# Predicting Demand

## Contents

# Introduction & Data

Bike-sharing systems are appearing all over the world; examples include Citi Bike in New York City, Santander Cycles in London, and ofo in China. These services allow users to make short-term bike rentals. In docked systems, docking stations are set up in prespecified locations, and users must pick up and return the bike to a docking station within the system. In dockless systems, users are able to pick up and return bikes to any desired location (pickups pending availability). There is a lot of research in Bike-sharing systems and as a start, in this problem, we will attempt to understand the factors that influence a high demand for this service.

Dataset: bikes.csv

In the dataset above, each observation represents one hour of the day (10886 hours). Here is a detailed description of the variables:

- ***season***: 1 = spring, 2 = summer, 3 = fall, 4 = winter
- ***holiday***: whether the day is considered a holiday
- ***workingday***: whether the day is neither a weekend nor holiday
- ***weather***:
  - ***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
- ***temp***: temperature in Celsius
- ***atemp***: "feels like" temperature in Celsius
- ***humidity***: relative humidity
- ***windspeed***: wind speed
- ***count***: number of total rentals
- ***demand_level***: 1 if count is at least 250, 0 otherwise
- ***hour***: the hour of the day (0-23)

In this problem, we will use various classification methods to try to predict the demand level.

# Exercices

## *Problem 1 : Exploratory Data Analysis*

```
## 'data.frame':    10886 obs. of  11 variables:
##  $ season      : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ holiday     : int  0 0 0 0 0 0 0 0 0 0 ...
```

```
## $ workingday  : int  0 0 0 0 0 0 0 0 0 0 ...
## $ weather     : int  1 1 1 1 1 2 1 1 1 1 ...
## $ temp        : num  9.84 9.02 9.02 9.84 9.84 ...
## $ atemp       : num  14.4 13.6 13.6 14.4 14.4 ...
## $ humidity    : int  81 80 80 75 75 75 80 86 75 76 ...
## $ windspeed   : num  0 0 0 0 0 ...
## $ count       : int  16 40 32 13 1 1 2 3 8 14 ...
## $ demand_level: int  0 0 0 0 0 0 0 0 0 0 ...
## $ hour        : int  0 1 2 3 4 5 6 7 8 9 ...
```

#### 1.1

**Which season has the most rentals?**

*Answer* :

1. Summer (2)
2. **Fall (3)**
3. Winter (4)
4. Spring (1)

#### 1.2

**What is the average temerature in Celsius?**

```
## [1] 20.23086
```

*Answer* : 20.23086

#### 1.3

**What is the average temerature in Celsius during the high demand hours?**

High demand is defined by demand_level = 1.

*Answer* : 24.48587

## *Problem 2 : Simple Logistic Regression*

### 2.1 : Preparing the Data

**2.1.1** We will now split the data into a training and testing set. To do this, we use the sample.split() function.
**Which variable will be used in this function?**

1. temp
2. count
3. **demand_level**
4. season

**2.1.2** Set your **random seed to 100** and create a training and test set using the sample.split() function in the caTools library, **with 70%** of the observations in the training set and 30% in the testing set.

**Why do we use the sample.split() function?**

1. It is the most convenient way to randomly split the data
2. It balances the independent variables between the training and testing sets
3. It balances the dependent variable between the training and testing sets

### 2.1.3 How many observation are there in the training set?

```
## [1] 7620
```

## 2.2

Train a logistic regression model using temp as the independent variable.
**What is the coefficient of temp?**

```
##
## Call:
## glm(formula = demand_level ~ temp, family = binomial, data = bikesTrain)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.6313  -0.8673  -0.5588   1.0896   2.5204
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -3.224047   0.090173  -35.75   <2e-16 ***
## temp         0.110214   0.003798   29.02   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 9304.5  on 7619  degrees of freedom
## Residual deviance: 8308.6  on 7618  degrees of freedom
## AIC: 8312.6
##
## Number of Fisher Scoring iterations: 4
```

*Answer*: 0.110214

## 2.3

**2.3.1** Using your logistic regression model, obtain predictions on the test set. Then, using a probability threshold of 0.5, create a confusion matrix for the test set.
**What is the (test) accuracy of your logistic regression model?**

$$Accuracy = \frac{TruePositive + TrueNegative}{Ntotal}$$

```
## [1] 0.7290263
```

*Answer* : 0.7290263

**2.3.2** Our baseline model in classification is to always predict the most frequent outcome in the test set. **What is the (test) accuracy of this baseline model?**

```
## [1] 0.7002449
```

