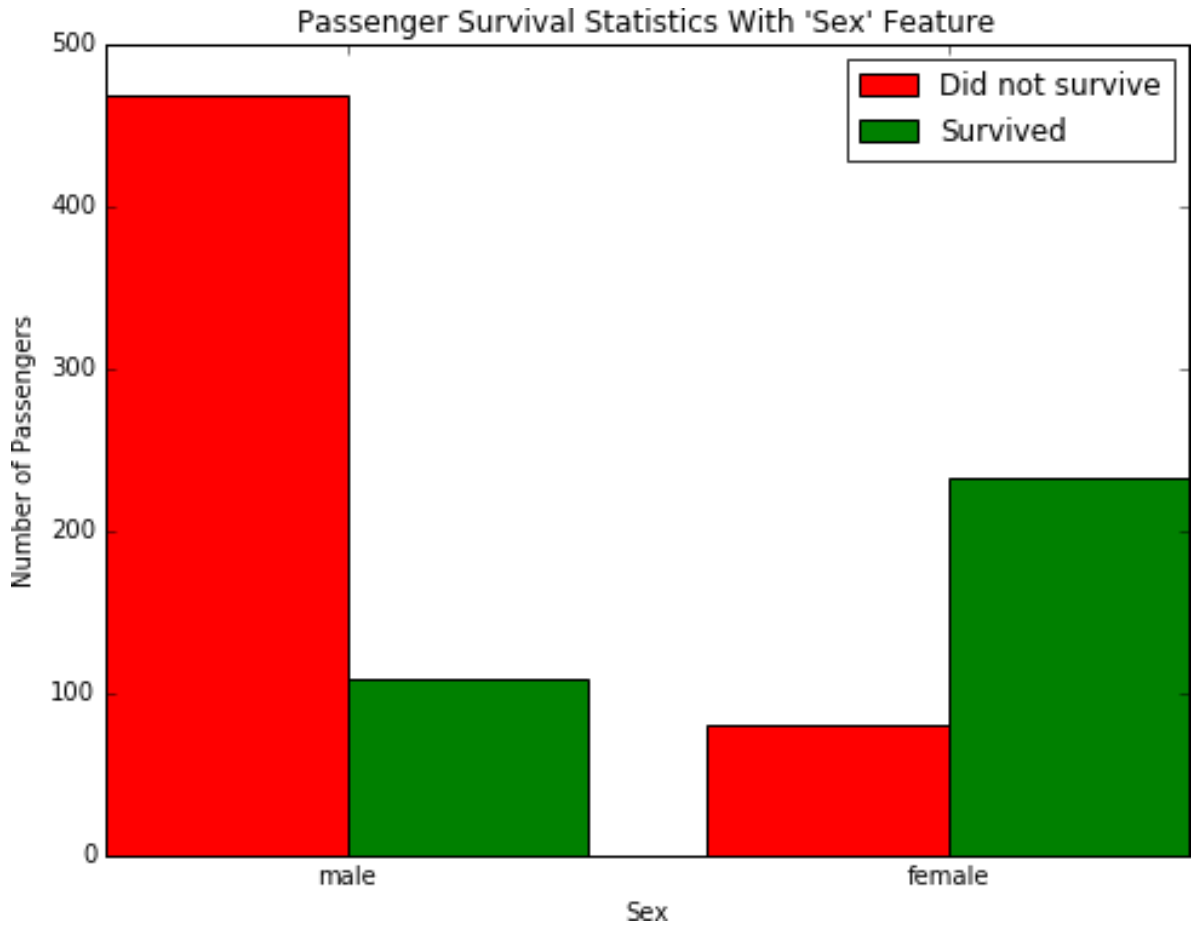
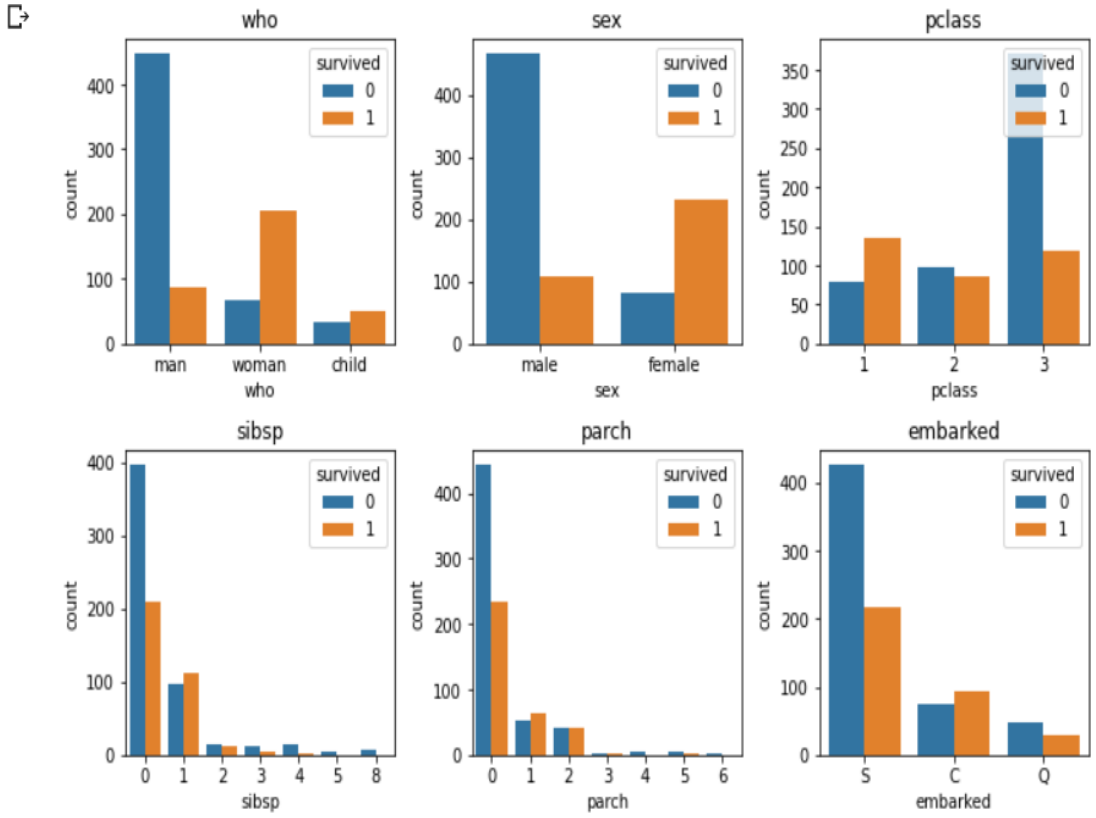
MACHINE LEARNING PROJECT AND DATA VISUALISATION ON ON TITANIC DATASET

