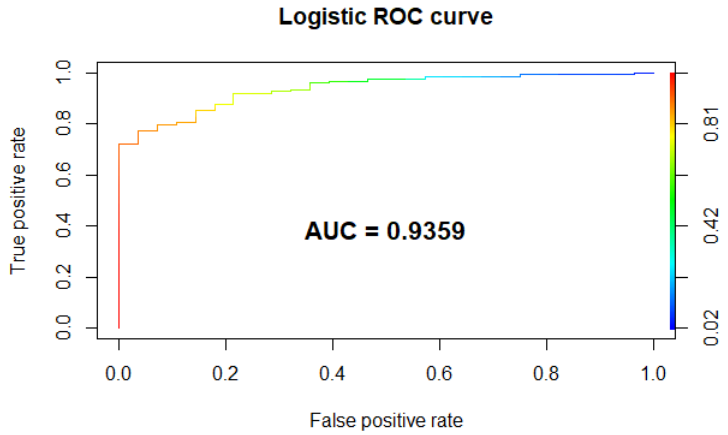
$$\log(\frac{\hat{\pi}_i}{1-\hat{\pi}_i}) = 16.085 + 0.217 * \texttt{feels\_like}_i - 0.177 * \texttt{humidity}_i - 0.136 * \texttt{wind\_speed}_i - 2.986 * \texttt{weekendYes}_i + \epsilon$$

The main difference between this and mean imputation is that we now did not include `temperature` or `holiday`. After building this model, we conducted a deviance test on it to see if it performed as good as the saturated model with all predictors. Our hypothesis test is as follows:

$$H_0 : l_{\text{saturated}} = l_{\text{fitted}}$$
$$H_a : l_{\text{saturated}} \neq l_{\text{fitted}}$$

We obtained a very high p-value of 1, which may be a result of overfitting the data. A p-value greater than $\alpha = 0.05$ tells us that we fail to reject the null hypothesis H_0, indicating that our fitted model is as good as the saturated model. As peculiar as the high p-value is, we have no better alternative to fit the data, so we continue our analysis with the model above.

We then used the model to predict on our test dataset. We obtained our posterior probabilities $\hat{\pi}_i$, and implemented Bayes classification rule for each observation i in the test set to determine if a day is profitable. With this model, we obtained a test error rate of 0.1, which is an indication of a fairly good classifier, although similar to the test error rate we obtained for mean imputation. We construct the ROC curve shown below.



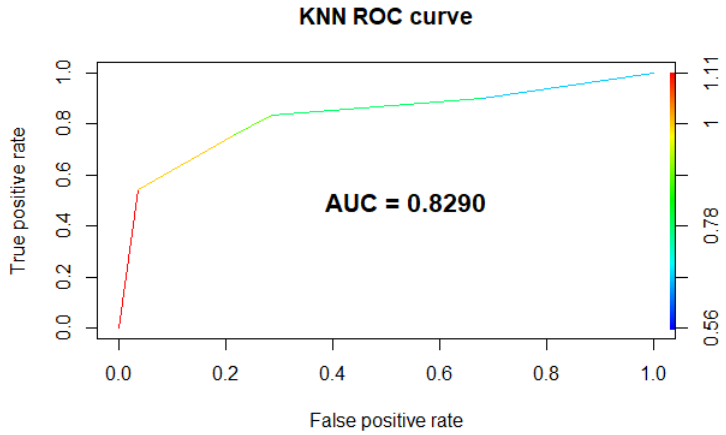
The key difference in this analysis is that the ROC curve approaches a true positive rate of 1 very quickly. We obtained a fairly low test error rate for our data and a high AUC. With the iteratively imputed dataset, we are able to predict with more true positives and a similarly low error rate.

K-nearest neighbors

We repeated the same process for this dataset as we did for the mean imputed dataset. We began by choosing the number of neighbors K to consider for each observation we want to predict. We choose this K via cross-validation with 10 folds, and it was determined that $K = 9$ was the best choice to retain high accuracy of prediction. For deeper insight into the analysis, we also ran the KNN analysis for various K values surrounding $K = 9$ to see how it affected error rates and flexibility. The results are in the table below.

K	Test ER
3	0.1467
5	0.1467
7	0.1267
9	**0.1267**
11	0.14
13	0.1333

As K decreases, flexibility increases. So even though we have the same test error rate when $K = 7$, we do not want to make the model too flexible, as this leads to increased variance in our prediction. Additionally, the error rates above were determined via the validation set method. Even though cross-validation chose an optimal $K = 9$, that may not be directly reflected above since the validation set depends on which observations were randomly allocated to the test and training sets, whereas cross-validation mitigates this issue by definition. Now that we have predicted on our test data, we assess the sensitivity and specificity via the ROC curve, shown below.



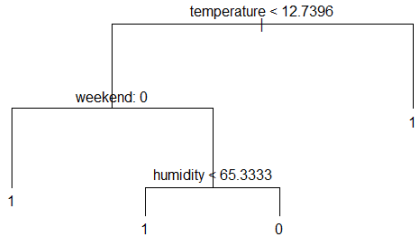
Compared to the results from the mean imputed version of the data, KNN performs better with this dataset. However, it still has a relatively high test error rate and a lower AUC than the other methods for the iteratively imputed data. It is still the worst of our three methods by both measures. This reinforces our idea mentioned from above, where we indicated that it may be reasonable to believe that the decision boundary for predicting `Profit` is linear, and thus the absence of the linearity assumption in KNN is harmful for analysis on this data.

Classification Tree / Random Forest

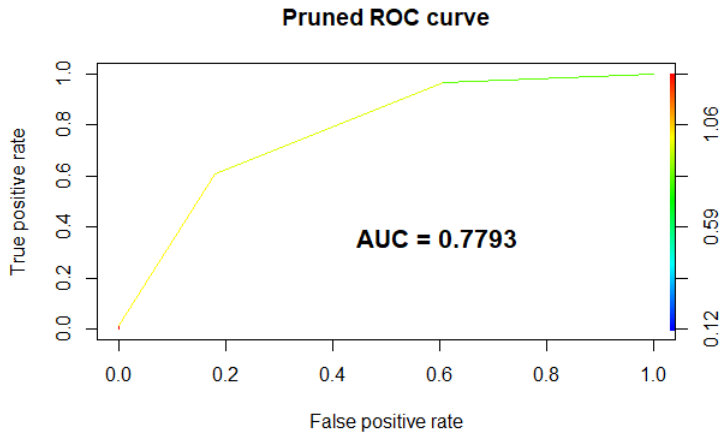
Again, we construct a classification tree with all available predictors, then prune it back and select the subtree that leads to the largest reduction in the Gini index. We then calculate the test error rate of the unpruned tree and compare it to the pruned tree. The results are in the table below.

Tree type	Test ER
Unpruned	0.16
Pruned	0.14

As we would expect, the pruned tree consists of better predictive performance. This is because we are not at risk of overfitting the data with too many predictors as we did with the unpruned tree. The pruned tree is displayed graphically below.

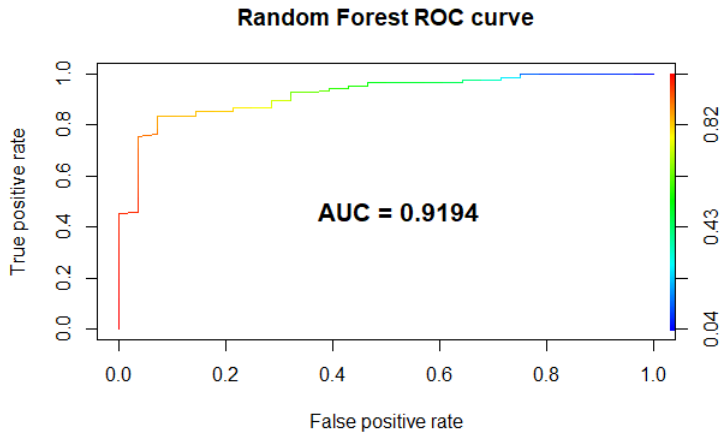


The pruned tree looks very similar to the one constructed for mean imputation and consists of the same predictors. The key difference, again, shows in our ROC curve which is shown below.