*Answer*: 0.7002449

**2.3.3 What is the true positive rate of your logistic regression model?**

$$Sensitivity = \frac{TruePositive}{TruePositive + FalseNegative}$$

```
## [1] 0.2778345
```

*Answer* : 0.2778345

**Problem 2.3.4 What is the false positive rate of your logistic regression model?**

$$FalsePositiveErrorRate = \frac{FalsePositive}{TrueNegative + FalsePositive}$$

```
## [1] 0.07783122
```

*Answer* : 0.07783122

**2.3.5** Currently, we are predicting many more low demand observations than high demand observations. **Which of the following is a way to change that?**

1. It is impossible to predict more high demand with this model. To change these results, another model can be used.
2. **To predict more high demand, decrease the prediction threshold.**
3. To predict more high demand hours, increase the prediction threshold.
4. To predict more high demand hours, create more observations with high demand.

## *Problem 3 : Adding More Variables*

### 3.1

**3.1.1** We would now like to train a logistic regression model using all of the variables in the training set.

```
##              season holiday workingday weather  temp atemp humidity windspeed
## season         TRUE   FALSE      FALSE   FALSE FALSE FALSE    FALSE     FALSE
## holiday       FALSE    TRUE      FALSE   FALSE FALSE FALSE    FALSE     FALSE
## workingday    FALSE   FALSE       TRUE   FALSE FALSE FALSE    FALSE     FALSE
## weather       FALSE   FALSE      FALSE    TRUE FALSE FALSE    FALSE     FALSE
## temp          FALSE   FALSE      FALSE   FALSE  TRUE  TRUE    FALSE     FALSE
## atemp         FALSE   FALSE      FALSE   FALSE  TRUE  TRUE    FALSE     FALSE
## humidity      FALSE   FALSE      FALSE   FALSE FALSE FALSE     TRUE     FALSE
## windspeed     FALSE   FALSE      FALSE   FALSE FALSE FALSE    FALSE      TRUE
## count         FALSE   FALSE      FALSE   FALSE FALSE FALSE    FALSE     FALSE
## demand_level  FALSE   FALSE      FALSE   FALSE FALSE FALSE    FALSE     FALSE
## hour          FALSE   FALSE      FALSE   FALSE FALSE FALSE    FALSE     FALSE
##              count demand_level  hour
## season       FALSE        FALSE FALSE
## holiday      FALSE        FALSE FALSE
## workingday   FALSE        FALSE FALSE
## weather      FALSE        FALSE FALSE
## temp         FALSE        FALSE FALSE
## atemp        FALSE        FALSE FALSE
## humidity     FALSE        FALSE FALSE
## windspeed    FALSE        FALSE FALSE
## count         TRUE         TRUE FALSE
## demand_level  TRUE         TRUE FALSE
## hour         FALSE        FALSE  TRUE
```

```
##              season holiday workingday weather  temp atemp humidity windspeed
## season        FALSE   FALSE      FALSE   FALSE FALSE FALSE    FALSE     FALSE
## holiday       FALSE   FALSE      FALSE   FALSE FALSE FALSE    FALSE     FALSE
## workingday    FALSE   FALSE      FALSE   FALSE FALSE FALSE    FALSE     FALSE
## weather       FALSE   FALSE      FALSE   FALSE FALSE FALSE    FALSE     FALSE
## temp          FALSE   FALSE      FALSE   FALSE FALSE FALSE    FALSE     FALSE
## atemp         FALSE   FALSE      FALSE   FALSE FALSE FALSE    FALSE     FALSE
## humidity      FALSE   FALSE      FALSE   FALSE FALSE FALSE    FALSE     FALSE
## windspeed     FALSE   FALSE      FALSE   FALSE FALSE FALSE    FALSE     FALSE
## count         FALSE   FALSE      FALSE   FALSE FALSE FALSE    FALSE     FALSE
## demand_level  FALSE   FALSE      FALSE   FALSE FALSE FALSE    FALSE     FALSE
## hour          FALSE   FALSE      FALSE   FALSE FALSE FALSE    FALSE     FALSE
##              count demand_level  hour
## season       FALSE        FALSE FALSE
## holiday      FALSE        FALSE FALSE
## workingday   FALSE        FALSE FALSE
## weather      FALSE        FALSE FALSE
## temp         FALSE        FALSE FALSE
## atemp        FALSE        FALSE FALSE
## humidity     FALSE        FALSE FALSE
## windspeed    FALSE        FALSE FALSE
## count        FALSE        FALSE FALSE
## demand_level FALSE        FALSE FALSE
## hour         FALSE        FALSE FALSE
```

