Load the dataset auto.csv (given in the shared Google folder) into R. This dataset has some columns:

**name**: name of the car model (e.g., bmw 2002)

**origin**: where the car is produced. 1 = US, 2 = EU, 3 = Asia.

**mpg**: miles per gallon (the higher the better).

**weight**: weight in lbs

**model\_year**: the model year

**horsepower**: the engine power (measured in horsepower). The higher the stronger.

**cylinders**: the number of cylinders in the car engine. The higher the stronger.

**acceleration**: the time (in seconds) for the car to speed up from 0 to 60 mph.

**displacement**: the volume of air the car engine can take in to burn.

To answer the below questions, we need to set the directory where the “auto.csv” file is located.

In my local machine, the code to set the working directory is

setwd(“C:/Users/cscha/OneDrive/Desktop/Software analytics”)

Now read the “auto.csv” file as typing the following code

> cars <- read.csv(“auto.csv”)

Here, “cars” work as a variable to perform operations on the “auto.csv” file

Q1. Compare mpg of cars: US vs Asia; EU vs Asia, US vs EU using t.test and wilcox.test. Describe the results in non-technical language.

To compare the mpg of cars between the US, EU, and Asia using t.test and wilcox. test.

At first, we compare the mpg of the US vs Asia using t.test.

The ‘R’ code to compare the US vs Asia using t.test is as follows:

> mpg\_of\_us <- cars[cars$origin==1, "mpg"]

> mpg\_of\_asia <- cars[cars$origin==3,"mpg"]

> t\_test\_of\_us\_asia <- t.test(mpg\_of\_us, mpg\_of\_asia)

> print(t\_test\_of\_us\_asia)

Here in the above code mpg\_of\_us <- cars[cars$origin==1, "mpg"], the values of mpg which contain origin value as “1” are stored in a variable called “mpg\_of\_us”.

The line mpg\_of\_asia <- cars[cars$origin==3, "mpg"], here the values of mpg which contain origin value as “3” are

stored in a variable called “mpg\_of\_asia”.

The line t.test(mpg\_of\_us, mpg\_of\_asia), does the t-test between mpg\_of\_us and mpg\_of\_asia.

Print(t\_test\_of\_us\_asia), print the results of the t-test between mpg\_of\_us and mpg\_of\_asia.

The output is:

data: mpg\_of\_us and mpg\_of\_asia

t = -13.034, df = 138.64, p-value < 2.2e-16

alternative hypothesis: true difference in means is not equal to 0

95 percent confidence interval:

-11.997430 -8.836897

sample estimates:

mean of x mean of y

20.03347 30.45063

From the outputs of the t-test between the mpg of US and Asia, we can state that the p-value is

7.513384e-26 which is significantly less than 0.05. From the know data if the p-value is less than 0.05 we can state

that the cars in the US and Asia have drastically differing fuel efficiency, or to put it another way, these two areas' average mpg is different.

To compare the mpg of cars between the US, EU, and Asia using wilcox test and wilcox. test. At first, we compare

the mpg of the US vs Asia using the wilcox test.

The ‘R’ code to compare the US vs Asia using wilcox test is as follows:

> mpg\_of\_us <- cars[cars$origin==1, "mpg"]

> mpg\_of\_asia <- cars[cars$origin==3,"mpg"]

> wilcox\_test\_of\_us\_asia <- wilcox.test(mpg\_of\_us, mpg\_of\_asia)

> print(wilcox\_test\_of\_us\_asia)

In the above line code, wilcox.test(mpg\_of\_us, mpg\_of\_asia) does the wilcox-test between mpg\_of\_us and mpg\_of\_asia and stores the result in wilcox\_text\_of\_us\_asia variable.

print(wilcox\_test\_of\_us\_asia), prints the resultant output.

The output is:

Wilcoxon rank sum test with continuity correction

data: mpg\_of\_us and mpg\_of\_asia

W = 2456.5, p-value < 2.2e-16

alternative hypothesis: true location shift is not equal to 0

From the output of the wilcox test between the mpg of the US and Asia, we can state that the p-value is

1.915342e-23 which is significantly less than 0.05. Since the value is less than 0.5 the low p-value backs up and

claims that US and Asian cars have a noticeable mpg gap. From both the testing we conclude that mpg values varies significantly in both the US and Asia.

Now we compare the mpg of the US vs EU using t.test.

The ‘R’ code to compare the US vs EU using t.test is as follows:

> mpg\_of\_us <- cars[cars$origin==1, "mpg"]

> mpg\_of\_eu <- cars[cars$origin==2, "mpg"]

> t\_test\_of\_us\_eu <- t.test(mpg\_of\_us, mpg\_of\_eu)

> print(t\_test\_of\_us\_eu)

Here in the above code mpg\_of\_us <- cars[cars$origin==1, "mpg"], the values of mpg which contain origin value as “1” are stored in a variable called “mpg\_of\_us”.

The line mpg\_of\_eu <- cars[cars$origin==2, "mpg"], here the values of mpg which contain origin value as “2” are

stored in a variable called “mpg\_of\_eu”.

The line t.test(mpg\_of\_us, mpg\_of\_eu), does the t-test between mpg\_of\_us and mpg\_of\_eu.

Print(t\_test\_of\_us\_eu), print the results of the t-test between mpg\_of\_us and mpg\_of\_eu.

The output is:

Welch Two Sample t-test

data: mpg\_of\_us and mpg\_of\_eu

t = -8.4311, df = 105.32, p-value = 1.93e-13

alternative hypothesis: true difference in means is not equal to 0

95 percent confidence interval:

-9.349583 -5.789361

sample estimates:

mean of x mean of y

20.03347 27.60294

From the outputs of the t-test between the mpg of US and EU, we can state that the p-value is

1.929839e-13 which is significantly less than 0.05.The exceptionally low p-value from the t-test indicates that there

is a highly significant difference in the mpg (miles per gallon) of US and EU cars. Simply put, it strongly suggests that the mpg ratings of cars in the US and the EU are considerably different on average.

Now we compare the mpg of the US vs Asia using the wilcox test.

The ‘R’ code to compare the US vs Asia using wilcox test is as follows:

> mpg\_of\_us <- cars [cars$origin==1, "mpg"]

> mpg\_of\_eu <- cars [cars$origin==2, "mpg"]

> wilcox\_test\_of\_us\_eu <- wilcox.test (mpg\_of\_us, mpg\_of\_eu)

> print(t\_test\_of\_us\_eu)

In the above line code,wilcox.test(mpg\_of\_us, mpg\_of\_eu) does the wilcox-test between mpg\_of\_us and mpg\_of\_eu and stores the result in wilcox\_text\_of\_us\_eu variable.

print(wilcox\_test\_of\_us\_eu), prints the resultant output.

The output is:

Wilcoxon rank sum test with continuity correction

data: mpg\_of\_us and mpg\_of\_eu

W = 3279, p-value = 1.957e-14

alternative hypothesis: true location shift is not equal to 0

From the output of the wilcox test between the mpg of the US and EU, we can state that the p-value is

1.956963e-14 which is significantly less than 0.05. The Wilcoxon test also offers convincing proof that US and EU

cars have significantly different mileage ratings. Given the incredibly low p-value, it is likely that the difference is

statistically significant. In plainer terms, it proves that fuel efficiency between US and EU cars differs greatly.

In conclusion, these test results find a difference in fuel efficiency between cars produced in the US and those

produced in the EU which also states that these differences advantage over the other in terms of fuel efficiency.

Now we compare the mpg of the US vs Asia using the t-test

The ‘R’ code to compare the mpg of Asia and the EU is as follows

> mpg\_of\_asia <- cars [cars$origin==3, "mpg"]

> mpg\_of\_eu <- cars [cars$origin==2, "mpg"]

> t\_test\_of\_asia\_eu <- t.test (mpg\_of\_asia, mpg\_of\_eu)

> print(t\_test\_of\_asia\_eu)

Here in the above code mpg\_of\_asia <- cars[cars$origin==3, "mpg"], the values of mpg which contain origin value as “3” are stored in a variable called “mpg\_of\_asia”.

The line mpg\_of\_eu <- cars[cars$origin==2, "mpg"], here the values of mpg which contain origin value as “2” are

stored in a variable called “mpg\_of\_eu”.

The line t.test(mpg\_of\_asia, mpg\_of\_eu), does the t-test between mpg\_of\_asia and mpg\_of\_eu.

Print(t\_test\_of\_asia\_eu), print the results of the t-test between mpg\_of\_asia and mpg\_of\_eu.

The output is:

Welch Two Sample t-test

data: mpg\_of\_asia and mpg\_of\_eu

t = 2.7075, df = 137.85, p-value = 0.007637

alternative hypothesis: true difference in means is not equal to 0

95 percent confidence interval:

0.7679996 4.9273838

sample estimates:

mean of x mean of y

30.45063 27.60294

From the outputs of the t-test between the mpg of Asia and EU, we can state that the p-value is 0.007637 indicating

that there is a moderate amount of evidence to support the notion that vehicles in Asia and the EU have lower fuel

efficiency (mpg). There is some evidence that cars in these places may have varied mpg ratings even if this p-value is less than 0.05 but is not exceptionally low.

Now we compare the mpg of the US vs Asia using the wilcox test

The ‘R’ code to compare the mpg of Asia and the EU is as follows

> mpg\_of\_asia <- cars [cars$origin==3, "mpg"]

> mpg\_of\_eu <- cars [cars$origin==2, "mpg"]

> wilcox\_test\_of\_asia\_eu <- wilcox.test (mpg\_of\_asia, mpg\_of\_eu)

> print(wilcox\_test\_of\_asia\_eu)

In the above line code, wilcox.test (mpg\_of\_asia, mpg\_of\_eu) does the wilcox-test between mpg\_of\_asia and

mpg\_of\_eu and stores the result in wilcox\_text\_of\_asia\_eu variable.

print(wilcox\_test\_of\_asia\_eu), prints the resultant output.

The output is:

Wilcoxon rank sum test with continuity correction

data: mpg\_of\_asia and mpg\_of\_eu

W = 3483, p-value = 0.001962

alternative hypothesis: true location shift is not equal to 0

From the outputs of the wilcox test between the mpg of Asia and the EU, it offers more support for the notion that

Asian and EU cars achieve significantly different mileage. With a p-value of 0.0019, the difference is more confidently predicted, which is quite low. To put it another way, it means that automobiles in Asia and the EU probably have fuel efficiency that is very different from one another.

In conclusion, both tests indicate that there is a difference in fuel efficiency between cars in Asia and the EU, but the Wilcox test, with its lower p-value, offers better support for this difference. Though less conclusive than the Wilcox

test, the t-test also shows a difference.

Q2. Compare mpg of cars by cylinders (engine size) of 4, 6, and 8. We exclude cars with cylinders of 3 and 5 because of small sample size. You can use both the t.test and wilcox. test. Describe the results in non-technical language.

To compute the above question, first, we segregate the car data into required\_data with engine size as 4,6, and 8.

The ‘R’ code for this is as follows:

>required\_data <- cars[cars$cylinders %in% c(4,6,8), ]

>cylinders\_4 <- required\_data[required\_data$cylinders ==4, “mpg”]

>cylinders\_6 <- required\_data[required\_data$cylinders == 6, “mpg”]

>cylinders\_8 <- required\_data[required\_data$cylinders == 8, “mpg”]

From the above code required\_data <- cars[cars$cylinders %in% c(4,6,8), ], it has the all data as similar to cars which contains the relevant data with respect to cylinders values as 4,6,8 .

cylinders\_4 <- required\_data [required\_data$cylinders ==4, “mpg”], it has the mpg data where cylinders value is 4.

cylinders\_6 <- required\_data [required\_data$cylinders ==6, “mpg”], it has the mpg data where cylinders value is 6.

cylinders\_8 <- required\_data[required\_data$cylinders ==4, “mpg”], it has the mpg data where cylinders value is 8.

