Workshop Day 5

Case Study 2

Machine learning project in python to predict titanic data set

Splitting the Data set

As we have seen already, In Machine learning we have two kinds of datasets

- Training dataset - used to train our model
- Testing dataset - used to test if our model is making accurate predictions

Our dataset has 480 records. We are going to use 80% of it for training the model and 20% of the records to evaluate our model. copy paste the below commands to prepare our data sets

Though our dataset has lot of columns, we are only going to use the Income fields, loan amount, loan duration and credit history fields to train our model.

Refer to the file Day 5 Session 1 Survival of Titanic disaster Prediction Machine Learning Model.ipynb (demo given in Day5 session2)

Tabulate the accuracies of the following ML Models for Survival of Titanic disaster Prediction ML algorithm by changing different parameters and percentages of training and testing data sets as shown below and conclude which ML model with percentages of training and testing percentages offers the best accuracy.

Model 1 (refer to datasets of training and testing in the demo pdf)

ML Algorithm	Training dataset – 90% Testing dataset – 10%	Training dataset – 70% Testing dataset – 30%	Training dataset – 60% Testing dataset – 40%
Logistic Regression	0.77777	0.76865	0.78431
Decision tree	0.77777	0.75000	0.75070
Random forest	0.82222	0.76865	0.80112

Model 2 (refer to datasets of training and testing in the demo pdf)

ML Algorithm	Training dataset – 90% Testing dataset – 10%	Training dataset – 70% Testing dataset – 30%	Training dataset – 60% Testing dataset – 40%
Logistic Regression	0.76666	0.74253	0.75070
Decision tree	0.80000	0.79477	0.78711
Random forest	0.84444	0.79850	0.78711

Model 3 (refer to datasets of training and testing in the demo pdf)

ML Algorithm	Training dataset – 90% Testing dataset – 10%	Training dataset – 70% Testing dataset – 30%	Training dataset – 60% Testing dataset – 40%
Logistic Regression	0.72222	0.69029	0.68067
Decision tree	0.76666	0.70149	0.66106
Random forest	0.80000	0.70522	0.66386

Results:

Model 2 is the best model as compared with Model 1 and Model 3. This is because it has more accuracy compared to two other models.

Random forest is the best ML algorithm as compared with Logistic Regression and Decision tree. This is because it has more accuracy compared to two other ML algorithms.

Finally prepare a precision, recall, f1-score, support factors and confusing matrix for all models

Training dataset – 90%, Testing dataset – 10%	
Model 1	precision recall f1-score support 0 0.81 0.86 0.83 58 1 0.71 0.62 0.67 32 accuracy 0.78 90 macro avg 0.76 0.74 0.75 90 weighted avg 0.77 0.78 0.77 90
	array([[50, 8], [12, 20]], dtype=int64)
Model 2	precision recall f1-score support 0 0.79 0.86 0.83 58 1 0.70 0.59 0.64 32 accuracy 0.77 90 macro avg 0.75 0.73 0.74 90 weighted avg 0.76 0.77 0.76 90
	array([[50, 8], [13, 19]], dtype=int64)
Model 3	precision recall f1-score support 0 0.73 0.91 0.81 58 1 0.71 0.38 0.49 32 accuracy 0.72 90 macro avg 0.72 0.64 0.65 90 weighted avg 0.72 0.72 0.70 90
	array([[53, 5], [20, 12]], dtype=int64)

Training dataset – 70%, Testing dataset – 30%

Model 1

	precision	recall	f1-score	support
0	0.77	0.85	0.81	156
1	0.76	0.65	0.70	112
accuracy			0.77	268
macro avg	0.77	0.75	0.76	268
weighted avg	0.77	0.77	0.77	268

```
array([[133,  23],
       [ 39,  73]], dtype=int64)
```

Model 2

	precision	recall	f1-score	support
0	0.75	0.83	0.79	156
1	0.73	0.62	0.67	112
accuracy			0.74	268
macro avg	0.74	0.72	0.73	268
weighted avg	0.74	0.74	0.74	268

```
array([[130,  26],
       [ 43,  69]], dtype=int64)
```

Model 3

	precision	recall	f1-score	support
0	0.68	0.88	0.77	156
1	0.72	0.42	0.53	112
accuracy			0.69	268
macro avg	0.70	0.65	0.65	268
weighted avg	0.70	0.69	0.67	268

```
array([[138,  18],
       [ 65,  47]], dtype=int64)
```

Training dataset – 60%, Testing dataset – 40%

Model 1

	precision	recall	f1-score	support
0	0.79	0.87	0.83	210
1	0.78	0.66	0.72	147
accuracy			0.78	357
macro avg	0.78	0.77	0.77	357
weighted avg	0.78	0.78	0.78	357

```
array([[183,  27],
       [ 50,  97]], dtype=int64)
```

Model 2

	precision	recall	f1-score	support
0	0.76	0.83	0.80	210
1	0.73	0.63	0.68	147
accuracy			0.75	357
macro avg	0.75	0.73	0.74	357
weighted avg	0.75	0.75	0.75	357

```
array([[175,  35],
       [ 54,  93]], dtype=int64))
```

Model 3

	precision	recall	f1-score	support
0	0.68	0.86	0.76	210
1	0.68	0.43	0.53	147
accuracy			0.68	357
macro avg	0.68	0.64	0.64	357
weighted avg	0.68	0.68	0.66	357

```
array([[180,  30],
       [ 84,  63]], dtype=int64)
```