Project_82_phase_03

1. **Base-line model and metrics**

Base-Line Model:

A baseline model is essentially a simple model that acts as a reference in a machine learning project. Its main function is to contextualize the results of trained models. Baseline models usually lack complexity and may have little predictive power.

Base line model chosen for the three class classification is K-Nearest Neighbours. KNN also called K- nearest neighbor is a supervised machine learning algorithm that can be used for classification and regression problems. K nearest neighbor is one of the simplest algorithms to learn. K nearest neighbor is non-parametric. It does not make any assumptions for underlying data assumptions.

Evaluation Metric:

From the dataset given, it is evident that it is an imbalanced dataset. So, micro averaged F1score is chosen as the performance metric. Micro averaging computes a global average F1 score by counting the sums of the True Positives (TP), False Negatives (FN), and False Positives (FP). Sum the respective TP, FP, and FN values across all classes and then put them into the F1 equation to get micro F1 score.

2. Classification Models used for Modeling are:

a. K-Nearest Neighbours

The k-nearest neighbors algorithms a non-parametric, supervised learning classifier, which uses proximity to make classifications or predictions about the grouping of an individual data point.

Advantages:

I. KNN modeling does not include training period as the data itself is a model which will be the reference for future prediction

II. It is very time efficient in term of improvising for a random modeling on the available data.

III. KNN is very easy to implement as the only thing to be calculated is the distance between different points on the basis of data of different features and this distance can easily be calculated using distance formula such as- Euclidian or Manhattan

Disadvantages:

i. Does not work well with large dataset as calculating distances between each data instance would be very costly.

ii. Does not work well with high dimensionality as this will complicate the distance calculating process to calculate distance for each dimension.
iii. Sensitive to noisy and missing data
iv. Data in the entire dimension should be scaled (normalized and standardized) properly.

Applications:

Text mining, Agriculture, Finance, Medical, Facial recognition, Recommendation systems (Amazon, Hulu, Netflix, etc)

b. Naive bayes

Naïve Bayes algorithm is a supervised learning algorithm, which is based on Bayes theorem and used for solving classification problems.

Advantages:

i. It is simple and easy to implement and It doesn't require as much training data
ii. It handles both continuous and discrete data
iii. It is highly scalable with the number of predictors and data points
iv. It is fast and can be used to make real-time predictions

Disadvantages:

I. If test data set has a categorical variable of a category that wasn't present in the training data set, the Naive Bayes model will assign it zero probability.
II. This algorithm is also notorious as a lousy estimator
III. It assumes that all the features are independent. Which is not possible in real life

Applications:

Real time Prediction, Multi class Prediction, Text classification/ Spam Filtering/ Sentiment Analysis, Recommendation System

c. XGboost

XGBoost is a decision-tree-based ensemble Machine Learning algorithm that uses a gradient boosting framework.

Advantages:

I. It is Highly Flexible and uses the power of parallel processing.
II. It is faster than Gradient Boosting and supports regularization.
III. It is designed to handle missing data with its in-build features.
IV. The user can run a cross-validation after each iteration.

Disadvantages:

I. Difficult interpretation , visualization tough

II. Over fitting possible if parameters not tuned properly.
III. Harder to tune as there are too many hyper parameters.

Applications:

Anomaly detection, quality prediction

d. Random Forest:

It builds decision trees on different samples and takes their majority vote for classification

Advantages:

I. It reduces over fitting problem in decision trees and also reduces the variance and therefore improves the accuracy.
II. Random Forest can be used to solve both classifications as well as regression problems.
III. Random Forest can automatically handle missing values and no feature scaling required
IV. Handles non-linear parameters efficiently and automatically handle missing values.
V. Random Forest is usually robust to outliers and can handle them automatically.
VI. Random Forest algorithm is very stable and less impacted by noise.

Disadvantages:

I. Complexity
II. Longer Training period

Applications:

Credit card default, fraud customer/not, easy to identify patient's disease or not, recommendation system for ecommerce sites.

e. Decision Tree

Decision Trees are a non-parametric supervised learning method used for classification

Advantages:

I. Normalization or scaling of data not needed and handling missing values
II. No considerable impact of missing values.
III. Easy visualization Automatic Feature selection
IV. Irrelevant features won't affect decision trees.

Disadvantages:

i. Prone to over fitting.

ii. Sensitive to data. If data changes slightly, the outcomes can change to a very large extent.

iii. Higher time required to train decision trees.

Applications:

Identifying buyers for products, prediction of likelihood of default,

3. Sampling of dataset:

   Dataset is divided randomly into Train data and train data with a ratio of 80:20. After dividing the dataset into two parts. Same can be used for all the classification models. So that, all models will work on the same splitted data and performance of the models can be compared accordingly.