**Which of the following is true?**

1. Weather and temp are highly correlated.

2. Season and weather are highly correlated.
3. **Workingday and holiday are not highly correlated.**
4. **Temp and atemp are highly correlated.**

**3.1.2** Train a logistic regression model now using all of the following variables in the training set: *season*, *holiday*, *workingday*, *weather*, *temp*, *humidity*, *windspeed*, and *hour*

**Which of the following variables are significant at a level of 0.001 or less?**

```
##
## Call:
## glm(formula = demand_level ~ season + holiday + workingday +
##     weather + temp + humidity + windspeed + hour, family = binomial,
##     data = bikesTrain)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.1282  -0.7468  -0.4471   0.8365   2.8967
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -2.895413   0.184754 -15.672  < 2e-16 ***
## season       0.342095   0.029190  11.719  < 2e-16 ***
## holiday     -0.016277   0.169255  -0.096  0.92339
## workingday  -0.175048   0.063433  -2.760  0.00579 **
## weather     -0.026242   0.052297  -0.502  0.61582
## temp         0.097964   0.004078  24.025  < 2e-16 ***
## humidity    -0.029654   0.001863 -15.915  < 2e-16 ***
## windspeed    0.003451   0.003748   0.921  0.35706
## hour         0.074677   0.004497  16.604  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 9304.5  on 7619  degrees of freedom
## Residual deviance: 7389.5  on 7611  degrees of freedom
## AIC: 7407.5
##
## Number of Fisher Scoring iterations: 5
```

1. **season**
2. holiday
3. workingday
4. weather
5. **temp**
6. **humidity**
7. windspeed
8. **hour**

### 3.2

Using your new logistic regression model, obtain predictions on the test set. Then, using a probability threshold of 0.5, create a confusion matrix for the test set.

#### 3.2.1 What is the (test) accuracy of your logistic regression model?

$$Accuracy = \frac{TruePositive + TrueNegative}{Ntotal}$$

```
## [1] 0.7672994
```

$Answer$ : 0.7672994

#### 3.2.2 Which of the following is true?

```
## [1] 512
```

```
## [1] 0.2327006
```

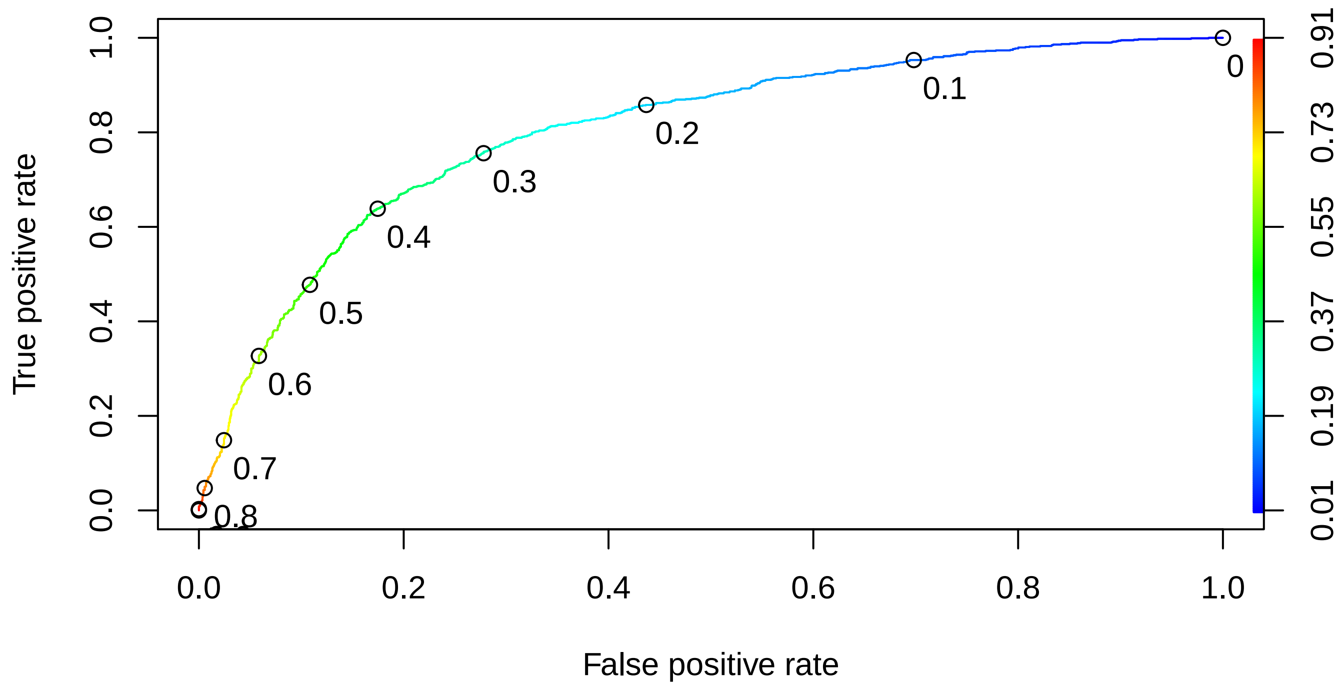
```
## [1] 0.4770174
```

```
## [1] 0.5229826
```

```
## [1] 0.108439
```