Now we compare the mpg value between the cylinders\_4 and cylinders\_6 using the t-test as follows:

> required\_data <- cars [cars$cylinders %in% c(4,6,8), ]

> cylinders\_4 <- required\_data [required\_data$cylinders == 4, "mpg"]

> cylinders\_6 <- required\_data [required\_data$cylinders == 6, "mpg"]

> cylinders\_8 <- required\_data [required\_data$cylinders == 8, "mpg"]

> t\_test\_of\_cylinders\_4\_cylinders\_6 <- t.test (cylinders\_4, cylinders\_6)

> print(t\_test\_of\_cylinders\_4\_cylinders\_6)

Here in the above code, t.test (cylinders\_4, cylinders\_6) does the t-test of mpg between the cylinders\_4 and cylinders\_6 and stores the result in t\_test\_of\_cylinders\_4\_cylinders\_6.

print(t\_test\_of\_cylinders\_4\_cylinders\_6), prints the output of the respective t-test

The output is:

Welch Two Sample t-test

data: cylinders\_4 and cylinders\_6

t = 16.01, df = 223.27, p-value < 2.2e-16

alternative hypothesis: true difference in means is not equal to 0

95 percent confidence interval:

8.164385 10.456466

sample estimates:

mean of x mean of y

29.28392 19.97349

We can obtain the p-value of the above output by the code as a print(t\_test\_of\_cylinders\_4\_cylinder\_6$p.value)

So, by the above code, we get p-value as 6.244862e-39. The observed difference in mpg between these two groups is extremely significant when the p-value is thus close to zero, making it almost implausible for this difference to have arisen by random variation. This test clearly shows that cars with 4 and 6 cylinders have drastically different fuel economy, with the former being significantly different from the latter.

Now we compare the mpg value between the cylinders\_4 and cylinders\_6 using the wilcox test as follows:

The ‘R’ code for comparison is as follows:

> cylinders\_4 <- required\_data[required\_data$cylinders == 4, "mpg"]

> cylinders\_6 <- required\_data[required\_data$cylinders == 6, "mpg"]

> wilcox\_test\_of\_cylinders\_4\_cylinders\_6 <- wilcox.test(cylinders\_4, cylinders\_6)

> print(wilcox\_test\_of\_cylinders\_4\_cylinders\_6)

Here in the above code, wilcox.test (cylinders\_4, cylinders\_6) does the wilcox test of mpg between the cylinders\_4 and cylinders\_6 and stores the result in wilcox\_test\_of\_cylinders\_4\_cylinders\_6.

print(wilcox\_test\_of\_cylinders\_4\_cylinders\_6), prints the output of the respective wilcox test

The output is:

Wilcoxon rank sum test with continuity correction

data: cylinders\_4 and cylinders\_6

W = 15347, p-value < 2.2e-16

alternative hypothesis: true location shift is not equal to 0

We can obtain the p-value of the above output by the code as print(wilcox\_test\_of\_cylinders\_4\_cylinder\_6$p.value)

So, by the above code, we get a p-value as 6.56572e-30. This low p-value states that mpg between the cylinders with capacities of 4 and 6 is not by a random chance. From the value of W which is 15347 in above output we can also state that it aids a strong piece of evidence to support the p-value statement.

