LOVELY PROFESSIONAL UNIVERSITY

PROJECT : PITTSBURGH BRIDGES

SUBMITTED BY:

Name : Upendar Reddy

Reg No : 11805157

Roll No : 14

Section : KM007

SUBMITTED TO:

Dr.Rachna Kohar

Introduction:

Pittsburgh Bridges: The Bridges of Pittsburgh play an important role in the city's transportation system. Without bridges, the Pittsburgh region would be a series of fragmented valleys, hillsides, river plains, and isolated communities. In 2006 they determined the number of bridges in Pittsburgh as 446. Pittsburgh is known as "City Of Bridges".

The data set on Pittsburgh Bridges gives us the location of bridges, number of lanes on each bridge, material they are made with, and the different types of models.

I worked on the data-set, trained it for multiple features and tested it on the target. I trained this using three different algorithms (models) to train and predict the output.

Problem statement: I have been provided with an Excel dataset that has 12 columns and 30 rows. My task is to analyse the dataset and predict the types of bridges based on the length, material they are made up of and other attributes.

Cleaning and feature selection of Data-set:

I checked for null values in the data-set. I dropped the null values from the dataset, I assigned “Nan” to all the special characters like “?” and then later on by counting the values of each column, I came out with the most commonly use values in that column and assigned it in the place of Nan.

	Slno	identif	river	location	erected	length	lanes	clear_g	material	span	rel_l	type
0	NaN	E1	M	3	1818	?	2	N	WOOD	SHORT	S	WOOD
1	0.0	E2	A	25	1819	1037	2	N	WOOD	SHORT	S	WOOD
2	1.0	E3	A	39	1829	?	1	N	WOOD	?	S	WOOD
3	2.0	E5	A	29	1837	1000	2	N	WOOD	SHORT	S	WOOD
4	3.0	E6	M	23	1838	?	2	N	WOOD	?	S	WOOD

There are some columns that are so noisy and cannot be used to train the model, so I deleted them from the dataset and worked on the remaining columns.

Dataset after deleting the unnecessary columns:

	river	location	erected	length	lanes	material	span	rel_l	type
0	M	3	1818	NaN	2	WOOD	SHORT	S	WOOD
1	A	25	1819	1037	2	WOOD	SHORT	S	WOOD
2	A	39	1829	NaN	1	WOOD	NaN	S	WOOD
3	A	29	1837	1000	2	WOOD	SHORT	S	WOOD
4	M	23	1838	NaN	2	WOOD	NaN	S	WOOD

Dataset after assigning values to the NaN:

As my data is categorical, the data is in type object but to train it we need it to be in int. So I assigned numerical values to each categorical value and turned it into type int.

	river	location	erected	length	lanes	material	span	rel_l	type
0	1	3	1818	0	2	1	1	1	1
1	2	25	1819	8	2	1	1	1	1
2	2	39	1829	0	1	1	2	1	1
3	2	29	1837	0	2	1	1	1	1
4	1	23	1838	0	2	1	2	1	1

Splitting Data and Model Selection:

Data Splitting:

For training and testing the model, the data is to be splitted into both training data and testing data. Out of 29 rows i considered 21 rows for training the model and 8 rows for testing the model. It is in the percentage 72%(training data) and 28%(testing data).

Model Selection:

After splitting the data I slected three models to train the data
they are:-

1.Logistic Regression
2.Random Forest Classifier
3.K-Neighbors Classifier

HyperParameter Tunning:

Performed Hyper Parameter Tuning for each of the three models so that I could get best parameter for the model . I used both Random Search and Grid Search technique.

Confusion Matrix:

Accuracy was calculated for all the 3 models using confusion matrix

. 1. Random Forest Classifier:(using random search)

Random forest, like its name implies, consists of a large number of individual decision trees that operate as an esemble. Each individual tree in the random forest spits out a class prediction and the class with the most votes becomes our model's prediction.

In this model I have used parameters [bootstrap, max_depth, max_features, min_samples_leaf, min_samples_split, n_estimators] .
The best score is above 85%.

2. Logistic Regression:(using random search)

Logistic Regression is one of the easiest and most commonly used supervised Machine learning algorithms for categorical classification. The basic fundamental concepts of Logistic Regression are easy to understand and can be used as a baseline algorithm for any binary classification problem. In this model I have used parameters[solver, penalty, C].
The best score is above 71%.

3. KNeighbors Classifier:(using grid search)

The k-nearest neighbors (KNN) algorithm is a simple, easy-to-implement supervised machine learning algorithm that can be used to solve both classification and regression problems. The KNN algorithm assumes that similar things exist in close proximity. In other words, similar things are near to each other. The KNN algorithm hinges on this assumption being true enough for the algorithm to be useful. KNN captures the idea of similarity (sometimes called distance, proximity, or closeness) with some mathematics we might have learned in our childhood— calculating the

distance between points on a graph. Here parameters used are [leaf_size, n_neighbors, weights]. The accuracy is above 62% .

Result:

Here based on the best score we take Random Forest Classifier as the best model for our dataset.

Prediction on Unknown data:

After doing all the things from scratch, I have tested My best model which is Random Forest Classifier on unknown data and it worked flawlessly.
Here i will take data from user to check it, for better GUI i have created tkinter window