

## Across Datasets - 4 trials were done

```
In [1]: from scipy import stats
import numpy as np
import pandas as pd
```

```
In [2]: #List of each accuracy

KNN_accuracy = [0.8196, 0.8187, 0.8146, 0.8172, 0.7914, 0.7926, 0.7873
, 0.7967, 0.9167, 0.9078, 0.9395, 0.9165]
RF_accuracy = [0.8086, 0.8119, 0.8125, 0.8094, 0.8308, 0.8307, 0.8342,
0.8322, 0.9447, 0.9422, 0.9602, 0.9439]
LG_accuracy = [0.8220, 0.8219, 0.8196, 0.8261, 0.7811, 0.7831, 0.7837,
0.7795, 0.7236, 0.7244, 0.7287, 0.7237]
```

```
In [3]: #T-Test comparing each algorithm to one another

KNN_RF = stats.ttest_ind(KNN_accuracy, RF_accuracy)
RF_LG = stats.ttest_ind(RF_accuracy, LG_accuracy)
KNN_LG = stats.ttest_ind(KNN_accuracy, LG_accuracy)
```

```
In [4]: #Print T-Test & P-Test results

print("KNN and Random Forest ", KNN_RF)
print("Random Forest and KNN ", RF_LG)
print("KNN and Logistic Regression ", KNN_LG)

KNN and Random Forest  Ttest_indResult(statistic=-0.815482178467262,
pvalue=0.42353768020683913)
Random Forest and KNN  Ttest_indResult(statistic=3.983516448907488,
pvalue=0.0006278668670921057)
KNN and Logistic Regression  Ttest_indResult(statistic=3.22534741446
69645, pvalue=0.0038923378245222687)
```

```
In [5]: #Mean of each algorithm across trials in each dataset into a list

mean_KNN = [0.817525, 0.792, 0.920125]
mean_RF = [0.8106, 0.831975, 0.94775]
mean_LG = [0.8224, 0.781725, 0.7251]
```

```
In [6]: #T-Tests comparing the means of each algorithm to one another
```

```
KNN_RF_mean = stats.ttest_ind(mean_KNN,mean_RF)
RF_LG_mean = stats.ttest_ind(mean_RF,mean_LG)
KNN_LG_mean = stats.ttest_ind(mean_KNN,mean_LG)
```

```
In [7]: #Print T-Test & P-Test results for the mean
```

```
print("KNN and Random Forest Mean ", KNN_RF_mean)
print("Random Forest and KNN Mean ", RF_LG_mean)
print("KNN and Logistic Regression Mean ", KNN_LG_mean)
```

```
KNN and Random Forest Mean  Ttest_indResult(statistic=-0.34953575523
76778, pvalue=0.7443135646124233)
Random Forest and KNN Mean  Ttest_indResult(statistic=1.703246718484
86, pvalue=0.1637308641458689)
KNN and Logistic Regression Mean  Ttest_indResult(statistic=1.384342
7389987122, pvalue=0.23847388981768075)
```

```
In [8]: #Mean of each algorithm across all datasets
```

```
KNN_total_mean = np.mean(mean_KNN)
RF_total_mean = np.mean(mean_RF)
LG_total_mean = np.mean(mean_LG)

print("KNN Total Mean ", KNN_total_mean)
print("Random Forest Total Mean ", RF_total_mean)
print("Logistic Regression Total Mean ", LG_total_mean)
```

```
KNN Total Mean  0.8432166666666667
Random Forest Total Mean  0.8634416666666667
Logistic Regression Total Mean  0.7764083333333334
```