Now we compare the mpg value between the cylinders\_4 and cylinders\_8 using the t-test as follows:

The ‘R’ code is as follows:

> cylinders\_4 <- required\_data[required\_data$cylinders == 4, "mpg"]

> cylinders\_8 <- required\_data[required\_data$cylinders == 8, "mpg"]

> t\_test\_of\_cylinders\_4\_cylinders\_8 <- t.test(cylinders\_4, cylinders\_8)

> print(t\_test\_of\_cylinders\_4\_cylinders\_8)

Here in the above code, t.test (cylinders\_4, cylinders\_8) does the t-test of mpg between the cylinders\_4 and cylinders\_8 and stores the result in t\_test\_of\_cylinders\_4\_cylinders\_8.

print(t\_test\_of\_cylinders\_4\_cylinders\_8), prints the output of the respective t-test

The output is:

Welch Two Sample t-test

data: cylinders\_4 and cylinders\_8

t = 29.251, df = 299.73, p-value < 2.2e-16

alternative hypothesis: true difference in means is not equal to 0

95 percent confidence interval:

13.35737 15.28426

sample estimates:

mean of x mean of y

29.28392 14.96311

We can obtain the p-value of the above output by the code as a print(t\_test\_of\_cylinders\_4\_cylinder\_8$p.value)

So, by the above code, we get p-value as 8.086399e-90. The extremely small p-value indicates that there is, in fact, a highly apparent and substantial difference in mpg between these two sets of vehicles. In other words, 4-cylinder cars often have far higher fuel efficiency than 8-cylinder cars. From this t-test, we can also state that cylinders\_4 will get more than cylinders\_8.

Now we compare the mpg value between the cylinders\_4 and cylinders\_8 using the wilcox test as follows:

The ‘R’ code is as follows:

> cylinders\_4 <- required\_data[required\_data$cylinders == 4, "mpg"]

> cylinders\_8 <- required\_data[required\_data$cylinders == 8, "mpg"]

> wilcox\_test\_of\_cylinders\_4\_cylinders\_8 <- wilcox.test(cylinders\_4, cylinders\_8)

> print(wilcox\_test\_of\_cylinders\_4\_cylinders\_8)

Here in the above code, wilcox.test (cylinders\_4, cylinders\_8) does the wilcox test of mpg between the cylinders\_4 and cylinders\_8 and stores the result in wilcox\_test\_of\_cylinders\_4\_cylinders\_8.

print(wilcox\_test\_of\_cylinders\_4\_cylinders\_8), prints the output of the respective wilcox test

The output is:

Wilcoxon rank sum test with continuity correction

data: cylinders\_4 and cylinders\_8

W = 20338, p-value < 2.2e-16

alternative hypothesis: true location shift is not equal to 0

We can obtain the p-value of the above output by the code as print(wilcox\_test\_of\_cylinders\_4\_cylinder\_8$p.value)

So, by the above code, we get a p-value as 1.021299e-44. This indicates strong evidence that it is exceedingly unlikely that the observed mpg difference between cylinders\_4 and cylinders\_8 could have arisen from random variation. This test confirms that automobiles with 4 cylinders and automobiles with 8 cylinders have very different fuel efficiency.

Now we compare the mpg value between the cylinders\_6 and cylinders\_8 using the t-test as follows:

The ‘R’ code is as follows:

> cylinders\_6 <- required\_data[required\_data$cylinders == 6, "mpg"]

> cylinders\_8 <- required\_data[required\_data$cylinders == 8, "mpg"]

> t\_test\_of\_cylinders\_6\_cylinders\_8 <- t.test(cylinders\_6, cylinders\_8)

> print(t\_test\_of\_cylinders\_6\_cylinders\_8)

Here in the above code, t.test (cylinders\_6, cylinders\_8) does the t-test of mpg between the cylinders\_6 and cylinders\_8 and stores the result in t\_test\_of\_cylinders\_6\_cylinders\_8.

print(t\_test\_of\_cylinders\_6\_cylinders\_8), prints the output of the respective t-test

The output is:

Welch Two Sample t-test

data: cylinders\_6 and cylinders\_8

t = 9.9274, df = 147.39, p-value < 2.2e-16

alternative hypothesis: true difference in means is not equal to 0

95 percent confidence interval:

4.012997 6.007778

sample estimates:

mean of x mean of y

19.97349 14.96311

We can obtain the p-value of the above output by the code as a print(t\_test\_of\_cylinders\_6\_cylinder\_8$p.value)

So, by the above code, we get p-value as 4.208539e-18. The extremely small p-value indicates that there is, in fact, a highly apparent and substantial difference in mpg between the cars with cylinders\_6 and cylinders\_8. In other words, vehicles with 6 cylinders typically use substantially less fuel than vehicles with 8 cylinders. The confidence interval gives an estimate of how much more fuel-efficient vehicles with 6 cylinders operate when compared to vehicles with 8 cylinders.

