Titanic Binary Classification

January 17, 2018

1 Importing the libraries

```
In [1]: import numpy as np
        import matplotlib.pyplot as plt
        import pandas as pd
```

2 Importing the dataset

```
In [4]: X_train = pd.read_csv('Data/train_wrangled_X.csv').values
        y_train = pd.read_csv('Data/train_wrangled_y.csv').values
        X_test = pd.read_csv('Data/test_wrangled.csv').values
        y_test = pd.read_csv('Data/y_test_wrangled.csv').values
```

3 Feature Scaling

```
In [5]: from sklearn.preprocessing import StandardScaler
        sc = StandardScaler()
        X_train = sc.fit_transform(X_train)
        X_test = sc.transform(X_test)
```

4 Fitting Random Forest Classification to the Training set

```
In [6]: from sklearn.ensemble import RandomForestClassifier
        classifier = RandomForestClassifier(n_estimators = 10,
                                            criterion = 'entropy', random_state = 0)
        classifier.fit(X_train, y_train)
```

```
C:\Users\Ripti\Anaconda3\lib\site-packages\ipykernel_launcher.py:7: DataConversionWarning: A col
  import sys
```

```
Out[6]: RandomForestClassifier(bootstrap=True, class_weight=None, criterion='entropy',
                    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=1,
                    oob_score=False, random_state=0, verbose=0, warm_start=False)
```

5 Predicting the Test set results

```
In [7]: y_pred = classifier.predict(X_test)
```

6 Making the Confusion Matrix to Determine Accuracy

```
In [21]: from sklearn.metrics import confusion_matrix
         cm = confusion_matrix(y_test, y_pred)
         print(cm)
         print("Accuracy: "+str(round((((
             241+114)/(241+114+38+25))*100),2))+"%")
```

```
[[241  25]
 [ 38 114]]
Accuracy: 84.93%
```