DMLL Group 3 Lab Report

Members:-

Gowresh

Yehia

Sheetal

Gunjan

Nasir

Lab 1:-

Date: 22-09-22

Summary of Discussion:-

Forest Fire Dataset

The dataset chosen for our lab work is obtained from satellites Tegra and Aqua using an instrument called MODIS based on hotspots of forest fire in Turkey. There are 36,011 records/instances in the dataset with each row pertaining to the satellite readings within a specific time interval. The columns represent 15 attributes/features. The target feature is the 'confidence' which helps to gauge the quality of the hotspot/fire. The 'confidence ' attribute is logged in terms of percentage, which can be represented as a nominal data with three categories Low, Nominal/Medium and High. This dataset can be used to train a model to predict whether the confidence of the forest fire is Low, Nominal or High.

The following are the each attribute names with their data types and attribute types:

Latitude (meters)

float (ratio)

Longitude (meters)

float(ratio)

brightness temperature (Kelvin)

float(ordinal)

scan (meters)

float(ratio)

track (meters)

float(ratio)

acq_date

string(interval)

acq_time

int(interval)

satellite

String(nominal)

instrument

String(nominal)

confidence (%)

int(can be represented as nominal)

version

float(nominal)

brightness 31 channel temperature (Kelvin)

float(ordinal)

frp (MW)

float(ratio)

type (inferred hotspot type)

int (can be represented as nominal)

day night

string(nominal)

This dataset has a good combination of categorical and ordinal data that will help us to best understand ML Practices. There are no missing values in the dataset which may benefit us with accuracy of prediction. The end outcome we are expecting from using this dataset is to predict the confidence levels of the hotspot given a set of attributes. The target feature can be used to create categorical data that will help us to understand the normalization and binning concepts effectively.

Other Datasets chosen:

Mobile Robot Floor Prediction -

A dataset containing 500,000 records containing quaternion, acceleration, and velocity features that all are ratio attributes and a target feature 'floor type' that is nominal

Hand Gesture Detection -

A dataset containing 778 training images of hand gestures and 6 different classes to predict. Additionally, labels are provided defining the bound boxes of the classes in each image.

Lab 2:

Date: 29-09-22

Summary of Discussion: -

There are totally 15 attributes in the dataset. The latitude and longitude attributes refer to the geographic location of the hotspots detected by the satellites. The brightness attribute is the pixel

measurement of the temperature in Kelvin. Scan and Track attributes refers to the pixel measurement of the scanned area. acq_date refers to the date of acquisition of data while acq_time refers to the time spent during the acquisition of data. Satellite attribute refers to which satellite was used for acquiring the data while instrument refers to the instrument used by the satellites for measurement. Confidence refers to the quality of the hostpot in the specified region of interest. This parameter can be used to predict whether a hotspot is a fired area or not. This will be our target feature (Values greater than 60% or 70% usually mean the hotspot is actually a fire spot). Version attribute refers to the collection of sources. bright_t31 is the pixel measurement of temperature in the 3-1 Channel. frp attribute depicts the pixel-integrated fire radiative power in MW (Mega Watts). Daynight attribute refers to when the instance was obtained, whether day or night. Finally, the type attribute refers to the type of environment of the instance.

There were no missing values in the dataset previously. However, to demonstrate the process of data cleaning, we have manually removed the 5 values pertaining to acq_time attribute.

The value_count() displays the number of values in each category of an attribute across the dataset. Here there are 28,203 values indicating that the respective measurements were taken during daytime while 7,808 values indicating that those measurements were taken during nighttime. The satellite dataset has two categories indicating the model of satellite that was used for measurement. It is observed that the instrument attribute has only one category across the entire dataset, which clearly indicates that the attribute can dropped from the dataset when processing.

The histogram plots help us to summarize between discrete and continuous data that are measured on an interval scale. Through the plot we can identify that frp attribute has an outlier. Similarly, it is also evident that the attributes version and type do not have any effect on the target feature.

While the acq_date is not categorical data, we must consider the activity of changing the data type.

The first plot does not help us in finding any pattern. The second plot is comparatively better in representing the density of areas where the measurements was taken. The third plot is more indicative as we can understand the different confidence levels of hotspot across the areas.

The most correlated attributes with the target attribute confidence are the brightness and the frp such that they have strong positive correlation. According to the definitions of these two attributes, it makes sense that the brightness, which is the temperature of hotspot region, and the frp, which is the radiative power of the hotspot, can affect the confidence of the hotspot more than the other features. The scan, track and acq_time attributes have negative correlation with the confidence attribute. This may be due to the fact that the more time the satellite spends on a particular region of interest, the less likely it can be that that region has low confidence of hotspot.

Due to the upward trend, it can be concluded that the correlation is strong. There are straight horizontal lines at around 100, 90, 85, 75 and even around 0. This needs to be considered to prevent data quirks.

We have represented the categorical values in numbers. Generally, in ML, there may be a possibility that nearby values are more similar than distant values. However as far as this dataset is concerned the categorical values have only binary classification

Since this dataset does not have null values, we are inserting null values to some records to demonstrate the Data Cleaning Process. The three different options of Dealing with Null Data were tried on the dataset: -

(i) Removing the records related to null values

(ii) Removing the entire attribute related to the null values

(iii) Filling the mean values of the attribute to the null values

Of these three options, we have decided to use the third option as it will help us in avoiding the elimination of the records in the dataset.

Normalisation or Standardisation need to be done on the dataset in order to scale the values. For eg, brightness attribute has a min and max values of 300.000000 and 504.4 while frp attribute has a min and max values of 0 and 3679.5. To avoid the numerical values from having such different scales, we perform normalisation or standardisation. To best eliminate outliers in the data set, we are going to use the standardised dataset for further processing.

Lab 3:

Date: 06-10-22

We dropped the attributes containing the same values for all instances in the dataset. The dropped attributes are instrument and version as they contained the same values for all instances.

The two attributes daynight and satellite are categorical in nature. So, we converted these categorical data into numerical data by mapping the classes with 0s and 1s.

We then found out the correlation of the remaining attributes with respect to the class attribute confidence.

Additionally, we tried out the following feature selection algorithms:

1. Univariate Selection
2. Feature Importance
3. Correlation Matrix with Heatmap

Ref: https://towardsdatascience.com/feature-selection-techniques-in-machine-learning-with-python-f24e7da3f36e

We tried to find the best features using Univariate Selection using the reference link: -

Due to using chi square method in Univariate Selection, it does not take negative values (Like Latitude and Longitude). For the purpose of demonstrating the Univariate Selection Method, we are dropping the latitude and longitude attributes. However due to this limitation the results cannot be considered for our dataset

Ref: https://stackoverflow.com/questions/25792012/feature-selection-using-scikit-learn

We found the best features using Feature Importance using the same link. According to the results of this method, the top attributes were brightness, frp, latitude, bright_t31 and longitude.

We finally found out the best features using Correlation Matrix with Heatmap using the same reference link. According to the results of this method, the top attributes were identified as brightness, frp and bright_t31.

Univariate Selection had equivalent results with the correlation matrix but it is not a fit for our dataset because it does not take negative values so attributes with negative values had to be dropped before implementation.

Feature importance also showed comparable results with the correlation matrix algorithm and works well with the dataset provided and makes it easier to see the most correlation features using the histogram.

Correlation matrix with heat map is a visualised form of correlation matrix which also makes it easier to identify the most correlating features.

Consolidating all the results, excluding Univariate Selection for reasons stated above, the top features are brightness, frp and bright_t31.