Now we compare the mpg value between the cylinders\_6 and cylinders\_8 using the wilcox test as follows:

The ‘R’ code is as follows:

> cylinders\_6 <- required\_data[required\_data$cylinders == 6, "mpg"]

> cylinders\_8 <- required\_data[required\_data$cylinders == 8, "mpg"]

> wilcox\_test\_of\_cylinders\_6\_cylinders\_8 <- wilcox.test(cylinders\_6, cylinders\_8)

> print(wilcox\_test\_of\_cylinders\_6\_cylinders\_8)

Here in the above code, wilcox.test (cylinders\_6, cylinders\_8) does the wilcox test of mpg between the cylinders\_6 and cylinders\_8 and stores the result in wilcox\_test\_of\_cylinders\_6\_cylinders\_8.

print(wilcox\_test\_of\_cylinders\_6\_cylinders\_8), prints the output of the respective wilcox test

The output is:

Wilcoxon rank sum test with continuity correction

data: cylinders\_6 and cylinders\_8

W = 7656.5, p-value < 2.2e-16

alternative hypothesis: true location shift is not equal to 0

We can obtain the p-value of the above output by the code as print(wilcox\_test\_of\_cylinders\_6\_cylinder\_8$p.value)

So, by the above code, we get a p-value as 1.640441e-20. The extremely small p-value indicates that there is, in fact, a very substantial and discernible difference between these two groups of cars. To put it another way, they differ in terms of the attributes that are being measured. It is exceedingly unlikely that this outcome is the result of random chance. From this test, we can also state that cars with 6 cylinders get more mpg than cars with 8 cylinders.

Q3. You want to compare the performance of two machine learning algorithms. You do the following experiments:

- Collect 10 datasets

- Run each algorithm on those datasets. For each dataset, you conduct a 10-fold cross-validation and measure the average accuracy and running time.

How the result table look like (generate an example table)? How can you use t.test on that result table? Can you use a paired t.test?

To compute this question, I have set the path in my local R studio as follows:

>setwd("C:/Users/cscha/OneDrive/Desktop/Software analytics/Datasets")

To compute the above question I have collected 10 dataset samples from the following reference and placed them in above mentioned path.

The reference link is: <https://archive.ics.uci.edu/datasets>

The above question was computed using Random Forest and Support Vector Machine (SVM), two efficient machine learning algorithms. To perform the machine learning algorithms operations on collected datasets, we need to install the below packages in ‘R’ studio to compute on data sets.

> install.packages("caret")

> install.packages("randomForest")

> install.packages("e1071")

After installing the above packages in ‘r’ studio we need to load them in the current workspace by following the commands

>library(caret)

>library(randomForest)

>library(e1071)

Now we consider the first data set as dataset1 and do the following code on the dataset1:

>data <- read.csv(“dataset1.csv”)

This line reads the dataset1.csv file in the above-specified path and stores it in the data variable.

> X <- data[, 1:4]

>Y <- data[, 5]

> number\_of\_folds <- 10

> output <- data.frame(Algorithm = character(0), Accuracy = numeric(0))

>

> execute\_cross\_validation <- function(model, algorithm\_name) {

+ set.seed(123)

+ folds <- createFolds(Y, k = number\_of\_folds, list = TRUE)

+ accuracy <- numeric(number\_of\_folds)

+ runtimes <- numeric(number\_of\_folds)

+

+ for (i in 1:number\_of\_folds) {

+

+ train\_index <- unlist(folds[i])

+ test\_index <- setdiff(1:length(Y), train\_index)

+

+ train\_data <- X[train\_index, ]

+ train\_labels <- Y[train\_index]

