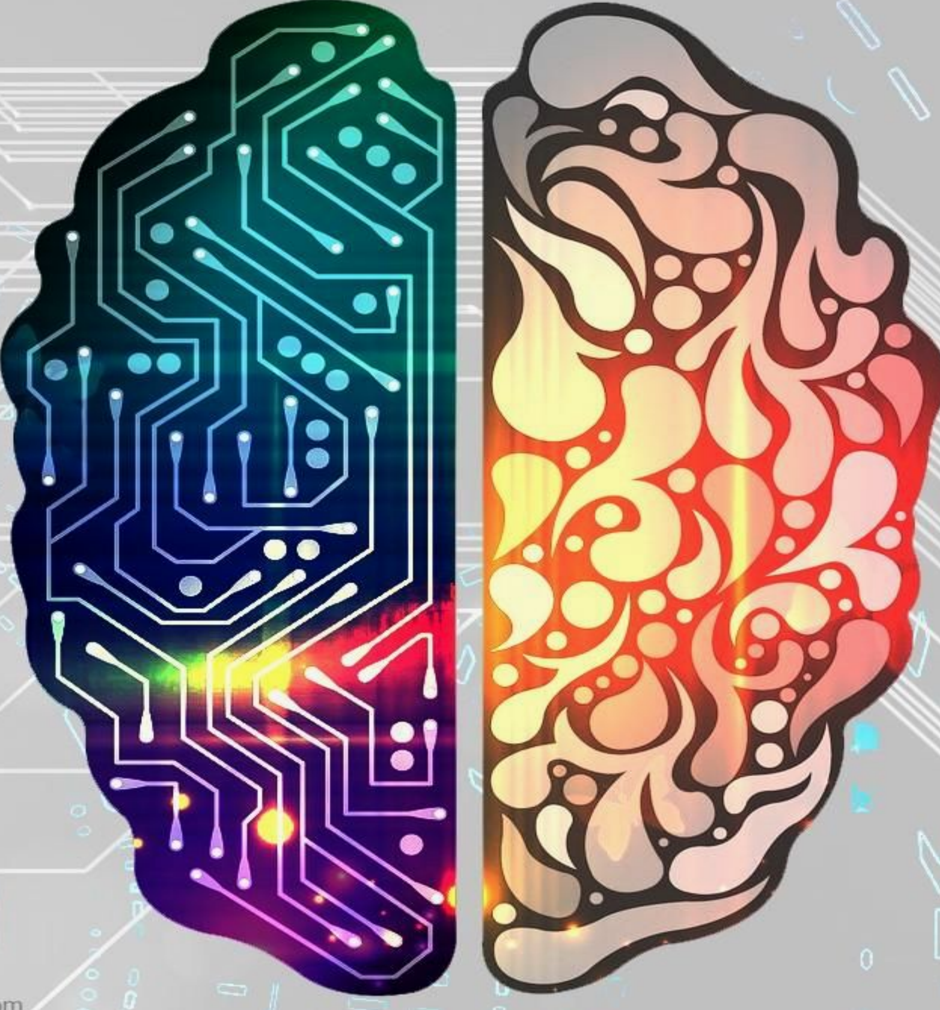
# MACHINE LEARNING

*Project Presentation*

*DS105*

Clement Leo



# *Table of Contents*

# *Dataset*

This dataset is on the unsinkable ship, RMS Titanic.

Size of dataset: 12 columns, 891 entries

Some of the features:

- Survived, Passenger Class, Sex, Age

Source: https://www.kaggle.com/competitions/titanic

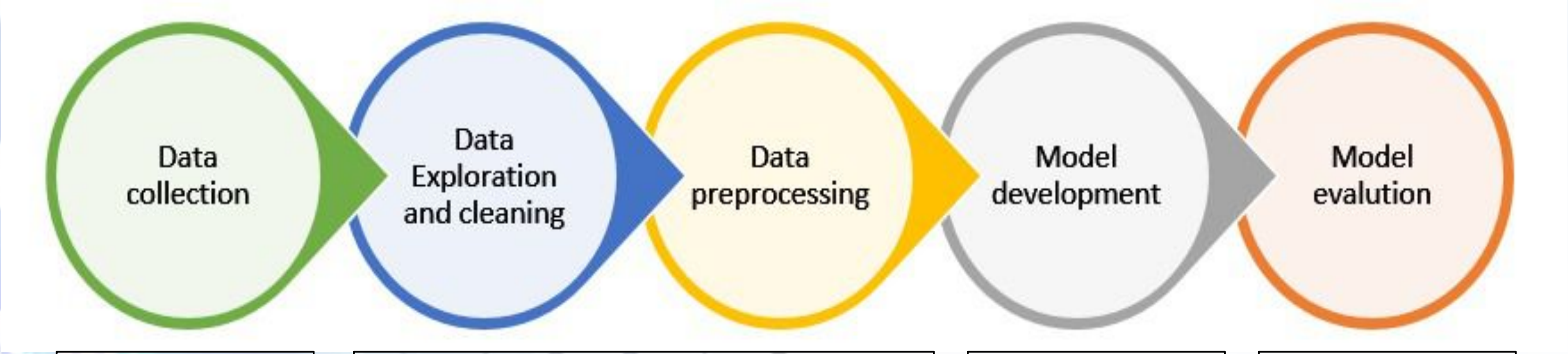
# *Goal*

To predict the survivability of a person on the ship

using ML models and to see which would be the best model to use.

| Passenger Id | Survived | Pclass | Name | Sex | Age | Sib Sp | Parch | Ticket | Fare | Cabin | Embarked |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 3 | Braund, Mr. Owe | male | 22 | 1 | 0 | A/5 21171 | 7.25 | | S |
| 2 | 1 | 1 | Cumings, Mrs. Jo | female | 38 | 1 | 0 | PC 17599 | 71.28 | C85 | C |
| 3 | 1 | 3 | Heikkinen, Miss. | female | 26 | 0 | 0 | STON/O2. 31012 | 7.93 | | S |
| 4 | 1 | 1 | Futrelle, Mrs. Jac | female | 35 | 1 | 0 | 113803 | 53.1 | C123 | S |
| 5 | 0 | 3 | Allen, Mr. William | male | 35 | 0 | 0 | 373450 | 8.05 | | S |
| 6 | 0 | 3 | Moran, Mr. Jame | male | | 0 | 0 | 330877 | 8.46 | | Q |
| 7 | 0 | 1 | McCarthy, Mr. Ti | male | 54 | 0 | 0 | 17463 | 51.86 | E46 | S |
| 8 | 0 | 3 | Palsson, Master. | male | 2 | 3 | 1 | 349909 | 21.08 | | S |
| 9 | 1 | 3 | Johnson, Mrs. Os | female | 27 | 0 | 2 | 347742 | 11.13 | | S |
| 10 | 1 | 2 | Nasser, Mrs. Nich | female | 14 | 1 | 0 | 237736 | 30.07 | | C |
| 11 | 1 | 3 | Sandstrom, Miss. | female | 4 | 1 | 1 | PP 9549 | 16.7 | G6 | S |
| 12 | 1 | 1 | Bonnell, Miss. Eli | female | 58 | 0 | 0 | 113783 | 26.55 | C103 | S |
| 13 | 0 | 3 | Saundercock, Mr | male | 20 | 0 | 0 | A/5. 2151 | 8.05 | | S |
| 14 | 0 | 3 | Andersson, Mr. A | male | 39 | 1 | 5 | 347082 | 31.28 | | S |
| 15 | 0 | 3 | Vestrom, Miss. H | female | 14 | 0 | 0 | 350406 | 7.85 | | S |
| 16 | 1 | 2 | Hewlett, Mrs. (Ma | female | 55 | 0 | 0 | 248706 | 16 | | S |
| 17 | 0 | 3 | Rice, Master. Eug | male | 2 | 4 | 1 | 382652 | 29.13 | | Q |
| 18 | 1 | 2 | Williams, Mr. Cha | male | | 0 | 0 | 244373 | 13 | | S |
| 19 | 0 | 3 | Vander Planke, M | female | 31 | 1 | 0 | 345763 | 18 | | S |
| 20 | 1 | 3 | Masselmani, Mrs | female | | 0 | 0 | 2649 | 7.23 | | C |
| 21 | 0 | 2 | Fynney, Mr. Jose | male | 35 | 0 | 0 | 239865 | 26 | | S |
| 22 | 1 | 2 | Beesley, Mr. Law | male | 34 | 0 | 0 | 248698 | 13 | D56 | S |
| 23 | 1 | 3 | McGowan, Miss. | female | 15 | 0 | 0 | 330923 | 8.03 | | Q |
| 24 | 1 | 1 | Sloper, Mr. Willia | male | 28 | 0 | 0 | 113788 | 35.5 | A6 | S |
| 25 | 0 | 3 | Palsson, Miss. To | female | 8 | 3 | 1 | 349909 | 21.08 | | S |
| 26 | 1 | 3 | Asplund, Mrs. Ca | female | 38 | 1 | 5 | 347077 | 31.39 | | S |
| 27 | 0 | 3 | Emir, Mr. Farred | male | | 0 | 0 | 2631 | 7.23 | | C |
| 28 | 0 | 1 | Fortune, Mr. Cha | male | 19 | 3 | 2 | 19950 | 263 | C23 C25 C27 | S |
| 29 | 1 | 3 | O'Dwyer, Miss. El | female | | 0 | 0 | 330959 | 7.88 | | Q |
| 30 | 0 | 3 | Todoroff, Mr. Lali | male | | 0 | 0 | 349216 | 7.9 | | S |
| 31 | 0 | 1 | Uruchurtu, Don. | male | 40 | 0 | 0 | PC 17601 | 27.72 | | C |
| 32 | 1 | 1 | Spencer, Mrs. Wi | female | | 1 | 0 | PC 17569 | 146.52 | B78 | C |
| 33 | 1 | 3 | Glynn, Miss. Mar | female | | 0 | 0 | 335677 | 7.75 | | Q |
| 34 | 0 | 2 | Wheadon, Mr. Ed | male | 66 | 0 | 0 | C.A. 24579 | 10.5 | | S |
| 35 | 0 | 1 | Meyer, Mr. Edgar | male | 28 | 1 | 0 | PC 17604 | 82.17 | | C |
| 36 | 0 | 1 | Holverson, Mr. A | male | 42 | 1 | 0 | 113789 | 52 | | S |
| 37 | 1 | 3 | Mamee, Mr. Han | male | | 0 | 0 | 2677 | 7.23 | | C |
| 38 | 0 | 3 | Cann, Mr. Ernest | male | 21 | 0 | 0 | A./5. 2152 | 8.05 | | S |
| 39 | 0 | 3 | Vander Planke, M | female | 18 | 2 | 0 | 345764 | 18 | | S |
| 40 | 1 | 3 | Nicola-Yarred, M | female | 14 | 1 | 0 | 2651 | 11.24 | | C |
| 41 | 0 | 3 | Ahlin, Mrs. Johan | female | 40 | 1 | 0 | 7546 | 9.48 | | S |
| 42 | 0 | 2 | Turpin, Mrs. Willi | female | 27 | 1 | 0 | 11668 | 21 | | S |

# *General Workflow*




| | | | |
|---|---|---|---|
| Titanic Dataset | Cleaning up, Preprocess & Insights | Use different algorithms | Compare the models |

# *Preprocessing & Cleaning*

```
1  #💡Null Checking
2  df_train.isnull().sum()
✓ 0.4s

PassengerId      0
Survived         0
Pclass           0
Name             0
Sex              0
Age            177
SibSp            0
Parch            0
Ticket           0
Fare             0
Cabin          687
Embarked         2
dtype: int64
```

There are a total of 891 rows, 12 Columns.

However there are missing values Age - 714 (177 Missing) Cabin - 204 (687 Missing) Embarked - 889 (2 Missing)

## Missing/NaN Values

3 features with null values.

Age ~20%

Cabin ~77%

Embarked ~0%

## What I did

Age - Imputed median

Cabin - Totally dropped

Embarked - Input the most used value, ‘S’

# *Preprocessing & Cleaning*

```python
# Combine Sibsp & ParCh together as Companions, since these are there together as a group

df_train['Companion'] = df_train['SibSp'] + df_train['Parch']
```

✓ 0.3s

```python
# Drop the SibSp and Parch columns

cols = ['SibSp', 'Parch']
df_train = df_train.drop(cols, axis=1)
```

✓ 0.3s

```python
cols = ['Name', 'Ticket', 'Cabin']
df_train = df_train.drop(cols, axis=1)
```

✓ 0.4s

## Combined similar features

'SibSp' & 'Parch'

