1.0 BIKE SHARING DATASET USING LINEAR REGRESSION

1.1 Problem Identification and Overview of the data set:

The dataset “Bike-Sharing-Dataset” was obtained from the UCI Machine Learning Repository. This archive was created in 2013 by Fanaee-T, Hadi Gama and Joao. This dataset contains the hourly and daily count of rental bikes between years 2011 and 2012 in Capital bikeshare system with corresponding weather and seasonal information. Capital bike share has about over 500 bike-sharing programs around the world which is composed of over 500 thousands bicycles. Bike sharing systems are the new modern styles of traditional bike rental processes. The entire process from membership, rental and return have become automatic which has enabled users to easily rent and own bikes from any location, also being able to return this bike at another location other than where it was rented from. (Fanaee, Hadi, & Joao, 2013).

Linear regression is one of the simplest and most common supervised machine learning algorithms that data scientists use for predictive modelling. In this report, I will use linear regression to build a model that predicts the daily normalized feeling temperature (atemp) from metrics that are much easier for peoples who study weather to measure for future dates. (Fanaee, Hadi, & Joao, 2013).

1.2 Data Exploration

The bike data set is not included in base R’s datasets package, hence, I had to download the dataset from the UCI Machine Learning Repository (http://archi/e.iics.iuci.iedu/me/). This dataset encompasses the hourly and daily count of rental bikes between years 2011 and 2012 in capital bikeshare system with the correlating weather and seasonal information.

This data set consists of 731 observations of 16 numeric variables describing information on rental bikes hourly and daily counts.

```
$ instant   : int  1 2 3 4 5 6 7 8 9 10 ...
$ dteday    : Factor w/ 731 levels "2011-01-01","2011-01-02",..: 1 2 3 4 5 6 7 8 9 10 ...
$ season    : int  1 1 1 1 1 1 1 1 1 1 ...
$ yr        : int  0 0 0 0 0 0 0 0 0 0 ...
$ mnth      : int  1 1 1 1 1 1 1 1 1 1 ...
$ holiday   : int  0 0 0 0 0 0 0 0 0 0 ...
$ weekday   : int  6 0 1 2 3 4 5 6 0 1 ...
$ workingday: int  0 0 1 1 1 1 1 0 0 1 ...
$ weathersit: int  2 2 1 1 1 1 2 2 1 1 ...
$ temp      : num  0.344 0.363 0.196 0.2 0.227 ...
$ atemp     : num  0.364 0.354 0.189 0.212 0.229 ...
$ hum       : num  0.806 0.696 0.437 0.59 0.437 ...
$ windspeed : num  0.16 0.249 0.248 0.16 0.187 ...
$ casual    : int  331 131 120 108 82 88 148 68 54 41 ...
$ registered: int  654 670 1229 1454 1518 1518 1362 891 768 1280 ...
$ cnt       : int  985 801 1349 1562 1600 1606 1510 959 822 1321 ...
```