+ train\_labels <- as.factor(train\_labels)

+ test\_data <- X[test\_index, ]

+ test\_labels <- Y[test\_index]

+

+ runtime <- system.time( {

+ model\_fit <- model(tarin\_data, tarin\_labels)

+ })[3]

+

+ model\_fit <- model(train\_data, train\_labels)

+ predictions <- predict(model\_fit, test\_data)

+ accuracy[i] <- sum(predictions == test\_labels) / length(test\_labels)

+ runtimes[i] <- runtime

+ }

+ mean\_accuracy <- mean(accuracy)

+ output <<- rbind(output, data.frame(datasetname="Dataset1", Algorithm = algorithm\_name, Accuracy = mean\_accuracy, Runtime = mean\_runtime))

+ }

> execute\_cross\_validation(

+ model = randomForest,

+ algorithm\_name = "Random Forest"

+ )

> execute\_cross\_validation(

+ model = svm,

+ algorithm\_name = "Support Vector Machine"

+ )

> print(output)

The output of the above code is:

datasetname Algorithm Accuracy Runtime

1 Dataset1 Random Forest 0.9333333 0.002

2 Dataset1 Support Vector Machine 0.9066667 0.001

The first four columns of dataset1, which are the feature columns (sepal length, sepal width, petal length, and petal width), are chosen by the expression X - data[, 1:4].

Y - data[, 5]: This line chooses the target variable (species), which is found in the fifth column of the dataset1.

Sets the number of folds for k-fold cross-validation to 10 in the cross-validation setup command.

Creates an empty data frame with the name "output" to hold the outcomes. frame(Algorithm = character(0), Accuracy = numeric(0)). The "Algorithm" column lists algorithm names, and the "Accuracy" column lists accuracy metrics. execute\_croos\_validation () takes two arguments: model (the machine learning algorithm) and algorithm\_name (the name of the algorithm). It applies the provided algorithm to k-fold cross-validation, with k in this example set to 10. It divides the data into test and training sets for each fold, trains the model on the training data, and determines the accuracy on the test data. The "output" data frame contains a calculation and storage of the mean accuracy over all folds. Executing Cross-Validation: The Random Forest algorithm's cross-validation is carried out using the execute\_cross\_validation (model = randomForest, algorithm\_name = "Random Forest") function, and the results are stored in the "output" data frame. Executes cross-validation for the Support Vector Machine algorithm and stores the results in the "output" data frame using the command execute\_cross\_validation (model = svm, algorithm\_name = "Support Vector Machine"). The algorithm names ("Random Forest" and "Support Vector Machine") and the related mean accuracy values on the dataset1 are now displayed in the "output" data frame, which may be viewed with the print(output) command.

From the above output, we can state that for dataset1 the accuracy is higher when we apply the Random Forest machine learning algorithm than the Support Vector Machine learning algorithm.

We follow the same above mentioned code for the rest of dataset such datset2,…. dataset9 which are stored in my local path as same as dateset1 location.

Now we bind the Random Forest accuracy average and Support Vector Machine average accuracy along with their respective runtimes for 10 datasets as below:

> dataset\_names <- c("Dataset1", "Dataset2", "Dataset3", "Dataset4", "Dataset5",

+ "Dataset6", "Dataset7", "Dataset8", "Dataset9", "Dataset10")

>

> random\_forest\_accuracy <- c(0.9333333,0.8971349,0.697071,0.8511495,0.638898, 0.5158129 , 0.5844227, 0.6329812, 0.6211, 0.5032657 )

> svm\_accuracy <- c(0.9066667, 0.8639264, 0.7722899,0.8726763,0.8896149, 0.7357 ,0.7381112, 0.7664932 ,0.8997312, 0.7318829 )

> random\_forest\_runtime <- c(0.002,0.007, 0.002,0.008,0.002, 0.004, 0.002, 0.008,0.002, 0.007 )

> svm\_runtime <- c(0.001, 005, 0.009,0.001, 0.003, 0.004,0.003,0.001, 0.006, 0.01 )