Combined into Companion

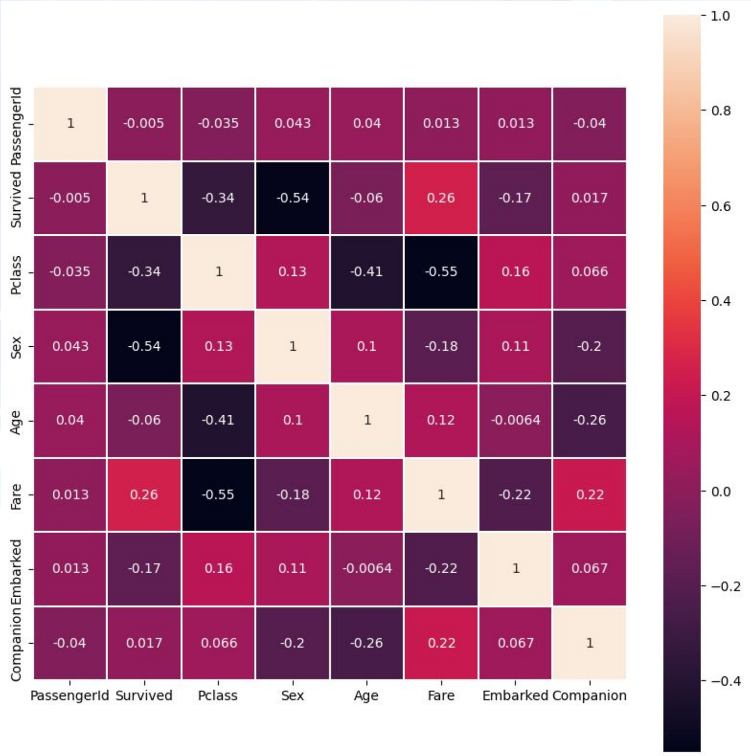
## Dropped unhelpful features

Cabin

Name

Ticket

# *Analysis & Insights*




## Correlationship Heatmap

The darker the color, the worse the relationship

-Age & Pclass,

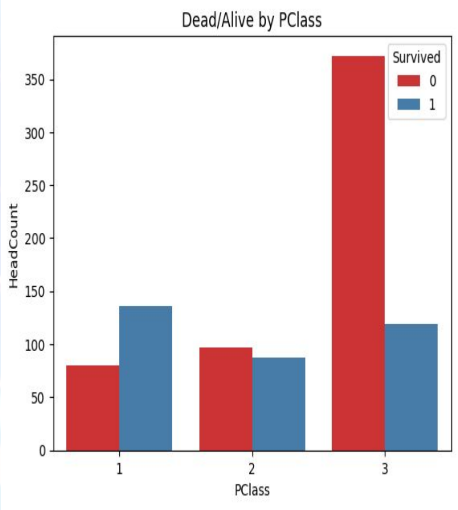
-Fare & PClass,

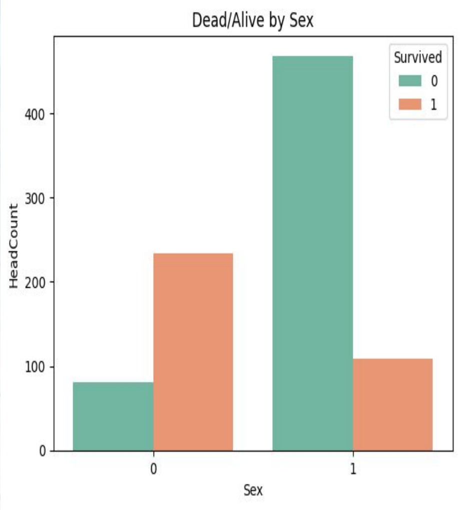
-Pclass & Survived

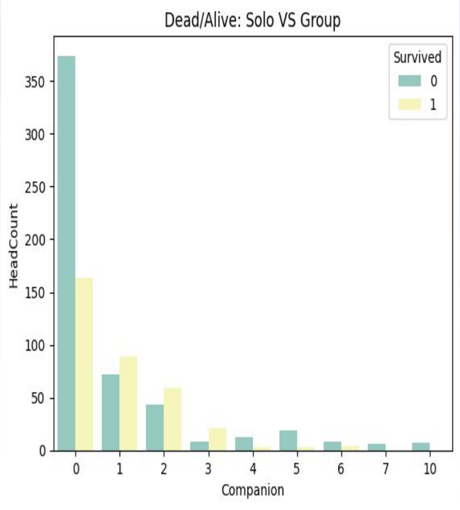
-Sex & Survived,

Have terrible relations.