by: Varshit Singh Yadav

SURVIVAL OUTCOME OF PASSENGERS BASED ON SEX




```
survival_stats(data, outcomes, 'Sex')
```



Execution and Outcomes of different Machine Learning algorithms.

```
def models(X_train,Y_train):
```

#Using Logistic Regression Algorithm to the

Training Set from sklearn.linear_model import

LogisticRegression log =

LogisticRegression(random_state = 0)

log.fit(X_train, Y_train)

Training Accuracy: 0.7978910369068541

```
[[73  9]
 [18 43]]
Model[0] Testing Accuracy = "0.8111888111888111
```

#Using KNeighborsClassifier Method of neighbors class to use Nearest Neighbor algorithm

```
from sklearn.neighbors import KNeighborsClassifier
knn = KNeighborsClassifier(n_neighbors = 5, metric =
'minkowski', p = 2) knn.fit(X_train, Y_train)
```

Training Accuracy: 0.866423374340949

```
[[71 11]
 [20 41]]
Model[1] Testing Accuracy = "0.7832167832167832
```

#Using DecisionTreeClassifier of tree class to use Decision Tree Algorithm

```
from sklearn.tree import DecisionTreeClassifier
tree = DecisionTreeClassifier(criterion = 'entropy', random_state =
0) tree.fit(X_train, Y_train)
```

Training Accuracy: 0.9929701230228472

```
[[60 22]
 [10 51]]
Model[5] Testing Accuracy = "0.7762237762237763
```

#Using RandomForestClassifier method of ensemble class to use Random Forest Classification algorithm

```
from sklearn.ensemble import RandomForestClassifier
forest = RandomForestClassifier(n_estimators = 10, criterion = 'entropy',
random_state = 0) forest.fit(X_train, Y_train)
```

Training Accuracy: 0.9753954305799648

```
[[67 15]
 [13 48]]
Model[6] Testing Accuracy = "0.8041958041958042
```

#Using GaussianNB method of naïve_bayes class to use Naïve Bayes Algorithm

from sklearn.naive_bayes import

GaussianNB gauss = GaussianNB()

gauss.fit(X_train, Y_train)

Training Accuracy: 0.8031634446397188

```
[[69 13]
 [23 38]]
Model[4] Testing Accuracy = "0.7482517482517482
```

#Using SVC method of svm class to use Support Vector Machine Algorithm

from sklearn.svm import SVC

svc_lin = SVC(kernel = 'linear',

random_state = 0) svc_lin.fit(X_train,

Y_train)

Training Accuracy: 0.7768014059753954

```
[[70 12]
 [18 43]]
Model[2] Testing Accuracy = "0.7902097902097902
```