>

> results\_table <- data.frame(

+ Dataset = character(0),

+ Algorithm = character(0),

+ Accuracy = numeric(0),

+ RunTime = numeric(0)

+ )

>

> for (i in 1:length(dataset\_names)) {

+ dataset\_name <- dataset\_names[i]

+ random\_forest\_acc <- random\_forest\_accuracy[i]

+ svm\_acc <- svm\_accuracy[i]

+ rtime <- random\_forest\_runtime[i]

+ stime<- svm\_runtime[i]

+ results\_table <- rbind(results\_table, data.frame(

+ Dataset = dataset\_name,

+ Algorithm = "Random Forest",

+ Accuracy = random\_forest\_acc,

+ RunTime = rtime

+ ))

+

+ results\_table <- rbind(results\_table, data.frame(

+ Dataset = dataset\_name,

+ Algorithm = "Support Vector Machine",

+ Accuracy = svm\_acc,

+ RunTime = stime

+ ))

+

+ }

>print(results\_table)

The output is:

Dataset Algorithm Accuracy RunTime

1 Dataset1 Random Forest 0.9333333 0.002

2 Dataset1 Support Vector Machine 0.9066667 0.001

3 Dataset2 Random Forest 0.8971349 0.007

4 Dataset2 Support Vector Machine 0.8639264 0.005

5 Dataset3 Random Forest 0.6970710 0.002

6 Dataset3 Support Vector Machine 0.7722899 0.009

7 Dataset4 Random Forest 0.8511495 0.008

8 Dataset4 Support Vector Machine 0.8726763 0.001

9 Dataset5 Random Forest 0.6388980 0.002

10 Dataset5 Support Vector Machine 0.8896149 0.003

11 Dataset6 Random Forest 0.5158129 0.004

12 Dataset6 Support Vector Machine 0.7357000 0.004

13 Dataset7 Random Forest 0.5844227 0.002

14 Dataset7 Support Vector Machine 0.7381112 0.003

15 Dataset8 Random Forest 0.6329812 0.008

16 Dataset8 Support Vector Machine 0.7664932 0.001

17 Dataset9 Random Forest 0.6211000 0.002

18 Dataset9 Support Vector Machine 0.8997312 0.006

19 Dataset10 Random Forest 0.5032657 0.007

20 Dataset10 Support Vector Machine 0.7318829 0.010

To calculate the t-test between the accuracy of two machine learning algorithms we write the below ‘R’ code:

>t\_test\_result <- t.test(random\_forest\_accuracy, svm\_accuracy, paired=TRUE)

>print(t\_test\_result)

The output is:

Paired t-test

data: random\_forest\_accuracy and svm\_accuracy

t = -3.5508, df = 9, p-value = 0.006208

alternative hypothesis: true mean difference is not equal to 0

95 percent confidence interval:

-0.21313496 -0.04724974

sample estimates:

mean difference

-0.1301923

From the above p-value of 0.006208 we can state that between Random Forest and Support Vector Machine, there is an accuracy gap that is statistically significant. In this instance, the mean accuracy of Random Forest is lower than the mean accuracy of the Support Vector Machine, which is a substantial difference.

To calculate the t-test between the run time of two machine learning algorithms we write the below ‘R’ code:

> t\_test\_result <- t.test(random\_forest\_runtime, svm\_runtime, paired=TRUE)

> print(t\_test\_result)

The output is:

Paired t-test

data: random\_forest\_runtime and svm\_runtime

t = -1, df = 9, p-value = 0.3434

alternative hypothesis: true mean difference is not equal to 0

95 percent confidence interval:

-1923561.6 744242.8

sample estimates:

mean difference

-589659.4

A measure of the evidence against a null hypothesis is the p-value. When there is no real difference between the two groups, it indicates the likelihood of seeing a t-statistic that is as extreme as the one calculated. The p-value in this instance is rather high at 0.3434, indicating that there may not be a statistically significant difference between the two algorithms' runtimes.