# *Analysis & Insights*




## Dead/Alive by Pclass

1st class has the highest probability of surviving

## Dead/Alive by Sex

Females are more likely to survive

## Dead/Alive wrt Companion

Solo passengers are much more likely to die

Groups between 1-3 have better chances of surviving

# *Machine Learning Models*

| Score | Model |
|---|---|
| 92.14 | Random Forest |
| 80.25 | Logistic Regression |
| 79.35 | Naive Bayes |
| 78.34 | Support Vector Machines |
| 73.51 | KNN |

# Machine Learning Models (RandomForest)

```
# Random Forest

random_forest = RandomForestClassifier(criterion='gini',
    n_estimators=700,
    min_samples_split=10,
    min_samples_leaf=1,
    max_features='auto',
    oob_score=True,
    random_state=1,
    n_jobs=-1)

random_forest.fit(X_train, Y_train)

Y_pred = random_forest.predict(X_test)

acc_random_forest = round(random_forest.score(X_train, Y_train) * 100, 2)
acc_random_forest
```

✓ 1.7s

```
c:\Users\user\anaconda3\envs\name-py10\lib\site-packages\sklearn\ensemble\_forest.py:425: Fut
be removed in 1.3. To keep the past behaviour, explicitly set `max_features='sqrt'` or remove
RandomForestClassifiers and ExtraTreesClassifiers.
  warn(

92.14
```

```
rf = RandomForestClassifier(n_estimators=700)
scores = cross_val_score(rf, X_train, Y_train, cv=10, scoring = "accuracy")
print("Scores:", scores)
print("Mean:", scores.mean())
print("Standard Deviation:", scores.std())
```

✓ 13.9s

```
Scores: [0.77777778 0.78651685 0.75280899 0.79775281 0.88764045 0.85393258
 0.85393258 0.80898876 0.82022472 0.82022472]
Mean: 0.8159800249687889
Standard Deviation: 0.03840732207728619
```

```
# Initial Run - 92.14
# X-Validation shows that it has an accuracy of 81.5% with deviation of 3.8%
```

Referenced a tuned RandomForest code of a user

Simpler Randomforest code gave a much higher score

Model ran a score of 92.14

Cross-Validation gave accuracy of 81.5%, deviation of 3.8%

# *Machine Learning Models (Log Regression)*

```
# Log Regression          (variable) Y_train: Series
                          (variable) Y_train: Any
logreg = LogisticReg
logreg.fit(X_train, Y_train)

Y_pred = logreg.predict(X_test)

acc_log = round(logreg.score(X_train, Y_train) * 100, 2)
acc_log
```

✓ 0.6s

```
c:\Users\user\anaconda3\envs\name-py10\lib\site-packages\sklearn\linear_model\_logis
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
  n_iter_i = _check_optimize_result(

80.25
```

```
rf = LogisticRegression()
scores = cross_val_score(rf, X_train, Y_train, cv=10, scoring = "accuracy")
print("Scores:", scores)
print("Mean:", scores.mean())
print("Standard Deviation:", scores.std())
```

✓ 0.2s

```
Scores: [0.78888889 0.80898876 0.76404494 0.84269663 0.80898876 0.7752809
 0.80898876 0.79775281 0.83146067 0.79775281]
Mean: 0.8024843945068664
Standard Deviation: 0.02242945427644189
```

Model ran a score of 80.25

Cross-Validation gave accuracy of 80.24%, deviation of 2.24%

However, would require more iterations as per the advice.

Could potentially be a good model to use

# *Machine Learning Models (Naive Bayes)*

```python
# Naive Bayes

gaussian = GaussianNB()
gaussian.fit(X_train, Y_train)

Y_pred = gaussian.predict(X_test)

acc_gaussian = round(gaussian.score(X_train, Y_train) * 100, 2)
acc_gaussian
```

✓ 0.1s

```
79.35
```

```python
rf = GaussianNB()
scores = cross_val_score(rf, X_train, Y_train, cv=10, scoring = "accuracy")
print("Scores:", scores)
print("Mean:", scores.mean())
print("Standard Deviation:", scores.std())
```

✓ 0.1s

```
Scores: [0.71111111 0.76404494 0.7752809  0.79775281 0.79775281 0.7752809
 0.82022472 0.83146067 0.7752809  0.82022472]
Mean: 0.7868414481897628
Standard Deviation: 0.0333370363637702
```

```python
# Initial Run - 79.35
# X-Validation shows that it has an accuracy of 78.7% with deviation of 3.33%
```