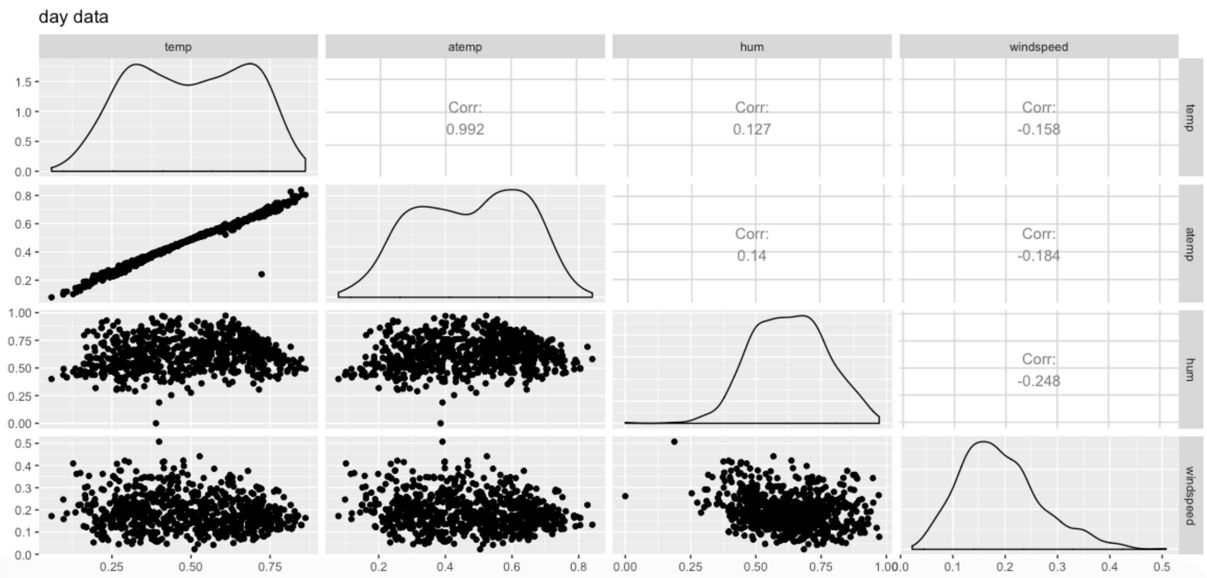
Variable names or column names

```
> names(day)
 [1] "instant"    "dteday"     "season"     "yr"         "mnth"       "holiday"
 [7] "weekday"    "workingday" "weathersit" "temp"       "atemp"      "hum"
[13] "windspeed"  "casual"     "registered" "cnt"
```

Get the first five rows

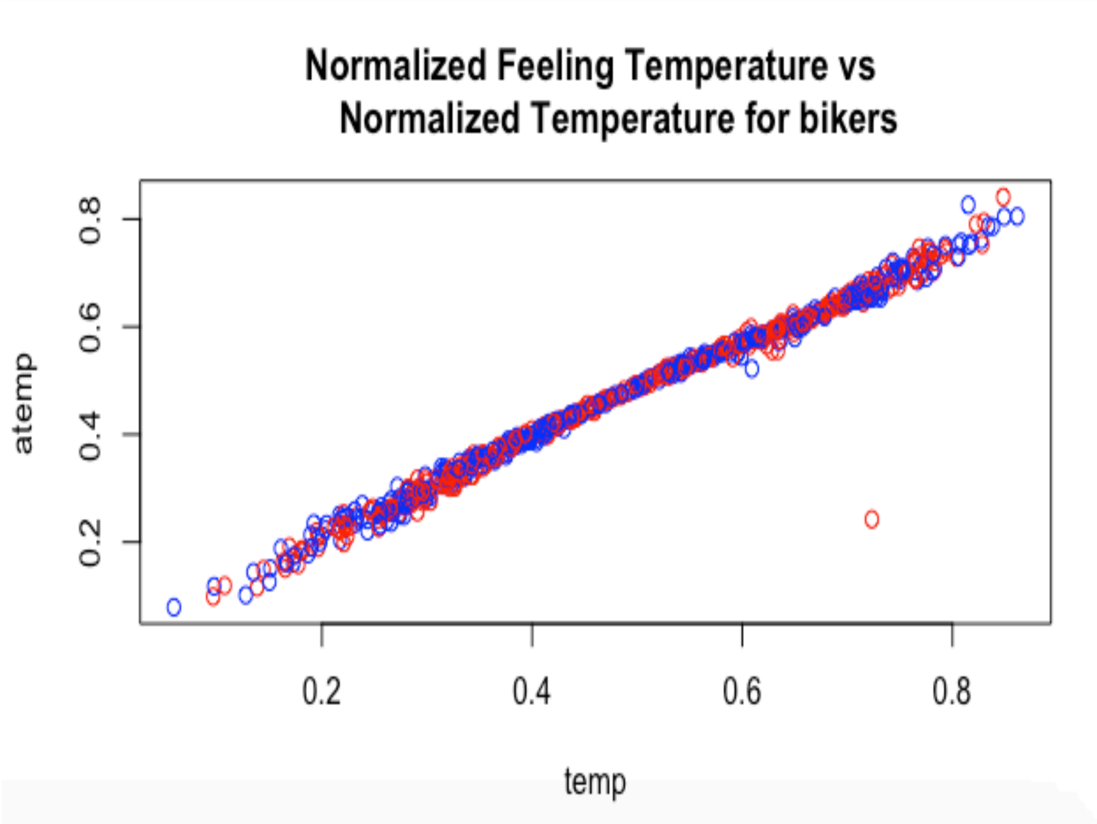
```
> day[1:5,]
  instant     dteday season yr mnth holiday weekday workingday weathersit     temp
1       1 2011-01-01      1  0    1       0       6          0          2 0.344167
2       2 2011-01-02      1  0    1       0       0          0          2 0.363478
3       3 2011-01-03      1  0    1       0       1          1          1 0.196364
4       4 2011-01-04      1  0    1       0       2          1          1 0.200000
5       5 2011-01-05      1  0    1       0       3          1          1 0.226957
     atemp      hum windspeed casual registered  cnt
1 0.363625 0.805833  0.160446    331        654  985
2 0.353739 0.696087  0.248539    131        670  801
3 0.189405 0.437273  0.248309    120       1229 1349
4 0.212122 0.590435  0.160296    108       1454 1562
5 0.229270 0.436957  0.186900     82       1518 1600
```