1. Close to a third of time that there is high demand, the model will predict high demand.
2. **Almost half of the times that there is high demand, the model will predict high demand.**
3. About 75% of the times that there is high demand, the model will predict high demand.
4. **About 10% of the times that there is low demand, the model will predict high demand.**
5. About 25% of the times that there is low demand, the model will predict high demand.
6. About 7% of the times that there is low demand, the model will predict high demand.

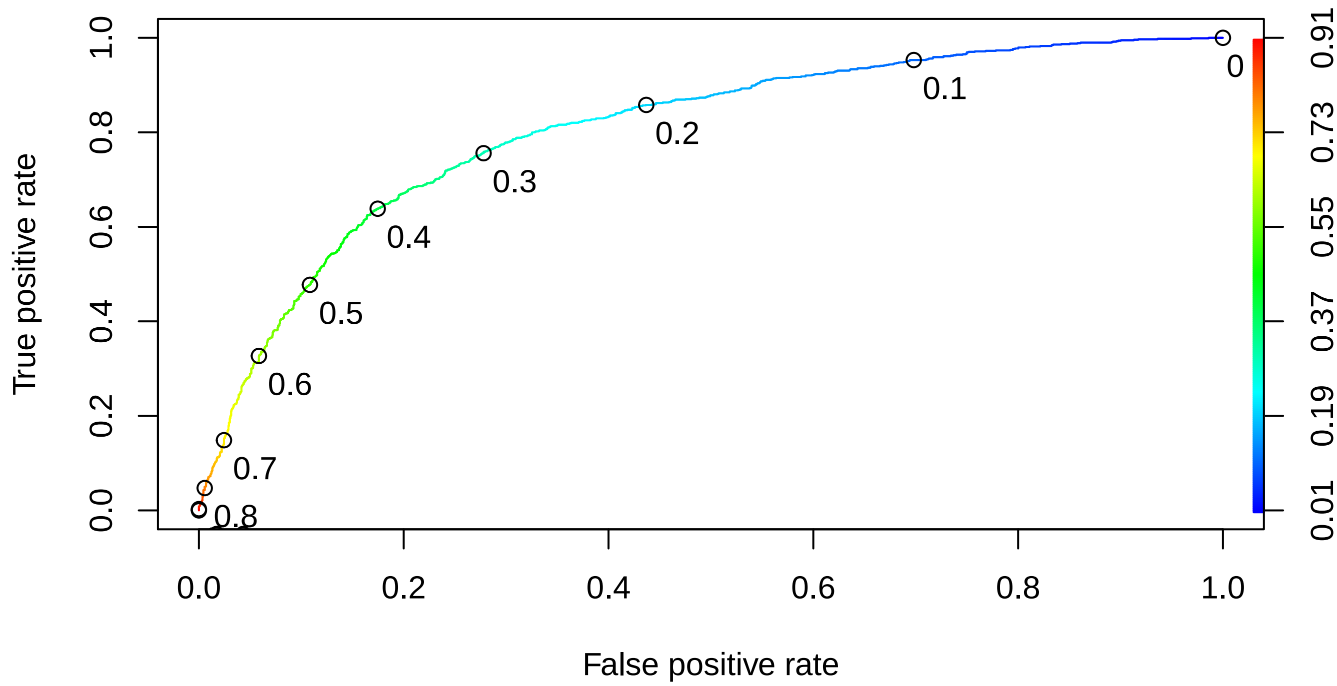
**3.2.3** Plot the ROC curve for your logistic regression model.