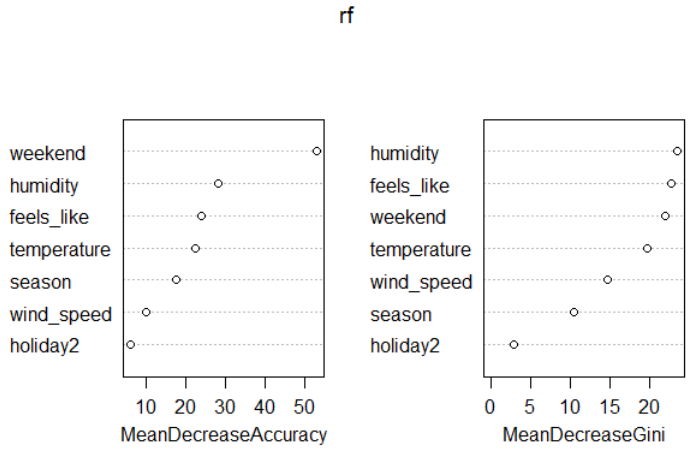


Compared to the mean imputation AUC, this is considerably higher but still low compared to other methods. We now consider the random forest algorithm.

Same as before, we construct $B = 500$ bootstrapped trees with a subset $m = 3$ of our predictors p. Using a random forest approach should lead to a reduction in the test error rate and a higher AUC relative to our pruned tree. For the random forest implementation, we obtained a test error rate of 0.1133. The ROC curve and AUC are below.



From above, we can see that the AUC for random forest is considerably higher than the AUC for the pruned tree, and our error rate is much lower than our pruned tree. Additionally, our error rate is lower and AUC is higher in this random forest implementation than for the mean imputation. To see which predictors were most important, we observe the variable importance plot below.



The left hand plot details the variables that contribute the most to a lower error rate. The right hand plot details the variables that are most important in constructing the tree. The top 4 variables, specifically `weekend`, `humidity`, `temperature` and `feels_like`, are the same in both this analysis and for mean imputation.

Our summary of the iteratively imputed results are below.

(n=500)	Logistic Regression	KNN (k=9)	Unpruned Tree	Pruned Tree	Random Forest
Test ER	0.10	0.1267	0.16	0.14	0.1133
AUC	0.9359	0.8290	N/A	0.7793	0.9194

Comparison of imputation methods

Method	Mean Imputation (n=438)		Iterative Imputation (n=500)	
	Test Error Rate	AUC	Test Error Rate	AUC
Logistic Regression	0.098	0.8471	0.10	0.9359
KNN (k=9)	0.1667	0.8239	0.1267	0.8290
Unpruned Tree	0.1667	N/A	0.16	N/A
Pruned Tree	0.1288	0.7056	0.14	0.7793
Random Forest	0.1212	0.8763	0.1133	0.9194

The table above shows the difference in test error rates and AUC for each method for the mean imputed data and the iteratively imputed data. We notice similar or lower test error rates for each method, but one notable change is the increase in the AUC for the respective ROC curves. A higher AUC means that we are more likely to classify profitable days as profitable and not profitable days as not profitable. In other words, our predictive performance is better with the dataset with iteratively imputed values.

Discussion

When conducting this analysis, the main goal was to build a model with low error rates to provide good predictive performance for `Profit`. We used three methods: logistic regression, KNN, and decision trees with random forest. Overall, our results show that logistic regression and random forest perform best with this data, both with mean imputation and iterative regression imputation. We obtain similarly low test error rates for these methods on both datasets, but the AUC is much higher for the iteratively imputed dataset, increasing our classification performance. Thus, our best method for future analysis are to iteratively impute the data and use a linear classifier like logistic regression or random forest.

The next goal was determining which variables are most important in predicting `Profit`. Through best subset selection for logistic regression and constructing a variable importance plot for random forest, we determined that our most important predictor was `weekend`, a binary variable that indicates whether a given day is on the weekend or not. This is indicative of the fact that linear classifiers would perform better than nonparametric methods, which is reflected in our results for logistic regression. We also observed that this is one of the important splits that leads to a large reduction in the test error rate and Gini index in our random forest analysis. This also explains why KNN performed poorly under both datasets relative to the other methods. We also observe some important continuous variables, like `humidity`, `feels_like`, and `temperature`. This is a straightforward indication that the weather plays a large role in determining whether people are willing to use the bike-sharing service and whether a day will be profitable or not.

Future analysis that can be conducted to further increase predictive performance could be to consider different classification methods, like the support vector classifier, which is a different linear classifier. We would expect it to perform similarly to logistic regression, but it may lead to better predictive performance if the classes are well separated. Another suggestion is to improve the iterative imputation method. We could be more selective with the regressors used to impute our continuous variables, rather than using all available predictors. This could lead to better estimates of our missing values and would help predict the correct classification.

Overall, our results show that logistic regression and random forest perform best in predictive performance. We determined the most important predictors using the results of these methods, and were able to adequately predict `Profit` with a relatively low error rate.