To determine if we can make a predictive model, the first approach is to see if there appears to be a connection between the predictor and response variables. Using some exploratory data visualization, the ggpairs() function from the GGally package , to create a plot matrix to see how the variables associate with one another.



The ggpairs() function gives scatter plots for each variable combination, and also density plots for each variable and the strength of correlations between variables. From looking at the ggpairs() output, temp definitely seems to be related to atemp: the correlation coefficient is closer to 1, and the points seem to have a linear pattern. The relationship appears to be linear; from the scatter plot, we can see that the atemp increases consistently as the temp increases. (Martin, 2018)

```
# plot scatterplot
attach(day)
plot(temp, atemp, main="Normalized Feeling Temperature vs
     Normalized Temperature for bikers", col=c("red","blue"))
cor(temp, atemp)
```



1.3 <u>Definition of Training & Test data</u>

The training dataset is the sample of data used to fit the model (Brownlee, 2017). This is the actual dataset that is used to train the model. The test dataset however, is the sample of data used to provide an unbiased evaluation of a final model fit on the training dataset (Shah, 2017). Set seed for random generation and set aside training and test datas.

```
# create a random sample for training and testing
set.seed(1)
day_rand = day[order(runif(731)),]
```

```
# split the data frames
day_train <- day[1:500, ]
day_test <- day[501:731, ]
```

1.4 <u>Model Generation</u>

Building a linear model relating the normalized feel temperature to normal temperature;

```
# build the model
day_model = lm(atemp ~ temp, data=day_train)
```

day_model	list [12] (S3: lm)	List of length 12
coefficients	double [2]	0.0328 0.8933
residuals	double [500]	0.023355 -0.003782 -0.018830 0.000639 -0.006294 0.017842 ...
effects	double [500]	-9.98e+00 3.58e+00 -1.95e-02 -5.98e-05 -7.02e-03 1.71e-02 ...
rank	integer [1]	2
fitted.values	double [500]	0.340 0.358 0.208 0.211 0.236 0.215 ...
assign	integer [2]	0 1
qr	list [5] (S3: qr)	List of length 5
df.residual	integer [1]	498
xlevels	list [0]	List of length 0
call	language	lm(formula = atemp ~ temp, data = day_train)
terms	formula	atemp ~ temp
model	list [500 x 2] (S3: data.frame)	A data.frame with 500 rows and 2 columns

1.5 Predictions & Evaluation of the test data

Calling the output model using summary() will provide the information that is needed to test the hypothesis and assess how well the model fits the data.

```
> summary(day_model)

Call:
lm(formula = atemp ~ temp, data = day_train)

Residuals:
      Min        1Q    Median        3Q       Max 
-0.054876 -0.008534  0.001837  0.009654  0.065495 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 0.032818   0.001707   19.23   <2e-16 ***
temp        0.893322   0.003441  259.61   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.01377 on 498 degrees of freedom
Multiple R-squared:  0.9927,	Adjusted R-squared:  0.9927 
F-statistic: 6.739e+04 on 1 and 498 DF,  p-value: < 2.2e-16
```