**Which logistic regression threshold is associated with the lower-left corner of the ROC plot (true positive rate 0 and false positive rate 0)?**

1. 0
2. 0.5
3. **1**

#### 3.2.4 At roughly which logistic regression cutoff does the model achieve a true positive rate of 80% and a false positive rate of 40%?




*Answer* :

1. 0.01
2. **0.19**
3. 0.37
4. 0.55
5. 0.73
6. 0.91

#### 3.2.5 What is the AUC for your logistic regression model?

```
## [1] 0.8031658
```

*Answer*: 0.8031658

## *Problem 4 : CART*

#### 4.1

#### Set the random seed to 100.

Then use the caret package and the train function to perform 10-fold cross validation with the training data set to select the best cp value for a CART model that predicts the dependent variable *demand_level* using *all of the possible independent variables except count* which was used to define the dependent variable. Select the cp value from a grid consisting of the values 0.0001, 0.0002, 0.0003, . . . , 0.02.

Remember to convert the demand_level column to a factor variable.

If you have called your training set train, use the following code:

train$demand_level = as.factor(train$demand_level)

#### 4.1.1

```
## CART
##
## 7620 samples
##   10 predictor
##    2 classes: '0', '1'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 6858, 6858, 6858, 6858, 6859, 6859, ...
## Resampling results across tuning parameters:
##
##   cp      Accuracy   Kappa
##   0.0001  0.8771645  0.7050045
##   0.0002  0.8796576  0.7108565
##   0.0003  0.8812336  0.7149349
##   0.0004  0.8812338  0.7147262
##   0.0005  0.8818891  0.7164062
##   0.0006  0.8814959  0.7153616
##   0.0007  0.8816275  0.7152725
##   0.0008  0.8817581  0.7152997
##   0.0009  0.8799189  0.7101193
##   0.0010  0.8808384  0.7116719
##   0.0011  0.8803135  0.7100818
##   0.0012  0.8800515  0.7087110
##   0.0013  0.8824142  0.7131980
##   0.0014  0.8821519  0.7122620
##   0.0015  0.8824147  0.7124114
##   0.0016  0.8822837  0.7119029
##   0.0017  0.8820212  0.7111081
##   0.0018  0.8807101  0.7073830
##   0.0019  0.8807101  0.7073830
##   0.0020  0.8792660  0.7047884
##   0.0021  0.8792660  0.7047884
##   0.0022  0.8783487  0.7022330
##   0.0023  0.8770350  0.6983291
##   0.0024  0.8770350  0.6983291
##   0.0025  0.8728399  0.6863527
##   0.0026  0.8725774  0.6847271
##   0.0027  0.8717900  0.6829382
##   0.0028  0.8704783  0.6777971
##   0.0029  0.8698213  0.6757721
##   0.0030  0.8687720  0.6719631
##   0.0031  0.8686407  0.6714946
##   0.0032  0.8687720  0.6717877
```

```
##    0.0033  0.8690348  0.6722047
##    0.0034  0.8690348  0.6722047
##    0.0035  0.8698218  0.6725801
##    0.0036  0.8687740  0.6679670
##    0.0037  0.8685117  0.6664037
##    0.0038  0.8675931  0.6630438
##    0.0039  0.8674619  0.6636094
##    0.0040  0.8674619  0.6636094
##    0.0041  0.8677243  0.6638259
##    0.0042  0.8678556  0.6644698
##    0.0043  0.8678556  0.6644698
##    0.0044  0.8666755  0.6619898
##    0.0045  0.8666755  0.6619898
##    0.0046  0.8666755  0.6619898
##    0.0047  0.8660202  0.6599013
##    0.0048  0.8660202  0.6599013
##    0.0049  0.8615558  0.6475848
##    0.0050  0.8615558  0.6475848
##    0.0051  0.8615558  0.6475848
##    0.0052  0.8611616  0.6462203
##    0.0053  0.8611616  0.6462203
##    0.0054  0.8599793  0.6437120
##    0.0055  0.8599793  0.6437120
##    0.0056  0.8595856  0.6426736
##    0.0057  0.8595856  0.6426736
##    0.0058  0.8595856  0.6426736
##    0.0059  0.8584045  0.6397574
##    0.0060  0.8573532  0.6364415
##    0.0061  0.8568276  0.6347088
##    0.0062  0.8568276  0.6347088
##    0.0063  0.8568276  0.6347088
##    0.0064  0.8568276  0.6347088
##    0.0065  0.8568276  0.6347088
##    0.0066  0.8573532  0.6354008
##    0.0067  0.8559078  0.6308312
##    0.0068  0.8559078  0.6308312
##    0.0069  0.8559078  0.6308312
##    0.0070  0.8559078  0.6308312
##    0.0071  0.8559078  0.6308312
##    0.0072  0.8559078  0.6308312
##    0.0073  0.8559078  0.6308312
##    0.0074  0.8559078  0.6308312
##    0.0075  0.8559078  0.6308312
##    0.0076  0.8559078  0.6308312
##    0.0077  0.8551214  0.6278967
##    0.0078  0.8551214  0.6278967
##    0.0079  0.8551214  0.6278967
##    0.0080  0.8551214  0.6278967
##    0.0081  0.8551214  0.6278967
##    0.0082  0.8551214  0.6278967
##    0.0083  0.8555156  0.6296165
##    0.0084  0.8555156  0.6296165
##    0.0085  0.8555156  0.6296165
##    0.0086  0.8555156  0.6296165
```