Model ran a score of 79.35

Cross-Validation gave accuracy of 78.7%, deviation of 3.33%

Not bad, but not the best

# Machine Learning Models (SVM)

```
# SVM

linear_svc = LinearSVC()
linear_svc.fit(X_train, Y_train)

Y_pred = linear_svc.predict(X_test)

acc_linear_svc = round(linear_svc.score(X_train, Y_train) * 100, 2)
acc_linear_svc
```

✓ 0.5s

```
c:\Users\user\anaconda3\envs\name-py10\lib\site-packages\sklearn\svm\_base.py:1244: ConvergenceWarning: Liblinear failed to converge,
iterations.
  warnings.warn(

78.34
```

```
rf = LinearSVC()
scores = cross_val_score(rf, X_train, Y_train, cv=10, scoring = "accuracy")
print("Scores:", scores)
print("Mean:", scores.mean())
print("Standard Deviation:", scores.std())
```

✓ 0.3s

```
c:\Users\user\anaconda3\envs\name-py10\lib\site-packages\sklearn\svm\_base.py:1244: ConvergenceWarning: Liblinear failed to converge,
iterations.
  warnings.warn(
c:\Users\user\anaconda3\envs\name-py10\lib\site-packages\sklearn\svm\_base.py:1244: ConvergenceWarning: Liblinear failed to converge,
iterations.
  warnings.warn(
c:\Users\user\anaconda3\envs\name-py10\lib\site-packages\sklearn\svm\_base.py:1244: ConvergenceWarning: Liblinear failed to converge,
iterations.
  warnings.warn(
c:\Users\user\anaconda3\envs\name-py10\lib\site-packages\sklearn\svm\_base.py:1244: ConvergenceWarning: Liblinear failed to converge,
iterations.
  warnings.warn(
c:\Users\user\anaconda3\envs\name-py10\lib\site-packages\sklearn\svm\_base.py:1244: ConvergenceWarning: Liblinear failed to converge,
iterations.
  warnings.warn(
c:\Users\user\anaconda3\envs\name-py10\lib\site-packages\sklearn\svm\_base.py:1244: ConvergenceWarning: Liblinear failed to converge,
iterations.
  warnings.warn(
c:\Users\user\anaconda3\envs\name-py10\lib\site-packages\sklearn\svm\_base.py:1244: ConvergenceWarning: Liblinear failed to converge,
iterations.
  warnings.warn(
c:\Users\user\anaconda3\envs\name-py10\lib\site-packages\sklearn\svm\_base.py:1244: ConvergenceWarning: Liblinear failed to converge,
iterations.
  warnings.warn(

Scores: [0.71111111 0.80898876 0.65168539 0.84269663 0.7752809  0.60674157
 0.80898876 0.78651685 0.70786517 0.79775281]
Mean: 0.7497627965043694
```

Model ran a score of 78.34

Cross-Validation gave accuracy of 75%, deviation of 7.3%

Repeated warnings of 'LibLinear failed to converge'

Current parameters are not good enough, hence predictions from this model might be useless

# Machine Learning Models (KNN)

```
# KNN

knn = KNeighborsClass
knn.fit(X_train, Y_tr

Y_pred = knn.predict(X_test)

acc_knn = round(knn.score(X_train, Y_train) * 100, 2)
acc_knn
```

(variable) X_test: DataFrame

(variable) X_test: Any

0.8s

```
73.51
```

```
rf = KNeighborsClassifier(n_neighbors = 6)
scores = cross_val_score(rf, X_train, Y_train, cv=10, scoring = "accuracy")
print("Scores:", scores)
print("Mean:", scores.mean())
print("Standard Deviation:", scores.std())
```

0.1s

```
Scores: [0.48888889 0.62921348 0.49438202 0.46067416 0.4494382  0.52808989
 0.50561798 0.50561798 0.68539326 0.65168539]
Mean: 0.5399001248439451
Standard Deviation: 0.07953945202686367
```

```
# Initial Run - 73.51
#X-Validation shows that it has an accuracy of 54% with deviation of 8%
# Hence this model isnt very good
```

Model ran a score of 73.51

Cross-Validation gave accuracy of 54%, deviation of 8%

Not a good predictive model


https://github.com/Clemykeileo/ML-Project.git

# Thank you!