UML501 Machine Learning Project Report

Prediction Of Survival Of a person in a Shipwreck

Submitted by:

MANINDER SINGH (101703325)

BE Third Year, COE

Submitted to:

DR. Prateek Bhatia

THAPAR INSTITUTE
OF ENGINEERING & TECHNOLOGY
(Deemed to be University)

Computer Science and Engineering Department
TIET, Patiala

November 2019

Problem Statement

In case of any accident, the chance of Survival of a person depends on the location of his Seat, the Age of a Person, Gender of a person. So I decided to build a predictive model that answers the question: “what sorts of people were more likely to survive in case of a shipwreck?” using passenger data (ie name, age, gender, socio-economic class, etc).

.

Datasets

Train.csv: contains data to Train the Model

	A	B	C	D	E	F	G	H	I	J	K	L
1	Passenger	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
2	1	0	3	Braund, M	male	22	1	0	A/5 21171	7.25		S
3	2	1	1	Cumings, M	female	38	1	0	PC 17599	71.2833	C85	C
4	3	1	3	Heikkinen,	female	26	0	0	STON/O2.	7.925		S
5	4	1	1	Futrelle, M	female	35	1	0	113803	53.1	C123	S
6	5	0	3	Allen, Mr. '	male	35	0	0	373450	8.05		S
7	6	0	3	Moran, Mr	male		0	0	330877	8.4583		Q
8	7	0	1	McCarthy,	male	54	0	0	17463	51.8625	E46	S
9	8	0	3	Palsson, M	male	2	3	1	349909	21.075		S
10	9	1	3	Johnson, M	female	27	0	2	347742	11.1333		S
11	10	1	2	Nasser, Mr	female	14	1	0	237736	30.0708		C
12	11	1	3	Sandstrom	female	4	1	1	PP 9549	16.7	G6	S
13	12	1	1	Bonnell, M	female	58	0	0	113783	26.55	C103	S

Test.csv: test data to check the accuracy of the model created

	A	B	C	D	E	F	G	H	I	J	K
1	Passenger	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
2	892	3	Kelly, Mr. J	male	34.5	0	0	330911	7.8292		Q
3	893	3	Wilkes, Mr	female	47	1	0	363272	7		S
4	894	2	Myles, Mr.	male	62	0	0	240276	9.6875		Q
5	895	3	Wirz, Mr. /	male	27	0	0	315154	8.6625		S
6	896	3	Hirvonen,	female	22	1	1	3101298	12.2875		S
7	897	3	Svensson,	male	14	0	0	7538	9.225		S
8	898	3	Connolly, I	female	30	0	0	330972	7.6292		Q
9	899	2	Caldwell, M	male	26	1	1	248738	29		S
10	900	3	Abrahim, M	female	18	0	0	2657	7.2292		C
11	901	3	Davies, Mr	male	21	2	0	A/4 48871	24.15		S
12	902	3	Ilieff, Mr. '	male		0	0	349220	7.8958		S

Data Dictionary

Variable	Definition	Key
survival	Survival	0 = No, 1 = Yes
pclass	Ticket class	1 = 1st, 2 = 2nd, 3 = 3rd
sex	Sex	
Age	Age in years	
sibsp	# of siblings / spouses aboard the Titanic	
parch	# of parents / children aboard the Titanic	
ticket	Ticket number	
fare	Passenger fare	
cabin	Cabin number	
embarked	Port of Embarkation	C = Cherbourg, Q = Queenstown, S = Southampton

pclass: A proxy for socio-economic status (SES)
1st = Upper
2nd = Middle
3rd = Lower

age: Age is fractional if less than 1. If the age is estimated, is it in the form of xx.5

sibsp: The dataset defines family relations in this way...
Sibling = brother, sister, stepbrother, stepsister
Spouse = husband, wife (mistresses and fiancés were ignored)

parch: The dataset defines family relations in this way...
Parent = mother, father
Child = daughter, son, stepdaughter, stepson
Some children travelled only with a nanny, therefore parch=0 for them.