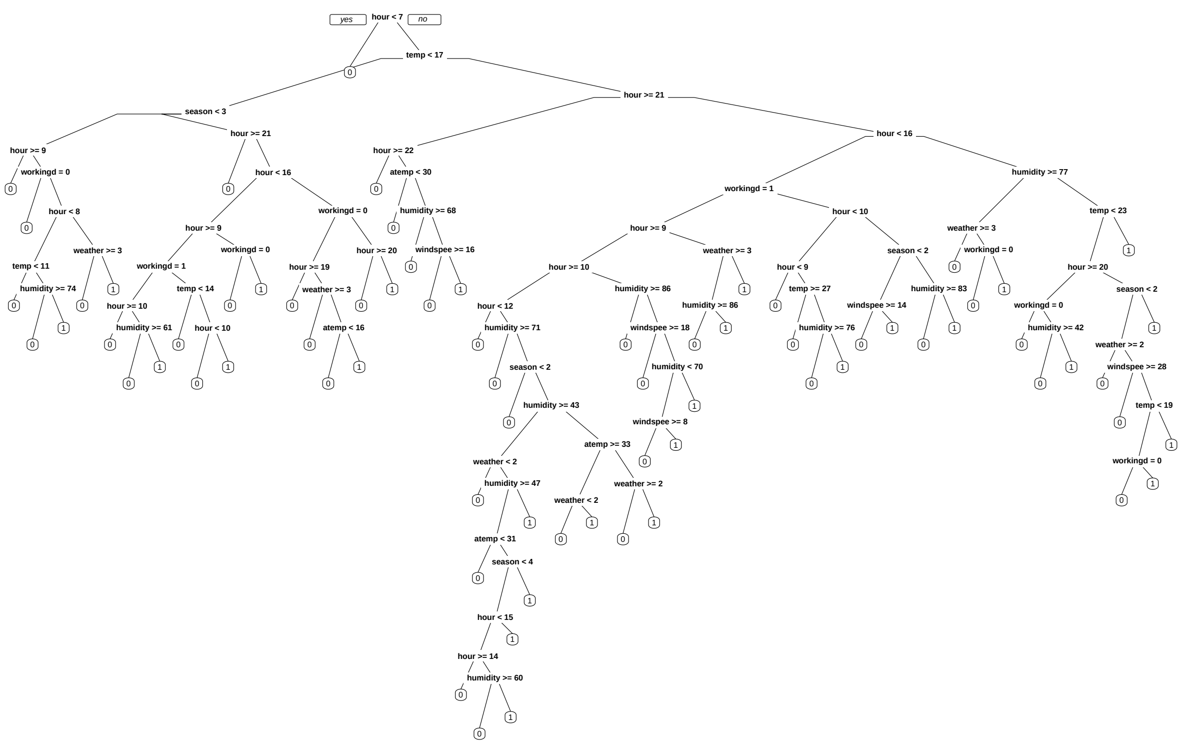
```
##   0.0087  0.8555156  0.6296165
##   0.0088  0.8555156  0.6296165
##   0.0089  0.8555156  0.6296165
##   0.0090  0.8555156  0.6296165
##   0.0091  0.8555156  0.6296165
##   0.0092  0.8555156  0.6296165
##   0.0093  0.8555156  0.6296165
##   0.0094  0.8555156  0.6296165
##   0.0095  0.8555156  0.6296165
##   0.0096  0.8555156  0.6296165
##   0.0097  0.8555156  0.6296165
##   0.0098  0.8553844  0.6294168
##   0.0099  0.8553844  0.6294168
##   0.0100  0.8553844  0.6294168
##   0.0101  0.8553844  0.6294168
##   0.0102  0.8553844  0.6294168
##   0.0103  0.8553844  0.6294168
##   0.0104  0.8553844  0.6294168
##   0.0105  0.8553844  0.6294168
##   0.0106  0.8553844  0.6294168
##   0.0107  0.8553844  0.6294168
##   0.0108  0.8553844  0.6294168
##   0.0109  0.8553844  0.6294168
##   0.0110  0.8553844  0.6294168
##   0.0111  0.8553844  0.6294168
##   0.0112  0.8577450  0.6371550
##   0.0113  0.8577450  0.6371550
##   0.0114  0.8577450  0.6371550
##   0.0115  0.8577450  0.6371550
##   0.0116  0.8577450  0.6371550
##   0.0117  0.8577450  0.6371550
##   0.0118  0.8577450  0.6371550
##   0.0119  0.8577450  0.6371550
##   0.0120  0.8577450  0.6371550
##   0.0121  0.8577450  0.6371550
##   0.0122  0.8577450  0.6371550
##   0.0123  0.8577450  0.6371550
##   0.0124  0.8577450  0.6371550
##   0.0125  0.8577450  0.6371550
##   0.0126  0.8577450  0.6371550
##   0.0127  0.8577450  0.6371550
##   0.0128  0.8577450  0.6371550
##   0.0129  0.8577450  0.6371550
##   0.0130  0.8577450  0.6371550
##   0.0131  0.8577450  0.6371550
##   0.0132  0.8577450  0.6371550
##   0.0133  0.8577450  0.6371550
##   0.0134  0.8577450  0.6371550
##   0.0135  0.8577450  0.6371550
##   0.0136  0.8577450  0.6371550
##   0.0137  0.8577450  0.6371550
##   0.0138  0.8577450  0.6371550
##   0.0139  0.8577450  0.6371550
##   0.0140  0.8577450  0.6371550
```

```