We performed the binning method to use our dataset as a multi-class problem. Using the binning method, the target feature was classified into 3 classes and the classes stored as numerical data from 0-2

70-100 – High (2)

40-70 – Medium (1)

-1-40 – Low (0)

Note: *We had to use -1 as the start point for Low bin as using 0 caused assignment of NaN values to the instances having 0 confidence.*

As per the lab sheet, to find the top features per each class, we divided the datasets into three as per the following criteria: -

- dataset _1 contains the output labels of class low as 0 while other classes will be denoted as 1
- dataset _2 contains the output label of class medium as 1 while other classes will be denoted as 0
- dataset _3 contains the output label of class high as 2 while other classes will be denoted as 0

On each of the three datasets, we used the heatmap correlation matrix to find out the top correlating features. The observations are as follows:

Note: *We chose top 2, top 5 and top 8 features per each class as our dataset contained only 11 features. Choosing top 10 features yielded in a dataset that was the same as the original dataset.*

Features	Low Class	Medium Class	High Class
Top 2	Brightness, Brightness_t31	Brightness, FRP	Brightness, FRP
Top	Brightness, Brightness_t31, Daynight, FRP, Acq_time	Brightness, FRP, Daynight, Brightness_t31, Track	Brightness, FRP, Brightness_t31, Track, Scan
Top 8	Brightness, Brightness_t31, Daynight, FRP, Acq_time, Scan, Track, Longitude	Brightness, FRP, Daynight, Brightness_t31, Track, Scan, Acq_time, Latitude	Brightness, FRP, Brightness_t31, Track, Scan, Acq_time, Daynight, Latitude

Three datasets were created containing the top 2, top 5 and top 8 correlating features of each of the 3 classes and were named dataset_1, dataset_2 and dataset_3, respectively.

Then using SciKit learn, we implement the PCA (Principal Component Analysis) to reduce the dimensions of our original dataset.

We first tried to reduce the entire dataset into 2 dimensions. We found out that the explained variance of the resulting principal components was less (0.27705587, 0.19335908). To explore further we tried to reduce the dimensions to a value such that the variance is preferred to 95%. The result was that 11 features were reduced to 8 principal components with variances (0.27705587, 0.19335908, 0.14166932, 0.1151124, 0.09112534, 0.08817307, 0.03785807, 0.02530086)

Using a simple classifier (Decision Tree Classifier) we trained and tested the accuracy of our model for our original dataset as well as the three subsets.

The following were the recorded accuracies by running Decision Tree Classifier on the different datasets: -

Dataset Type	**Prediction Accuracy(%)**
Original	67.4
Dataset_1 containing Top 2 features per each class	61
Dataset_2 containing Top 5 features per each class	66.4
Dataset_3 containing Top 8 features per each class	67.2

It is our observation that for our dataset, creating subsets had not increased the accuracy of the classifier model. But it is also clear that the accuracy of the dataset containing top 8 features is the same as the accuracy of the original dataset containing the 11 features. It was also observed that running the PCA on our original dataset reduced the dimensions from 11 to 8.

Lab 4:

Date: 13-10-22

In this lab, we used three different classifiers on our dataset and evaluated each classifier with accuracy, precision, recall, F1 score, confusion matrix and ROC.

To begin with we had to adapt our dataset so that it can be used for binary classification. As ROCs can be computed only for binary classification models, we decided to use binning method to convert our target feature, confidence, into two different classes. As in the previous lab, we used binning method but with two labels. The label 1 classifies the confidence values greater than 70%(High Confidence) while label 0 classifies the confidence values less that 70% (Low Confidence).

For evaluation purposes, we split our dataset into a training set and a test set. We used the sciket-learn's train_test_split function to split the dataset such that 75% of the original dataset was in training set while the remaining 25% was in test set.

As recommended by the Python Tutorial Module 2, we further changed the target features test set and train set into an array of True or False, True referring to High Class of Confidence and False referring to Low class of Confidence.