4. K-fold cross validation

   In K-fold cross validation method, total dataset will be used in both training the model and testing the model as well. Based on choosing the K value, data set is divided into k times and each time some portion will be used for testing and remaining will be used for training. Optimum value k=5 is chosen for cross validation

	Model	Train data_F1 score	Test data_F1score	Crossvalid_mean	Crossvalid_std	Grid_bestscore
0	K Nearest Neighbours	0.75	0.68	0.67	0.004	0.67
1	Naive Bayes	0.70	0.70	0.70	0.004	0.70
2	XGboost	0.81	0.77	0.77	0.002	0.78
3	Randomforest	0.87	0.74	0.74	0.005	0.75
4	Decision Tree	0.70	0.70	0.70	0.004	0.74

   From the above table cross validation mean and standard deviation shows that, classification models chosen are performing the same level after changing the train and test data values 5 different combinations also. XGboost is having the highest mean value and less standard deviation. Random forest model has more standard deviation compared to remaining models.

5. Hyper-parameters tuning methods:

   Grid search:

   For modeling on the given dataset for classification, hyper parameters are initialized. But, for getting the optimum hyper parameters for improving the model performance grid search is used.

	Model	Train data_F1 score	Test data_F1score	Crossvalid_mean	Crossvalid_std	Grid_bestscore
0	K Nearest Neighbours	0.75	0.68	0.67	0.004	0.67
1	Naive Bayes	0.70	0.70	0.70	0.004	0.70
2	XGboost	0.81	0.77	0.77	0.002	0.78
3	Randomforest	0.87	0.74	0.74	0.005	0.75
4	Decision Tree	0.70	0.70	0.70	0.004	0.74

After using the grid search for best hyper parameters except for Decision Tree model, remaining models F1_score is same and it shows, best hyper parameters are already initialized in the model. Whereas, after getting the best hyper parameters for Decision Tree and using the same, model F1_score increased from 70 % to 74%.

```
#Finding the best hyperparameters for improving the performance of the model using gridsearch
grid_p = {"criterion":['gini','entropy','log_loss'],"random_state":[50,100,200],
                      "max_depth":range(1,10),"min_samples_leaf":range(1,5)}

grid_search = GridSearchCV(DT, grid_p, n_jobs=-1, cv=5, scoring='f1_micro')
grid_search.fit(X_train, y_train)
best=grid_search.best_score_
grid_search.best_params_
```

```
{'criterion': 'gini',
 'max_depth': 9,
 'min_samples_leaf': 1,
 'random_state': 200}
```

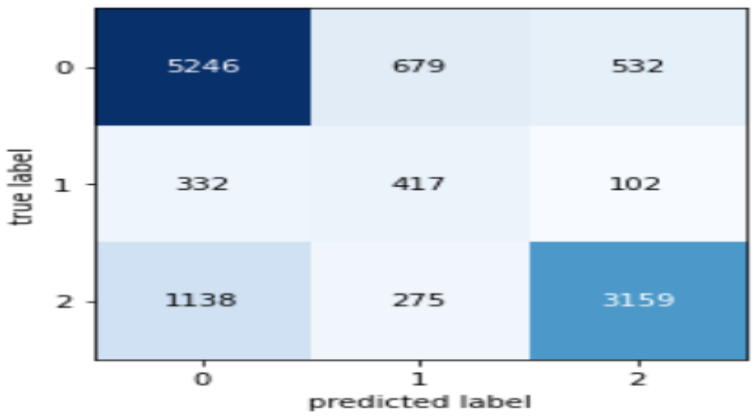
Best hyperparameters for Decision Tree using gridsearch are:{'criterion': 'gini', 'max_depth': 9,'min_samples_leaf': 1,'random_state': 200}

6. Error analysis on models

	Model	Train data_F1 score	Test data_F1score	Crossvalid_mean	Crossvalid_std	Grid_bestscore
0	K Nearest Neighbours	0.75	0.68	0.67	0.004	0.67
1	Naive Bayes	0.70	0.70	0.70	0.004	0.70
2	XGboost	0.81	0.77	0.77	0.002	0.78
3	Randomforest	0.87	0.74	0.74	0.005	0.75
4	Decision Tree	0.70	0.70	0.70	0.004	0.74

Increasing order of classification models based on evaluation metric F1_score Naïve Bayes>Decision Tree>K nearest Neighbours>XGBoost>Random Forest Compared to baseline model K nearest neighbors (F1 score= 0.75) XGBoost and Random forest model performance is more. Random Forest has the highest micro F1 score 0.87 on train data and XGBoost has highest F1_score 0.77 on the test data.

Confusion Matrix of Random Forest



Dataset has a shape of 59400 data points with 40 features. After EDA and preprocessing no.of features are reduced to 90. In sampling the dataset for modeling into train and test data, it is splitted into 80:20 ratio. Now, test data has 11880 points. Adding all the values in confusion matrix will be equivalent to test data size.
Observations from the confusion matrices of all the classification models:

- Models with more diagonal elements sum predicts the better compared to remaining and it classification is better. XGBoost has highest diagonal elements sum
- Decision Tree model unable to classify the test data into class 1 classification and it is biased towards class 0 and class 2
- Random Forest model correctly classified more no.of class 2 data points compared to remaining models.
- Random Forest model correctly classified more no.of class 1 data points compared to remaining models.