## 0.0141 0.8577450 0.6371550
## 0.0142 0.8577450 0.6371550
## 0.0143 0.8577450 0.6371550
## 0.0144 0.8577450 0.6371550
## 0.0145 0.8577450 0.6371550
## 0.0146 0.8577450 0.6371550
## 0.0147 0.8577450 0.6371550
## 0.0148 0.8577450 0.6371550
## 0.0149 0.8577450 0.6371550
## 0.0150 0.8577450 0.6371550
## 0.0151 0.8577450 0.6371550
## 0.0152 0.8577450 0.6371550
## 0.0153 0.8577450 0.6371550
## 0.0154 0.8577450 0.6371550
## 0.0155 0.8577450 0.6371550
## 0.0156 0.8577450 0.6371550
## 0.0157 0.8577450 0.6371550
## 0.0158 0.8577450 0.6371550
## 0.0159 0.8577450 0.6371550
## 0.0160 0.8577450 0.6371550
## 0.0161 0.8577450 0.6371550
## 0.0162 0.8577450 0.6371550
## 0.0163 0.8577450 0.6371550
## 0.0164 0.8577450 0.6371550
## 0.0165 0.8577450 0.6371550
## 0.0166 0.8574826 0.6371088
## 0.0167 0.8574826 0.6371088
## 0.0168 0.8574826 0.6371088
## 0.0169 0.8574826 0.6371088
## 0.0170 0.8574826 0.6371088
## 0.0171 0.8574826 0.6371088
## 0.0172 0.8574826 0.6371088
## 0.0173 0.8574826 0.6371088
## 0.0174 0.8574826 0.6371088
## 0.0175 0.8574826 0.6371088
## 0.0176 0.8574826 0.6371088
## 0.0177 0.8574826 0.6371088
## 0.0178 0.8574826 0.6371088
## 0.0179 0.8574826 0.6371088
## 0.0180 0.8574826 0.6371088
## 0.0181 0.8574826 0.6371088
## 0.0182 0.8574826 0.6371088
## 0.0183 0.8574826 0.6371088
## 0.0184 0.8574826 0.6371088
## 0.0185 0.8574826 0.6371088
## 0.0186 0.8574826 0.6371088
## 0.0187 0.8574826 0.6371088
## 0.0188 0.8574826 0.6371088
## 0.0189 0.8574826 0.6371088
## 0.0190 0.8574826 0.6371088
## 0.0191 0.8574826 0.6371088
## 0.0192 0.8574826 0.6371088
## 0.0193 0.8574826 0.6371088
## 0.0194 0.8574826 0.6371088
```

```
##   0.0195  0.8545954  0.6331569
##   0.0196  0.8545954  0.6331569
##   0.0197  0.8545954  0.6331569
##   0.0198  0.8545954  0.6331569
##   0.0199  0.8545954  0.6331569
##   0.0200  0.8545954  0.6331569
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was cp = 0.0015.
```