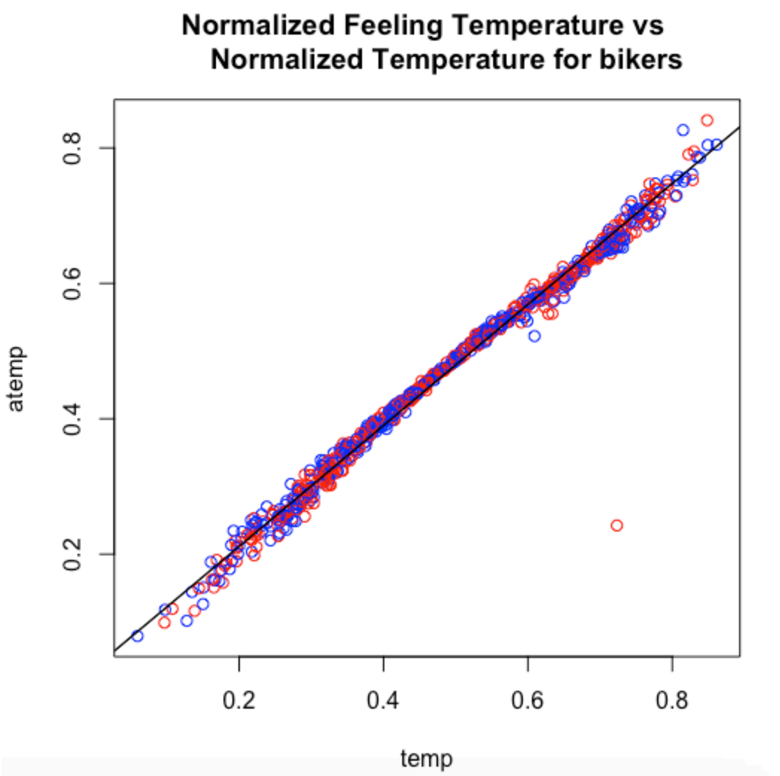
The coefficient estimate contains two rows; the first one is the intercept. The intercept is essentially the expected value of the normalized feel temperature required for the weather to reach when we consider the average normal temperature of all days in the dataset. In other words, it takes an average day in this dataset 0.032818˚C to reach a normalized feel temperature. The second row in the Coefficients is the slope. We see that for each additional 0.002˚C of normalized temperature, the normalized feel temperature increases by 0.893322˚C. The Normalized temperature and Intercept are statistically significant at the 99.9% level as its p value is < .001.

We can also look at the errors and calculate the RMSE in the data;

```
> errors <- residuals(day_model)
> squared.errors <- errors ^ 2
> mse <- mean(squared.errors)
> rmse <- sqrt(mse)
> rmse
[1] 0.01374588
```

Therefore, 0.01374588 is the difference between the observed value and the predicted value by the model. Now let’s have a look at the abline fitted to the data for temp and atemp.

```
# add line of best fit to the scatterplot
abline(day_model)
```



From this, I can use the model to have prediction values of Normalized Feeling temperatures from the normal temperature for days that were left out of the dataset or future dates to be considered. Taking a random values in testdata predictions for temp & atemp, we can make predictions;

```
> # use the model to predict atemp values for the temp values
> testdata = data.frame(temp=0.2334567, atemp=0.2123425)
> prediction = predict(day_model, testdata)
> prediction
        1
0.2413703
```

1.6 Conclusion

When there is no access to human experts, background knowledge can be an appropriate alternative. The scale of the train and test data sets necessarily should not be the same. Looking at the analysis, for every additional 0.002˚C of normalized temperature(temp), the normalized feel temperature(atemp) increases by 0.893322˚C which makes it feel colder for bikers than the actual temperature for that day. You can again see that the predicted value for the test data is very close to the p-value in the summary of the model, which in turn means this prediction is correct.