## For Each Dataset - 4 trials were done

### Adult

```
In [9]: #List of each accuracy
```

```
KNN_accuracy_adult = [0.8196, 0.8187, 0.8146, 0.8172]
RF_accuracy_adult = [0.8086, 0.8119, 0.8125, 0.8094]
LG_accuracy_adult = [0.8220, 0.8219, 0.8196, 0.8261]
```

```
In [10]: #T-Test comparing each algorithm to one another in the adult dataset

KNN_RF_adult = stats.ttest_ind(KNN_accuracy_adult,RF_accuracy_adult)
RF_LG_adult = stats.ttest_ind(RF_accuracy_adult,LG_accuracy_adult)
KNN_LG_adult = stats.ttest_ind(KNN_accuracy_adult,LG_accuracy_adult)
```

```
In [11]: #Print T-Test & P-Test results

print("KNN and Random Forest ", KNN_RF_adult)
print("Random Forest and KNN ", RF_LG_adult)
print("KNN and Logistic Regression ", KNN_LG_adult)

KNN and Random Forest  Ttest_indResult(statistic=4.789406596394901,
pvalue=0.0030334549284151504)
Random Forest and KNN  Ttest_indResult(statistic=-7.150420894733969,
pvalue=0.00037725845734914857)
KNN and Logistic Regression  Ttest_indResult(statistic=-2.8034106485
1531, pvalue=0.03102422525511558)
```

```
In [12]: #Mean of each algorithm across trials in each dataset into a list

mean_KNN_adult = [0.817525]
mean_RF_adult = [0.8106]
mean_LG_adult = [0.8224]
```

```
In [13]: #T-Tests comparing the means of each algorithm to one another

KNN_RF_mean_adult = stats.ttest_ind(mean_KNN_adult,mean_RF_adult)
RF_LG_mean_adult = stats.ttest_ind(mean_RF_adult,mean_LG_adult)
KNN_LG_mean_adult = stats.ttest_ind(mean_KNN_adult,mean_LG_adult)

/Users/adriannahohil/anaconda3/lib/python3.7/site-packages/numpy/core/
fromnumeric.py:3157: RuntimeWarning: Degrees of freedom <= 0 for s
lice
    **kwargs)
/Users/adriannahohil/anaconda3/lib/python3.7/site-packages/numpy/core/
_methods.py:132: RuntimeWarning: invalid value encountered in doub
le_scalars
    ret = ret.dtype.type(ret / rcount)
```

```
In [14]: #Print T-Test & P-Test results for the mean

print("KNN and Random Forest Mean ", KNN_RF_mean_adult)
print("Random Forest and Logistic Regression Mean ", RF_LG_mean_adult)
print("KNN and Logistic Regression Mean ", KNN_LG_mean_adult)

KNN and Random Forest Mean  Ttest_indResult(statistic=nan, pvalue=nan)
Random Forest and Logistic Regression Mean  Ttest_indResult(statistic=nan, pvalue=nan)
KNN and Logistic Regression Mean  Ttest_indResult(statistic=nan, pvalue=nan)
```

```
In [15]: #Mean of each algorithm in the dataset

KNN_total_mean_adult = np.mean(mean_KNN_adult)
RF_total_mean_adult = np.mean(mean_RF_adult)
LG_total_mean_adult = np.mean(mean_LG_adult)

print("KNN Total Mean ", KNN_total_mean_adult)
print("Random Forest Total Mean ", RF_total_mean_adult)
print("Logistic Regression Total Mean ", LG_total_mean_adult)

KNN Total Mean  0.817525
Random Forest Total Mean  0.8106
Logistic Regression Total Mean  0.8224
```

## Cov\_Type

```
In [16]: #List of each accuracy

KNN_accuracy_cov_type = [0.7914, 0.7926, 0.7873, 0.7967]
RF_accuracy_cov_type = [0.8308, 0.8307, 0.8342, 0.8322]
LG_accuracy_cov_type = [0.7811, 0.7831, 0.7837, 0.7795]

In [17]: #T-Test comparing each algorithm to one another in the cov type dataset

KNN_RF_cov_type = stats.ttest_ind(KNN_accuracy_cov_type, RF_accuracy_cov_type)
RF_LG_cov_type = stats.ttest_ind(RF_accuracy_cov_type, LG_accuracy_cov_type)
KNN_LG_cov_type = stats.ttest_ind(KNN_accuracy_cov_type, LG_accuracy_cov_type)
```

```
In [18]: #Print T-Test & P-Test results

print("KNN and Random Forest ", KNN_RF_cov_type)
print("Random Forest and Logistic Regression ", RF_LG_cov_type)
print("Logistic Regression Total Mean ", KNN_LG_cov_type)

KNN and Random Forest  Ttest_indResult(statistic=-19.037962855159524
, pvalue=1.3578735238410088e-06)
Random Forest and Logistic Regression  Ttest_indResult(statistic=39.
75431526881637, pvalue=1.6930820589346116e-08)
Logistic Regression Total Mean  Ttest_indResult(statistic=4.69979604
8931753, pvalue=0.0033268653390252346)
```

```
In [19]: #Mean of each algorithm across trials in each dataset into a list

mean_KNN_cov_type = [0.792]
mean_RF_cov_type = [0.831975]
mean_LG_cov_type = [0.781725]
```

```
In [20]: #T-Tests comparing the means of each algorithm to one another

KNN_RF_mean_cov_type = stats.ttest_ind(mean_KNN_cov_type,mean_RF_cov_t
ype)
RF_LG_mean_cov_type = stats.ttest_ind(mean_RF_cov_type,mean_LG_cov_ty
pe)
KNN_LG_mean_cov_type = stats.ttest_ind(mean_KNN_cov_type,mean_LG_cov_t
ype)
```

```
In [21]: #Print T-Test & P-Test results for the mean

print("KNN and Random Forest Mean ", KNN_RF_mean_cov_type)
print("Random Forest and Logistic Regression Mean ", RF_LG_mean_cov_ty
pe)
print("KNN and Logistic Regression Mean ", KNN_LG_mean_cov_type)

KNN and Random Forest Mean  Ttest_indResult(statistic=nan, pvalue=na
n)
Random Forest and Logistic Regression Mean  Ttest_indResult(statisti
c=nan, pvalue=nan)
KNN and Logistic Regression Mean  Ttest_indResult(statistic=nan, pva
lue=nan)
```

```
In [22]: #Mean of each algorithm in the dataset

KNN_total_mean_cov_type = np.mean(mean_KNN_cov_type)
RF_total_mean_cov_type = np.mean(mean_RF_cov_type)
LG_total_mean_cov_type = np.mean(mean_LG_cov_type)

print("KNN Total Mean ", KNN_total_mean_cov_type)
print("Random Forest Total Mean ", RF_total_mean_cov_type)
print("KNN and Logistic Regression ", LG_total_mean_cov_type)
```