*Answer* : 0.001

**4.1.2**




**What does the first split indicate?**

1. There will be a high demand of bikes before 7 AM.
2. **There will not be a high demand of bikes before 7 AM.**
3. If the hour is before 7 AM, we should look at the temperature.
4. If the hour is before 7 AM and the temperature is less than 17, there will not be a high demand

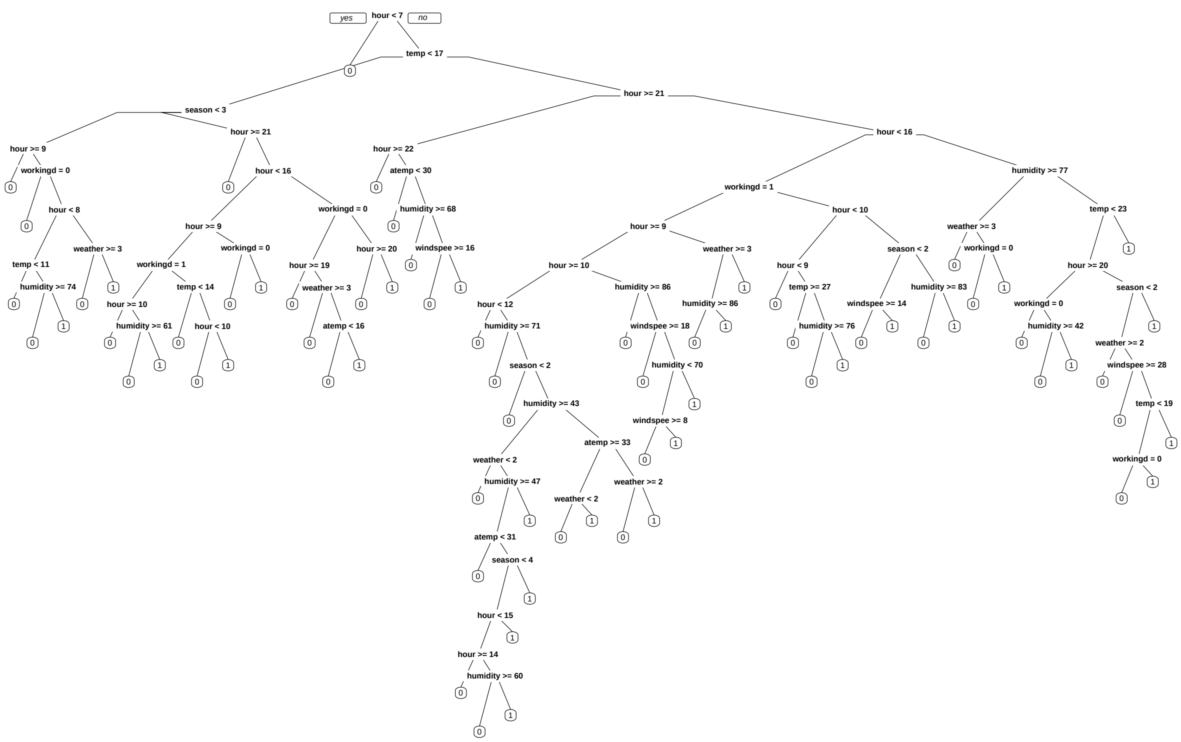
###**4.2**

### 4.2.1 What is the (test) accuracy of your CART model?

$$Accuracy = \frac{TruePositive + TrueNegative}{Ntotal}$$

```
## [1] 0.8876301
```

### 4.2.2 What does the CART model predict on a Saturday, spring day at 9 AM when the temparature is 15 degrees Celsius?




1. high demand
2. **low demand**
3. Not enough information