In essence, this analysis shows that there is a positive relationship between the normalized feel temperature and the normalized temperature. The warmer or colder it is, the warmer or colder it actually feels for bikers. This could be important of planning new bike rental stations for future years.

2.0 ADULT DATA SET USING DECISION TREE

2.1 Problem Identification and Overview of the dataset:

This dataset is gotten from the UCI Repository. The Adult dataset also known as "Census Income" was written by Ronny Kohavi and Barry Becker from the 1994 census database. The dataset contains a set of many records ranging from age, employment, occupation, sex and so on. The aim of this task is to predict whether a person makes over 50k income a year. (Kohavi & Becker, 1996).

2.2 Data Exploration

The dataset is read into R by importing the from the UCI repo. It consists of 32561 observations and 12 variables. Some of the variables are not self-explanatory. The continuous variable 'fnlwgt' represents final weight, which is the number of units in the target population that the responding unit represents. The variable 'education_num' stands for the number of years of education in total, which is a continuous representation of the discrete variable education. The variable relationship represents the responding unit's role in the family. 'capital_gain' and 'capital_loss' are income from investment sources other than wage/salary. (Zhu, 2016).

```
'data.frame':	32561 obs. of  12 variables:
 $ age           : int  39 50 38 53 28 37 49 52 31 42 ...
 $ workclass     : Factor w/ 9 levels " ?"," Federal-gov",..: 8 7 5 5 5 5 5 7 5 5 ...
 $ educatoin_num : int  13 13 9 7 13 14 5 9 14 13 ...
 $ marital_status: Factor w/ 7 levels " Divorced"," Married-AF-spouse",..: 5 3 1 3 3 3 4 3 5 3 ...
 $ occupation    : Factor w/ 15 levels " ?"," Adm-clerical",..: 2 5 7 7 11 5 9 5 11 5 ...
 $ race          : Factor w/ 5 levels " Amer-Indian-Eskimo",..: 5 5 5 3 3 5 3 5 5 5 ...
 $ sex           : Factor w/ 2 levels " Female"," Male": 2 2 2 2 1 1 1 2 1 2 ...
 $ capital_gain  : int  2174 0 0 0 0 0 0 0 14084 5178 ...
 $ capital_loss  : int  0 0 0 0 0 0 0 0 0 0 ...
 $ hours_per_week: int  40 13 40 40 40 40 16 45 50 40 ...
 $ native_country: Factor w/ 42 levels " ?"," Cambodia",..: 40 40 40 40 6 40 24 40 40 40 ...
 $ income        : Factor w/ 2 levels " <=50K"," >50K": 1 1 1 1 1 1 1 2 2 2 ...
```

The summary of adult occupations are;

```
> summary(adult$occupation)
                ?      Adm-clerical      Armed-Forces 
             1843              3770                 9 
     Craft-repair   Exec-managerial   Farming-fishing 
             4099              4066               994 
Handlers-cleaners Machine-op-inspct     Other-service 
             1370              2002              3295 
  Priv-house-serv    Prof-specialty   Protective-serv 
              149              4140               649 
            Sales      Tech-support  Transport-moving 
             3650               928              1597 
```

Some tables are as follows;

```
> table(adult$marital_status)

             Divorced     Married-AF-spouse    Married-civ-spouse 
                 4443                    23                 14976 
Married-spouse-absent         Never-married             Separated 
                  418                 10683                  1025 
              Widowed 
                  993 
```

```
> table(adult$race)

Amer-Indian-Eskimo  Asian-Pac-Islander               Black               Other 
               311                1039                3124                 271 
             White 
             27816 
```

2.3 Definition of Training and Test Data:

The training dataset is the sample of data used to fit the model (Brownlee, 2017). This is the actual dataset that is used to train the model. The test dataset however, is the sample of data used to provide an unbiased evaluation of a final model fit on the training dataset (Shah, 2017). Set seed for random generation and set aside training and test data.

```
# set aside some data for testing purposes
set.seed(42)
adult_rand = adult[order(runif(32561)), ]
summary(adult$workclass)
summary(adult_rand$workclass)
head(adult$workclass)
head(adult_rand$workclass)

#split the dataframe
adult_train = adult_rand[1:30000, ]
adult_test = adult_rand[30001:32561,]
prop.table(table(adult_train$income))
prop.table(table(adult_test$income))
```

