Hams Machine learning challenge task report

Abhishek G.L

1 Introduction and Bootstrap ML pipeline

The challenge here is to predict the classes given the data points, As mentioned there are more than two classes hence this is a multi-class classification problem. To start with machine learning process initially we look at the raw data and clean,pre-process it to prepare train and test dataset's and train an initial model with a metric in hand to evaluate the performance of the model.

The initial bootstrap is to organize the given data into test and train sets and perform initial predictions. The following estimators were trained and tested over the entire dataset without involving any modifications or scaling mechanisms, obviously we need to eliminate null values if present which has been done. The following are the initial observations and from fig 5 we can attain the performance of each estimator in the beginning

No of Instances	66137
No of Features	295
No of Classes	5 [A,B,C,D,E]
Missing or null values	0

Table 1: Overview of dataset

From fig 5 we can see there is moderate performance from most of the estimator's except NaiveBayes. Here since we have not yet analyzed the dataset and we also don't know what is the distribution of target classes, Accuracy wont be a efficient metric to evaluate. Lets perform some data analysis and basic feature engineering to adapt the model's better and try to improve their accuracy along with a confusion matrix to better understand the model's performance

2 Data Analysis

2.1 Data description

The first step in order to gain further insights from a given data-set is to analyze its typical properties.

	0	0.1	0.2	20000	0.3	0.4	1	0.5	0.6	0.7	...	0.272	0.273	0.274	0.275	0.276	1.10	0.277	0.278	259.227165	B
0	0	0	0	7059.0	0	0	1	0	0	0	...	0	0	0	0	0	0	1	0	271.983584	E
1	0	0	0	3150.0	0	0	1	0	0	0	...	0	0	0	0	0	1	0	0	235.233437	D
2	0	0	0	24000.0	0	0	1	0	0	0	...	0	0	0	0	0	0	1	0	415.104389	C
3	0	0	0	5600.0	0	0	1	0	0	0	...	0	0	0	0	0	0	0	1	462.230610	D
4	0	0	0	16507.0	1	0	1	0	0	0	...	0	0	0	0	1	0	0	0	824.520326	C

5 rows × 296 columns

Figure 1: Raw data

From the fig 1 we can get a general idea of the dataset, first point being the column names are not assigned therefore we can say its an anonymized dataset, from this point we can also make a assumption that the data is synthetic, but before stating that it is a synthetic dataset we need to further examine the data-points and certain relations or dependencies between the features and the target variables.

2.1.1 Lets analyze the dataset step by step

Organizing the dataset with column names and also we can see the properties with pandas describe method. Some observations here would be col3 with some other cols(not visible in the fig 2) having higher values from mean and std, looks like a potential outlier since majority of the columns contain 0's and 1's. If the values 0's and 1's are encoded previously from a different range of values this could be a potential influencer towards the target.

	col1	col2	col3	col4	col5	col6	col7	col8	col9	col10	...
count	66137.000000	66137.000000	66137.000000	66137.000000	66137.000000	66137.000000	66137.000000	66137.000000	66137.000000	66137.000000	...
mean	0.014757	0.003765	11226.207294	0.213511	0.049261	0.869498	0.030588	0.005867	0.004249	0.003735	...
std	0.120581	0.061244	8153.148240	0.609365	0.216415	0.336857	0.172200	0.076369	0.065044	0.060998	...
min	0.000000	0.000000	13.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	...
25%	0.000000	0.000000	4000.000000	0.000000	0.000000	1.000000	0.000000	0.000000	0.000000	0.000000	...
50%	0.000000	0.000000	9614.800000	0.000000	0.000000	1.000000	0.000000	0.000000	0.000000	0.000000	...
75%	0.000000	0.000000	18984.000000	0.000000	0.000000	1.000000	0.000000	0.000000	0.000000	0.000000	...
max	1.000000	1.000000	72360.000000	10.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	...

8 rows × 294 columns

Figure 2: Dataset properties

- Class Distribution We now know that there are 5 classes, hence this problem is a multi class classification problem. Lets look at the class distribution for the entire dataset. From the fig 3 we can observe that class "C" has the majority compared to other classes.