Data Pre-processing:

1. Dealing With Attributes having Null Values:

```
dataset=pd.read_csv("train.csv")
testdata=pd.read_csv("test.csv")



dataset['Age'].fillna(dataset['Age'].median(skipna=True),inplace=True)
dataset['Embarked'].fillna(dataset['Embarked'].value_counts().idxmax(),inplace=True)
dataset.drop(['Cabin','Ticket'],axis=1,inplace=True)

```

Since There are some NULL values in the attribute Age. We are replacing the null value in Age with the median of all the non-Null values in the attribute Age.

Similarly, Embarked attribute also has Null or empty values. To deal with them, we replaced them with mode (The value occurring most) of non-empty values of attribute embarked.

Since Cabin No and Ticket No has no impact on Our prediction , we are dropping these attributes.

2. Dropping Attributes that no Effect on our prediction:

```

X=dataset.iloc[:,[2,4,5,6,7,8,9]]
y=dataset.iloc[:,1]

```

3. Label Encoding and One Hot Encoding ‘Sex’ and ‘Embarked’ Attributes:

```

from sklearn.preprocessing import LabelEncoder, OneHotEncoder
label_encoder=LabelEncoder()
X.iloc[:,1]=label_encoder.fit_transform(X.iloc[:,1])
print(X)


label_encoder_x=LabelEncoder()
X.iloc[:,6]=label_encoder_x.fit_transform(X.iloc[:,6])
onehotencoder=OneHotEncoder(categorical_features=[6])
X=onehotencoder.fit_transform(X).toarray()
#print(X)


X=X[:,1:]

```

Since These Attributes Non-Numeric Labels, we need to Label Encode Them.
Since ‘Age ’ has Binary Values we don’t need to One hot Encode it On the Other hand ‘Embarked’ which categorical variables are converted into a form that could be provided to ML algorithms to do a better job in prediction.

4. Machine Learning Models Applied:

- Logistic Regression

Logistic Regression is a Machine Learning classification algorithm that is used to predict the probability of a categorical dependent variable. In logistic regression, the dependent variable is a binary variable that contains data coded as 1 (yes, success, etc.) or 0 (no, failure, etc.). In other words, the logistic regression model predicts P(Y=1) as a function of X.

```
from sklearn.linear_model import LogisticRegression
logisticreg=LogisticRegression(random_state=0)
logisticreg.fit(X,y)
```

- Decision Tree

```
from sklearn.tree import DecisionTreeClassifier
classifier = DecisionTreeClassifier(criterion='entropy', random_state=0)
classifier.fit(X,y)
y_pred=classifier.predict(X_test)
accu_score=accuracy_score(y_test,y_pred)
print(' DECISION TREE  :  Accuracy:', accu_score)
```

- Naïve Bayes

```
from sklearn.naive_bayes import GaussianNB
classifier = GaussianNB()
classifier.fit(X, y)
y_pred=classifier.predict(X_test)
accu_score=accuracy_score(y_test,y_pred)
print(' NAIVE BAYES  :  Accuracy:', accu_score)
```

- Random Forest Classifier

```
from sklearn.ensemble import RandomForestClassifier
classifier = RandomForestClassifier(n_estimators = 10,criterion = 'entropy', random_state = 0)
classifier.fit(X, y)
Y_pred=classifier.predict(X_test)
accu_score=accuracy_score(y_test,y_pred)
print('RANDOM FOREST   :  Accuracy:', accu_score)
```

- SVM (Support Vector Machine)

```
from sklearn.svm import SVC
classifier = SVC(kernel = 'rbf', random_state = 0)
classifier.fit(X, y)
y_pred=classifier.predict(X_test)
accu_score=accuracy_score(y_test,y_pred)
print(' SVM   :  Accuracy:', accu_score)
```

5. Results

The Accuracies Achieved by using different models is as sown below:

```
 SVM   :  Accuracy: 0.6746411483253588



 LOGISTIC   :  Accuracy: 0.6746411483253588
 DECISION TREE   :  Accuracy: 0.7679425837320574
 NAIVE BAYES   :  Accuracy: 0.9186602870813397
RANDOM FOREST    :  Accuracy: 0.9186602870813397

C:\Users\SANDHU\Chances Of Survival In Accident>
```

Random Forest Provided best Accuracy over all other modeks