2.4 <u>Model Generation:</u>

Majority of the original data is used as the training set, while the rest is used as test set. A decision tree is used using income as the response variable, and all other variables as predictors is fitted. Its predictors, samples and tree size are reported as below.

```
# build the model
adult_model <- C5.0(income~., data=adult_train)
adult_model
# display detailed information about the tree
summary(adult_model)
```

```
> adult_model

Call:
C5.0.formula(formula = income ~ ., data = adult_train)

Classification Tree
Number of samples: 30000
Number of predictors: 14

Tree size: 91

Non-standard options: attempt to group attributes
```

2.5 <u>Prediction and Evaluation of Test data:</u>

Having built my model, I can now make predictions in the test data;

```
# Create a factor vector of predictions on test data
adult_predict <- predict(adult_model, adult_test)
adult_predict
summary(adult_predict)

# cross tabulation of predicted versus actual classes
CrossTable(adult_predict, adult_test$income,
           prop.chisq = FALSE, prop.c = FALSE, prop.r = FALSE,
           dnn = c('predicted income', 'actual income'))
```

The output predicted 2041 values of incomes <=50k and 520 values of incomes >50k. Using cross table() function to produce a confusion matrix, we can see the output as the following;

```
> summary(adult_predict)
 <=50K   >50K
  2041    520
> # cross tabulation of predicted versus actual classes
> CrossTable(adult_predict, adult_test$income,
+            prop.chisq = FALSE, prop.c = FALSE, prop.r = FALSE,
+            dnn = c('predicted income', 'actual income'))


   Cell Contents
|-------------------------|
|                       N |
|         N / Table Total |
|-------------------------|


Total Observations in Table:  2561


                 | actual income
predicted income |     <=50K |      >50K | Row Total |
-----------------|-----------|-----------|-----------|
           <=50K |      1821 |       220 |      2041 |
                 |     0.711 |     0.086 |           |
-----------------|-----------|-----------|-----------|
            >50K |       103 |       417 |       520 |
                 |     0.040 |     0.163 |           |
-----------------|-----------|-----------|-----------|
    Column Total |      1924 |       637 |      2561 |
-----------------|-----------|-----------|-----------|
```

The model had a total number of 2561 predictions. 1924 predictions were <=50k, while 637 predictions were >50k. Though in the real data, 2041 values were <=50k, while 520 values were >50k.

2.6 <u>Conclusion</u>:

From the confusion matrix, the model has an 87.39% accuracy of being correct and a misclassification rate (error rate) of 12.61% to be wrong. The True positive rate or sensitivity is 80.19%, while the false positive rate is 10.85%. The true negative rate is 89.22%.

Decision trees are a non-parametric supervised learning method used for classification. While it performed great on the training set, it performed less than expected on the test set.

3.0 ADULT DATA SET USING KNN

3.1 Problem Identification and Overview of the dataset:

This dataset is gotten from the UCI Repository. The Adult dataset also known as "Census Income" was written by Ronny Kohavi and Barry Becker from the 1994 census database. The dataset contains a set of many records ranging from age, employment, occupation, sex and so on. The aim of this task is to predict whether a person makes over 50k income a year. (Kohavi & Becker, 1996).

3.2 Data Exploration

This is the same data set with the section 2.0, hence the same attributes are explored.

3.3 Definition of Training and Test Data:

The training dataset is the sample of data used to fit the model (Brownlee, 2017). This is the actual dataset that is used to train the model. The test dataset however, is the sample of data used to provide an unbiased evaluation of a final model fit on the training dataset (Shah, 2017). Set seed for random generation and set aside training and test data.

```
# create normalization function
normalize <- function(x) { return ((x - min(x)) / (max(x) - min(x))) }

# normalize the adult data
adult_n <- as.data.frame(lapply(adult[c(1,3,5,11,12,13)], normalize))

# set aside some data for testing purposes
set.seed(42)

#split the dataframe
adult_train = adult_n[1:30000, ]
adult_test = adult_n[30001:32561,]
```