  This indicates that the dataset is imbalanced. We can employ Over or under sampling techniques to over come imbalance

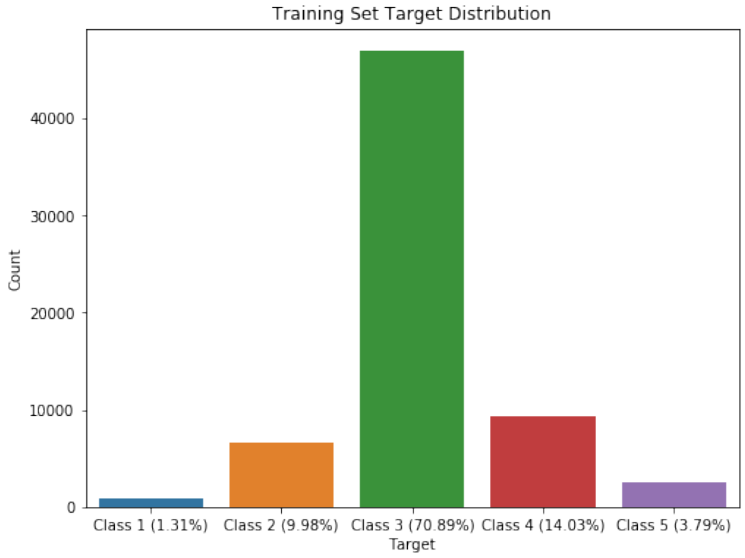


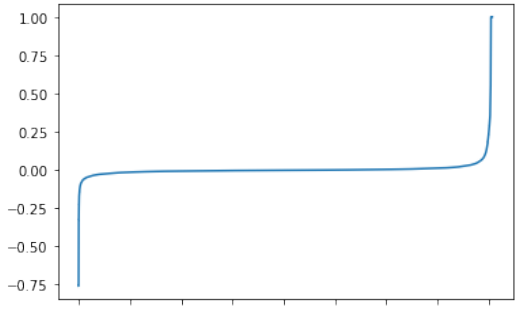
Figure 3: Class distribution

- Correlation

  When dealing with large datasets we need to utilize important features and eliminate those features which have no importance/relevance towards the target classes. Lets look at some important correlations among the features. From the fig **??** below we can observe there is lowest correlations among the listed features, and from the graph majority of the features have highest correlations among them.

col144	col185	1.453386e-08
col166	col249	5.926615e-08
col117	col244	1.003557e-06
col292	col175	1.635319e-06
col35	col133	1.743414e-06
col36	col153	2.465893e-06
col247	col162	2.832268e-06
col194	col96	4.113680e-06

(a) label 1



(b) label 2

Figure 4: Correlations

3 Normalizing, Feature Importance and Cross Validation

Analyzing the features we can say the dataset is sparse, by applying basic feature selection techniques we can gain much more insights and variation between the estimator performances. From the fig 5 we can see from the average of 3 techniques very small variation in performance is observed.With feature selection, SelectKbest(20) most important features were selected and

trained again the impact on the estimators is relatively similar in both the case.

In sampling of classes, Since the classes were imbalanced, Over sampling and under sampling proved to be having a huge impact on the performance since Logistic Regression which was consistent among other techniques proved to be minimal and due to balanced classes, class wise predictions took hold.

With the help Confusion matrix plots we can observe which classes are being predicted correctly from each estimator, this will give us a deeper insight into feature importance and class distribution. Following are some of the figures 6 describing the normalized matrix with percentage of accuracy in predictions across classes.

Finally with Cross validation we can conclude that Logistic Regression and Tree classifiers could be the potential candidates for further tuning and optimization.

4 Conclusion

The process and methods implemented in this report describe only the basic and fundamental data analysis which should be carried out in the beginning of the Machine learning workflow, with narrowing down the potential estimators one can focus more on its hyperparameters and other optimization techniques. For a baseline to start on with data analysis i would follow the mentioned steps as a first step.

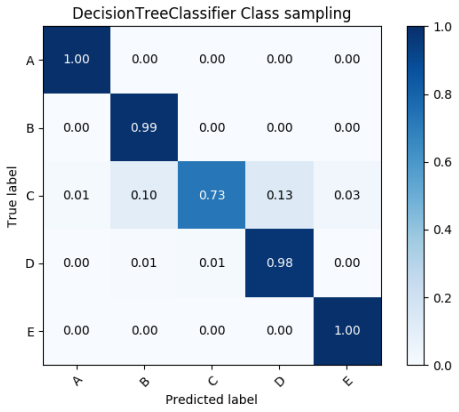
Further notes : The source code is organized as follows

Each file is named after its technique which is implemented inside the file, file 'Compute.py' is the starting point which executes all the methodologies in a sequential manner and with blocked Graphs and print statements describing the process along the way popping up in between.

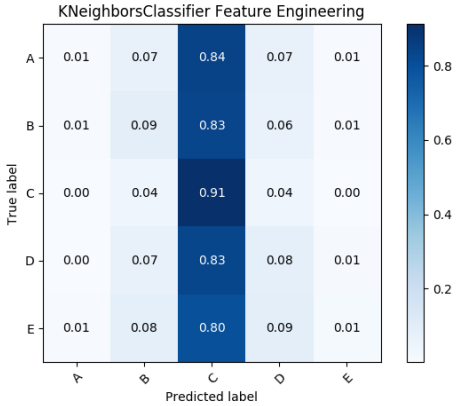
	Bootstrap	Normalization (Avg scores from simple min_max, z_score scaling)	FeatureEngg SelectKbest:20, VarianceThreshold(0.8)	Sampling		Cross-Validation 10 fold CV Avg of 10 iterations
				OverSampling SMOTE,RANDOM	UnderSampling ClusterCentroids,RANDOM	
KNeighborsClassifier	66.89%	68.2%	66.5%, 66.8%	73.1%, 85.3%	15.5%, 25.2%	67%
Naive_Bayes	38.32%	33.2%	64.7%, 15.5%	30.3%, 30.5%	28.4%, 27.2%	41%
LogisticRegression	70.67%	70.8%	70.68%, 70.69%	36.7%, 35.6%	38.2%, 30.7%	70%
DecisionTreeClassifier	62.89%	62.8%	60%, 60.1%	78.5%, 93.8%	53.4%, 32.4%	62%
RandomForestClassifier	70.35%	70.6%	67.8%, 68.56%	89.7%, 80%	64.5%, 34.5%	70%
MLP	67.93%	70.1%	18.65%, 60.68%	50.1%, 22%	25.6%, 22.6%	57%

Figure 5: Overall Performance

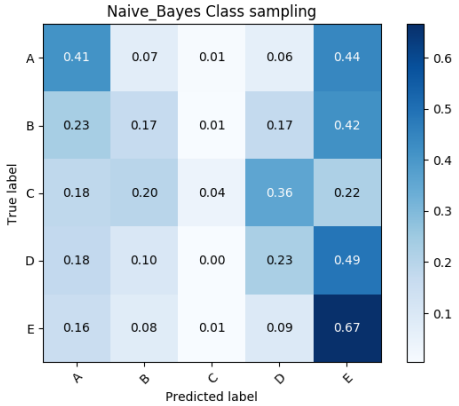
(a) LogisticRegression_OverSampling 1



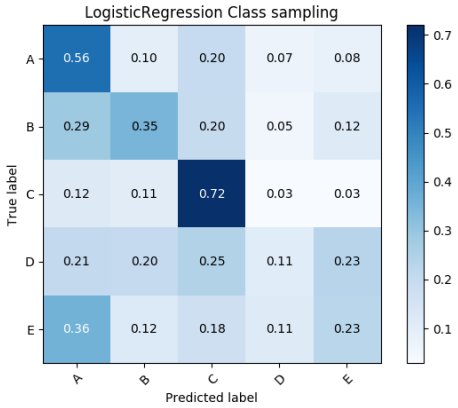
(b) DecisionTreeClassifier_OverRandomSampling 2



(c) KNeighborsClassifier_20features 2



(d) Naive_Bayes_OverSampling 2



(e) LogisticRegression_UnderSampling 2

Figure 6: Confusion Matrix