

```
In [26]: import pandas as pd

#HERE I AM IMPORTING MY ALGORITHMS
from sklearn.linear_model import LogisticRegression, ElasticNetCV
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
```

```
In [27]: #LET US READ OUR CLEANED TRAIN AND TEST DATASETS INTO THIS NOTEBOOK

train = pd.read_csv('CLEANED_TRAIN.csv')
test = pd.read_csv('CLEANED_TEST.csv')
```

```
In [28]: #Overview of the first 5 rows of the train dataset (Microsoft Excel can be used for this)

train.head()
```

Out[28]:

	Survived	Pclass	Age	SibSp	Parch	Fare	male	Q	S
0	0	3	22.0	1	0	7.2500	1	0	1
1	1	1	38.0	1	0	71.2833	0	0	0
2	1	3	26.0	0	0	7.9250	0	0	1
3	1	1	35.0	1	0	53.1000	0	0	1
4	0	3	35.0	0	0	8.0500	1	0	1

```
In [29]: #Overview of the first 5 rows of the test dataset (Microsoft Excel can be used for this)

test.head()
```

Out[29]:

	PassengerId	Pclass	Age	SibSp	Parch	Fare	male	Q	S
0	892	3	34.5	0	0	7.8292	1	1	0
1	893	3	47.0	1	0	7.0000	0	0	1
2	894	2	62.0	0	0	9.6875	1	1	0
3	895	3	27.0	0	0	8.6625	1	0	1
4	896	3	22.0	1	1	12.2875	0	0	1

```
In [30]: test.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 418 entries, 0 to 417
Data columns (total 9 columns):
PassengerId    418 non-null int64
Pclass         418 non-null int64
Age            418 non-null float64
SibSp          418 non-null int64
Parch          418 non-null int64
Fare           418 non-null float64
male           418 non-null int64
Q              418 non-null int64
S              418 non-null int64
dtypes: float64(2), int64(7)
memory usage: 29.5 KB
```

```
In [31]: #LET US ASSIGN OUR X and y variables

X = train.drop('Survived', axis = 1)

y = train['Survived']

#LET US DROP THE PASSENGER ID COLUMN FROM TEST AND ASSIGN IT TO ID
test_copy = test.drop('PassengerId', axis =1)

ID = test['PassengerId']
```

NOW LET US ASSIGN VARIABLES TO THE THREE ALGORITHMS WE WILL BE TRAINING AND USING TO PREDICT

```
In [32]: linearmodel = LogisticRegression()
treemodel = DecisionTreeClassifier()
forestmodel = RandomForestClassifier()
```

NOW LET US FIT EACH ALGORITHM AND PREDICT ON THE TEST DATA

In [33]: *#Fitting of each algorithm with X and y*

```
linearmodel.fit(X, y)
```

```
treemodel.fit(X, y)
```

```
forestmodel.fit(X, y)
```

C:\Users\Adeoluwa Adeboye\Anaconda3\lib\site-packages\sklearn\linear\_model\logistic.py:432: FutureWarning: Default solver will be changed to 'lbfgs' in 0.22. Specify a solver to silence this warning.

FutureWarning)

C:\Users\Adeoluwa Adeboye\Anaconda3\lib\site-packages\sklearn\ensemble\forest.py:245: FutureWarning: The default value of n\_estimators will change from 10 in version 0.20 to 100 in 0.22.

"10 in version 0.20 to 100 in 0.22.", FutureWarning)

Out[33]: RandomForestClassifier(bootstrap=True, class\_weight=None, criterion='gini', max\_depth=None, max\_features='auto', max\_leaf\_nodes=None, min\_impurity\_decrease=0.0, min\_impurity\_split=None, min\_samples\_leaf=1, min\_samples\_split=2, min\_weight\_fraction\_leaf=0.0, n\_estimators=10, n\_jobs=None, oob\_score=False, random\_state=None, verbose=0, warm\_start=False)

In [34]: *#Let us predict on our test dataset*

```
linear_prediction = linearmodel.predict(test_copy)
```

```
tree_prediction = treemodel.predict(test_copy)
```

```
forest_prediction = forestmodel.predict(test_copy)
```

HERE WE PUT OUT PREDICTION ON THE TEST DATA INTO A CSV FORMAT FOR SUBMISSION ON KAGGLE

```
In [35]: #THE CSV FORMAT MUST BE OF SIMILAR FORMAT WITH THE FORMAT STATED ON THE COMPETITION PAGE

#outputting a submission file fpr the linear prediction
linear_output = pd.DataFrame({'PassengerId' : ID, 'Survived' : linear_prediction})

linear_output.to_csv('linear_prediction.csv', index=False)

#outputting a submission file for the tree prediction
tree_output = pd.DataFrame({'PassengerId' : ID, 'Survived' : tree_prediction})

tree_output.to_csv('tree_prediction.csv', index=False)

#outputting a submission file for the forest prediction
forest_output = pd.DataFrame({'PassengerId' : ID, 'Survived' : forest_prediction})

forest_output.to_csv('forest_prediction.csv', index=False)
```

NOW LET US SEE HOW EACH OF OUR PREDICTION SCORE ON THE LEADERBOARD

In [ ]: