DYNAMIC DUO

LCC MACHINE LEARNING WORKSHOP

HACKATHON

Team Member 1 : Aarthi T

Team Member 2 : Akshay Anand

SUPERVISED CLASSIFICATION PROBLEM :

- Given a dataset with 14 columns and 1 target column which is a binary classification , i.e., Class 0 or Class 1.
- The 14 columns are independant features and the target column is the dependant feature. Hence, the given dataset is multivariate.
- Out of the 14 columns, 11 columns have numerical attributes and 3 of them have categorical attributes.
- The number of instances are 32561, which we need to slice in the ratio 90:10 for the training set and test set respectively.

DATA VISUALISATION AND STATISTICS :

- We have used **matplotlib.pyplot** in our project to plot the Cost vs Iterations graph in the later part to find the accuracy of the model.

- From the dataset, by applying statistical formulas to the numerical columns, we observe few important details about the dataset :

  # Columns 6, 7 and 8 have missing values.

  # In column 2, more than 90% of the values are 0.

  # In column 6, more than 80% of the values is 1.

  # In column 7, more than 75% of the values is 3.

  # In column 12, very large values are stored.

DATA PREPROCESSING :

- In column 6, there are 583 NaN values. The mean of the column is 2.4 and the mode is 1. The column contains 29170 values as 1. So we replace the NaN values with the mode of the column, i.e., 1.

- In column 7, there are 1836 NaN values. The mean of the column is 3.15 and the mode is 3. The column contains 22696 values as 3. As the values 3 and 3.15 are close and mean is better option because less than 75% values are equal to the mode 3. Hence, we replace the NaN values with the mean of the column, i.e., 3.15.

- In column 7, there are 1843 NaN values. The mean of the column is 5.29 and the mode is 4. The column contains 4140 values as 4. Mean is better option because less than 13% values are equal to the mode 4. Hence, we replace the NaN values with the mean of the column, i.e., 5.29.

- 3 columns of the dataset have categorical values, where column 1 contains values {a,b,c,d,e,f,g}, column 9 contains {a,b} and column 10 contains {a,b,c,d,e,f}.

- As all the 3 columns have the same categorical values {a,b,…}, we use Categorical Encoding to replace them with numerical values.
- One Hot Encoding (Dummy Encoding) is not not used as there are 7 different categorical values, as a result it will increase the number of columns in the dataset, which is not necessary.

FEATURE SCALING :

- We know that, if one of the features has a broad range of values, the distance will be governed by this particular feature. Therefore, the range of all features should be normalized or standardized, so that each feature contributes approximately proportional to the final distance.
- Hence, columns 3, 4, 5, 12, 13 and 14 have been normalized to scale them down proportionately due to their broad range of values.
- Formula used :

$$X_{norm} = (X - X_{min}) / (X_{max} - X_{min})$$

MODEL SELECTION :

- The different algorithms that aid as tools for solving classification problems are :

- Naïve Bayes classifier
- K Nearest Neighbor classification
- Decision Trees
- Logistic Regression
- Neural Networks

- Decision trees assume that our decision boundaries are parallel to the axes, So decision trees chop up the feature space into rectangles (or in higher dimensions, hyper-rectangles). There can be many partitions made and so decision trees naturally scale up to creating more complex (say, higher VC) functions - which can be a problem with over-fitting.

- Therefore, descision trees are usually used for a multi-class classification. And the given dataset being a binary classification, if descision trees are used it will result in the increase in complexity and is more liable to overfit.

- And we know that, logistic regression assumes that there is one smooth linear decision boundary. It finds that linear decision boundary by making assumptions that the P(Y|X) of some form, like the inverse logit function applied to a weighted sum of our features. Then it finds the weights by a maximum likelihood approach.

- So, logistic regression will work better if there's a single decision boundary, not necessarily parallel to the axis. And logistic regression is intrinsically simple, it has low variance and so is less prone to over-fitting.

- We know that, Naive Bayes (NB) is a very simple algorithm based around conditional probability and counting. Essentially, your model is actually a probability table that gets updated through your training data. To predict a new observation, we'd simply "look up" the class probabilities in your "probability table" based on its feature values.

- Due to their sheer simplicity, NB models are not properly trained and tuned as the other algorithms which are present.

- Also as the number of featuers increase, the complexity also increases. As a result, to obtain a better efficiency and accuracy, avoiding NB is a good idea.

- KNN needs to be carefully tuned, the choice of K and the metric (distance) to be used are critical. For many datapoints KNN has performance problems. If you have a higher number of dimensions then you need an approximation to the nearest neighbors problems and whenever we use an approximation we have to think if KNN with the NN approximation is still better than other algorithms.

- KNN is also very sensitive to bad features (attributes) so feature selection is also important. KNN is also sensitive to outliners. So for the above reasons, KNN is also not used.

- Therefore, for the given dataset as there isn't any need of feature reduction or dimentionality reduction,we use Logistic regression for a better performance.

EVALUATION METRIC :

The observed acuuracies for different values of alpha on the tarining set and the test set are as follows with the number of iterations = 1500 :

ALPHA	TRAINING SET ACCURACY	TEST SET ACCURACY
0.1	78.87	57.51
0.05	77.14	60.92
0.01	76.30	74.60
0.001	76.00	76.04
0.0001	75.99	75.30

❖ We observe that the best case is obtained when alpha is equal to 0.001. The graph of Cost vs No. Of Iterations shows us that the convergence is attained near 100 iterations itself. This means that the initial cost is close to minimum cost, hence it is achieved in less number of iterations with decent accuracy for both training set and test set, i.e., 76% and 76.04% respectively.

- We observe the worst case when alpha = 0.1. The graph does not converge even after 1500 iterations. This happens because the cost value is oscillating near the minimum cost and alpha being large, hence it is taking more iterations to reach the convergence.
- As a result, overfitting takes place in the training set and so the accuracy obtained for the test set is very less, i.e., 57.51%.

THANK YOU