3.4 Model Generation:

Using KNN classification does not require a model formula unlike decision trees. Using the library class, a test prediction is made after creating training and test labels from the image above

```
# Training a model on the data
library(class)
adult_test_pred <- knn(train = adult_train, test =
                         adult_test, cl = adult_train_labels, k=1)
```

3.5 Prediction and Evaluation of Test data:

Having trained my model, I can then analyse the cross tables of the predicted on the actual data.

```
> summary(adult_test_pred)
 <=50K  >50K
  1949   612
```

```r
# Create the cross tabulation of predicted vs. actual
CrossTable(x = adult_test_labels, y = adult_test_pred, prop.chisq=FALSE)
```

```
   Cell Contents
|-------------------------|
|                       N |
|           N / Row Total |
|           N / Col Total |
|         N / Table Total |
|-------------------------|


Total Observations in Table:  2561


                  | adult_test_pred
adult_test_labels |     <=50K |      >50K | Row Total |
------------------|-----------|-----------|-----------|
            <=50K |      1619 |       300 |      1919 |
                  |     0.844 |     0.156 |     0.749 |
                  |     0.831 |     0.490 |           |
                  |     0.632 |     0.117 |           |
------------------|-----------|-----------|-----------|
             >50K |       330 |       312 |       642 |
                  |     0.514 |     0.486 |     0.251 |
                  |     0.169 |     0.510 |           |
                  |     0.129 |     0.122 |           |
------------------|-----------|-----------|-----------|
     Column Total |      1949 |       612 |      2561 |
                  |     0.761 |     0.239 |           |
------------------|-----------|-----------|-----------|
```

The model had a total number of 2561 predictions. 1949 predictions were <=50k, while 612 predictions were >50k. Though in the real data, 1919 values were <=50k, while 642 values were >50k.

3.6 Conclusion:

From the confusion matrix, the model has an 81.37% accuracy of being correct and a misclassification rate (error rate) of 18.63% to be wrong. The True positive rate or sensitivity is 43.93%, while the false positive rate is 6.1%. The true negative rate is 93.90%.

Two types of classification techniques were carried out on the same data set, adult income. The accuracy for using decision tree technique was 87.83%, while 81.37% using KNN. In other words, using decision tree for this dataset yields a more accurate prediction on if the census income exceeds $50K/yr.

Works Cited

- Brownlee, J. (2017, July 14). *What is the Difference Between Test and Validation Datasets?* Retrieved from Machine Learning Mastery: https://machinelearningmastery.com/difference-test-validation-datasets/
- Fanaee, T., Hadi, G., & Joao. (2013). *Event labeling combining ensemble detectors and background knowledge.* Retrieved from UCI Machine Learning Repository: http://dx.doi.org/10.1007/s13748-013-0040-3
- Kohavi, R., & Becker, B. (1996). *Scaling Up the Accuracy of Naive-Bayes Classifiers: a Decision-Tree Hybrid.* Retrieved from UCI Machine Learning Repository: https://archive.ics.uci.edu/ml/datasets/Adult
- Martin, R. (2018, May 16). *Using Linear Regression for Predictive Modeling in R*. Retrieved from DATAQUEST: https://www.dataquest.io/blog/statistical-learning-for-predictive-modeling-r/
- Rego, F. (2015, oct 23). *QUICK GUIDE: INTERPRETING SIMPLE LINEAR MODEL OUTPUT IN R*. Retrieved from Felipe Rego: https://feliperego.github.io/blog/2015/10/23/Interpreting-Model-Output-In-R
- School, D. (2018, October 31). *Making sense of the confusion matrix*. Retrieved from Youtube: https://www.youtube.com/watch?v=8Oog7TXHvFY
- Shah, T. (2017, December 6). *About Train, Validation and Test Sets in Machine Learning*. Retrieved from Towards Data Science: https://towardsdatascience.com/train-validation-and-test-sets-72cb40cba9e7
- Zhu, H. (2016, December 5). *Predicting Earning Potential using the Adult Dataset* . Retrieved from RPubs: https://rpubs.com/H_Zhu/235617