KNN Total Mean 0.792  
Random Forest Total Mean 0.831975  
KNN and Logistic Regression 0.781725

## Letters

```
In [23]: #List of each accuracy

KNN_accuracy_letters = [0.9167, 0.9078, 0.9395, 0.9165]
RF_accuracy_letters = [0.9447, 0.9422, 0.9602, 0.9439]
LG_accuracy_letters = [0.7236, 0.7244, 0.7287, 0.7237]
```

```
In [24]: #T-Test comparing each algorithm to one another in the cov type dataset

KNN_RF_letters = stats.ttest_ind(KNN_accuracy_letters, RF_accuracy_letters)
RF_LG_letters = stats.ttest_ind(RF_accuracy_letters, LG_accuracy_letters)
KNN_LG_letters = stats.ttest_ind(KNN_accuracy_letters, LG_accuracy_letters)
```

```
In [25]: #Print T-Test & P-Test results

print("KNN and Random Forest ", KNN_RF_letters)
print("Random Forest and KNN ", RF_LG_letters)
print("KNN and Logistic Regression ", KNN_LG_letters)
```

KNN and Random Forest Ttest\_indResult(statistic=-3.466471558187397, pvalue=0.013361234834874312)  
Random Forest and KNN Ttest\_indResult(statistic=51.1254018451443, pvalue=3.757243689210903e-09)  
KNN and Logistic Regression Ttest\_indResult(statistic=28.301483920974867, pvalue=1.2880709498426782e-07)

In [26]: *#Mean of each algorithm across trials in each dataset into a list*

```
mean_KNN_letters = [0.920125]
mean_RF_letters = [0.94775]
mean_LG_letters = [0.7251]
```

In [27]: *#T-Tests comparing the means of each algorithm to one another*

```
KNN_RF_mean_letters = stats.ttest_ind(mean_KNN_letters,mean_RF_letters)
RF_LG_mean_letters = stats.ttest_ind(mean_RF_letters,mean_LG_letters)
KNN_LG_mean_letters = stats.ttest_ind(mean_KNN_letters,mean_LG_letters)
```

In [28]: *#Print T-Test & P-Test results for the mean*

```
print("KNN and Random Forest Mean ", KNN_RF_mean_letters)
print("Random Forest and Logistic Regression Mean ", RF_LG_mean_letters)
print("KNN and Logistic Regression Mean ", KNN_LG_mean_letters)
```

```
KNN and Random Forest Mean  Ttest_indResult(statistic=nan, pvalue=nan)
Random Forest and Logistic Regression Mean  Ttest_indResult(statistic=nan, pvalue=nan)
KNN and Logistic Regression Mean  Ttest_indResult(statistic=nan, pvalue=nan)
```

In [29]: *#Mean of each algorithm in the dataset*

```
KNN_total_mean_letters = np.mean(mean_KNN_letters)
RF_total_mean_letters = np.mean(mean_RF_letters)
LG_total_mean_letters = np.mean(mean_LG_letters)

print("KNN Total Mean ", KNN_total_mean_letters)
print("Random Forest Total Mean ", RF_total_mean_letters)
print("Logistic Regression Total Mean ", LG_total_mean_letters)
```

```
KNN Total Mean  0.920125
Random Forest Total Mean  0.94775
Logistic Regression Total Mean  0.7251
```