```
from scipy import stats


# Response time of the 3 predictor models
# Logistic Regression, Naïve Bayes Classifier, vs Decision Tree

model_lr = [2.50, 2.53, 3.15, 3.02]
model_nb = [0.33, 0.42, 0.76, 0.41]
model_dt = [4.37, 3.28, 4.12, 3.42]


# Calculate means
print("Mean LR: ", sum(model_lr)/len(model_lr))
print("Mean NB: ", sum(model_nb)/len(model_nb))
print("Mean DT: ", sum(model_dt)/len(model_dt))
```

```
Mean LR:  2.8
Mean NB:  0.48
Mean DT:  3.7975
```

```
# Friedman Test to test between the 3 predictor models
stats.friedmanchisquare(model_lr, model_nb, model_dt)
```

```
FriedmanchisquareResult(statistic=8.0, pvalue=0.018315638888734182)
```

```
# Can reject null hypothesis - there is a significant difference between the mean time
# It takes to run Logistic Regression, Naïve Bayes Classifier, and Decision Tree


# Response time of the 4 train sample sets
# Unbalanced, Balanced Random Under-Sampling, Balanced SMOTE Oversampling, Balanced SMOTETome

unb = [2.50, 0.33, 4.37]
bal7 = [2.53, 0.42, 3.28]
bal8 = [3.15, 0.76, 4.12]
bal9 = [3.02, 0.41, 3.42]


# Calculate means
print("Mean Unbalanced: ", sum(unb)/len(unb))
print("Mean Bal7: ", sum(bal7)/len(bal7))
print("Mean Bal8: ", sum(bal8)/len(bal8))
print("Mean Bal9: ", sum(bal9)/len(bal9))
```

```
Mean Unbalanced:  2.4
Mean Bal7:  2.0766666666666667
Mean Bal8:  2.676666666666667
Mean Bal9:  2.283333333333333
```

```
# Friedman Test to test between the 3 predictor models
```

```
stats.friedmanchisquare(unb, bal7, bal8, bal9)
```

```
FriedmanchisquareResult(statistic=3.4000000000000057, pvalue=0.33396524909015995)
```

```
# Stability Test - Cross Validation Accuracy from code file 7
cv_lr = [1, 1, 1, 1, 1, 1, 1, 1, 0.95723014, 0.95918367]
cv_nb = [0.99796334, 0.99592668, 0.99796334, 0.99796334, 0.99592668, 0.99389002, 0.99592668,
cv_dt = [1, 1, 1, 1, 1, 1, 1, 1, 1, 1]

# Calculate means
print("Mean cv_lr: ", sum(cv_lr)/len(cv_lr))
print("Mean cv_nb: ", sum(cv_nb)/len(cv_nb))
print("Mean cv_dt: ", sum(cv_dt)/len(cv_dt))

# Friedman Test to test between the 3 predictor models
stats.friedmanchisquare(cv_lr, cv_nb, cv_dt)
```

```
Mean cv_lr:  0.991641381
Mean cv_nb:  0.996944594
Mean cv_dt:  1.0
FriedmanchisquareResult(statistic=13.000000000000007, pvalue=0.0015034391929775667)
```

```
# Stability Test - Cross Validation Accuracy from code file 8
cv_lr = [1, 1, 1, 1, 1, 1, 1, 1, 0.95723014, 0.95918367]
cv_nb = [0.99796334, 0.99592668, 0.99796334, 0.99796334, 0.99592668, 0.99389002, 0.99592668,
cv_dt = [1, 1, 1, 1, 1, 1, 1, 1, 1, 1]

# Calculate means
print("Mean cv_lr: ", sum(cv_lr)/len(cv_lr))
print("Mean cv_nb: ", sum(cv_nb)/len(cv_nb))
print("Mean cv_dt: ", sum(cv_dt)/len(cv_dt))

# Friedman Test to test between the 3 predictor models
stats.friedmanchisquare(cv_lr, cv_nb, cv_dt)
```

```
Mean cv_lr:  0.991641381
Mean cv_nb:  0.996944594
Mean cv_dt:  1.0
FriedmanchisquareResult(statistic=13.000000000000007, pvalue=0.0015034391929775667)
```

```
# Stability Test - Cross Validation Accuracy from code file 9
cv_lr = [1, 1, 1, 1, 1, 1, 1, 1, 0.95723014, 0.95918367]
cv_nb = [0.99796334, 0.99592668, 0.99796334, 0.99796334, 0.99592668, 0.99389002, 0.99592668,
cv_dt = [1, 1, 1, 1, 1, 1, 1, 1, 1, 1]


# Calculate means
print("Mean cv_lr: ", sum(cv_lr)/len(cv_lr))
```

```
print("Mean cv_nb: ", sum(cv_nb)/len(cv_nb))
print("Mean cv_dt: ", sum(cv_dt)/len(cv_dt))
```

```
Mean cv_lr:  0.991641381
Mean cv_nb:  0.996944594
Mean cv_dt:  1.0
```

```
# Friedman Test to test between the 3 predictor models
stats.friedmanchisquare(cv_lr, cv_nb, cv_dt)
```

```
FriedmanchisquareResult(statistic=13.000000000000007, pvalue=0.0015034391929775667)
```