We fitted three different classifiers on the training dataset. The classifiers are: -

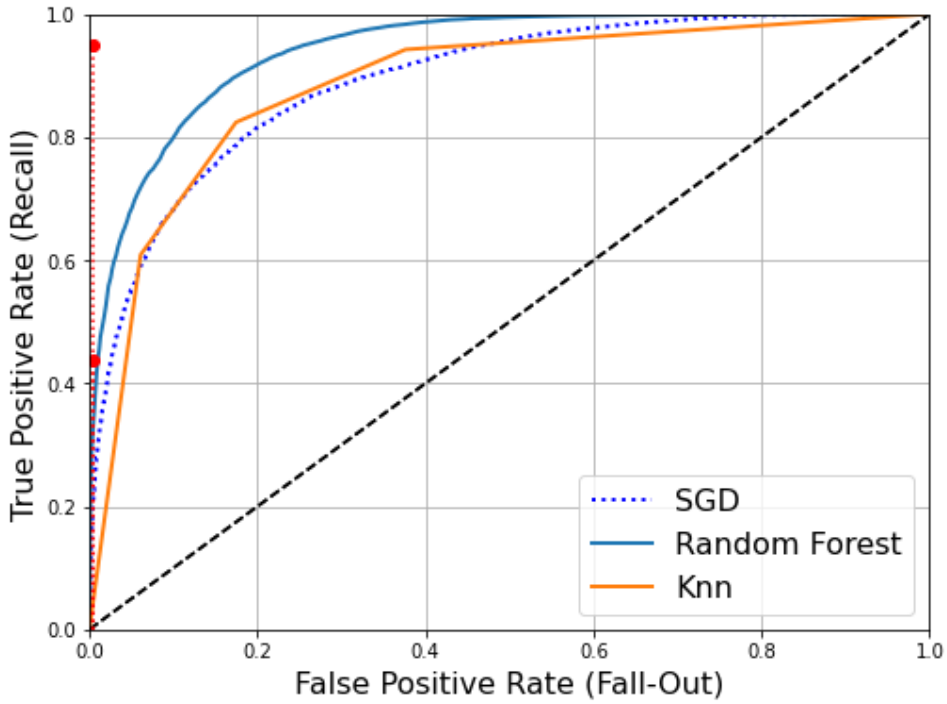
1. Stochastic Gradient Descent Classifier
2. Random Forest Classifier
3. K-Nearest Neighbour Classifier

With the test set, the following are the results obtained using SGD Classification

Metrics	**SGD Classifier**
Accuracy	69.6%
Precision	76.3%
Recall	74.3%
F1 Score	74.08%
Confusion Matrix	([[12025,1713], [5140, 8130]])
ROC Area	89.17%

After the two other classifiers were used, the Receiver Operating Curve (ROC) was plotted and the region under the curve was found out for each of the classifiers. The results are tabulated as follows: -

Metrics	SGD Classifier	RF Classifier	KNN Classifier
Precision	76.3%	85.4%	82.0%
Recall	74.3%	86.3%	82.4%
ROC Area	89.17%	94.18%	88.3%
True Positives	8130	11462	10936
False Negatives	5140	1808	2334
True Negatives	12025	11786	11341
False Positives	1713	1952	2397



From the above plot and the tabulated results, it is evident that the Random Forest Classifier is most suitable for this dataset as its area under the curve is higher than the other classifiers.

The Confusion Matrix compares the decisions made by the model with the actual decisions. This will give an understanding of the level of accuracy of the model in terms of how well the model will perform on new previously unseen data. It is useful to compare the performance as measured using the validation and the testing dataset with the performance as measured using the training dataset. In our case of prediction of the forest fire confidence, the Random Forest Classifier has the highest True Positives, which is essential for the prediction of whether a given hotspot is a forest fire or not.

ROC provides more visual clarity on where to make the cut off when evaluating a continuous measure. It gives an idea about the range of behaviours between the true positives and the false

positives rate. It is plotted with the two metrics against each other, TPR on y axis and FPR on x axis.

Implementation of Pipeline

The pipeline in Python is used to automate the flow of code in modelling for machine learning. The pipeline was implemented to automate the flow of transformation from categorical to numerical data.