3 Methodology

We have used the following in our analysis- Data selection, Data-set preprocessing, Algorithm selection, Training and testing of algorithms and noting our observation of cost, accuracy, Fbeta, R2 and time taken to train the ML model.

3.1 Data Set

3.1.1 Regression

For regression algorithm’s analysis we used two data set. One simple data set with 9 features and one complex data set with 384 features.

1. Dataset 1: Physicochemical Properties of Protein Tertiary Structure Data Set with 9 features and 45,730 data points. Format =\*.csv.
2. Dataset 2: Consisted of 384 features extracted from CT images. The class variable is numeric and denotes the relative location of the CT slice on the axial axis of the human body. The data was retrieved from a set of 53500 CT images from 74 different patients (43 male, 31 female). Format =\*.csv.

Figure 1 shows the frequency distribution of value of size of residue in the given collection Dataset 1 indicates maximum frequency between 2-4. Figure 2 shows the frequency distribution of relative location of the image on the axial axis in the given collection dataset 2 indicates maximum frequency which is between 30-40.

3.1.2 Classification

For classification problems we used 2 datasets, one for binary classification and another for multiclass classification

a) Dataset 3: Letter Recognition with 16 features and 20000 data-points having 26 classes each representing one alphabet. Format =\*.csv.

b) Dataset4: Defaulters of credit card clients with 23 features and 30000 data-points, having 2 classes. Class 0 representing non-defaulter and class 1 representing defaulters. Format =\*.csv.

The histogram Figure3 shows the frequency distribution of Customer Segment in the given collection Dataset3 indicating dataset is highly biased towards non-defaulters. The histogram Figure 4 shows the frequency distribution of different classes of Alphabets for the given collection Dataset4 indicating approximately equal distribution among all the classes.

3.2 Pre-processing

3.2.1 Regression

For dataset1 the value of size of residue are continuous and distributed between 0 and 22 similarly for dataset2 the value of relative location of the image on the axial axis is continuously distributed between 0 and 100. We have removed features by analyzing result of on univariant selection technique (SelectKBest class using score function as “mututal\_info\_regression”) and for the missing values we have taken the mean of available values. For database1 we reduced the features from 9 to 5, for database2 feature selection was highly computationally expensive. Both these datasets don’t have any categorical values as all the values are numeric. We also did normalization as well as standardization and compared the results on huge amount of data.

3.2.1 Classification

For dataset 3 the value of output varies between 26 different classes(A-Z) whereas for dataset 4 the value of output varies between two classes (0,1). We have removed features which were highly corelated with each other based on Correlation Matrix. We have only included the features which are having strongest relationship with the output based on univariant selection technique (SelectKBest class using score function as chi2) and feature importance technique (Tree Based Classifiers) Figure 5,6. We reduced the features from 16 to 10 for dataset3 and from 23 to 10 for dataset4. Figure7,8 shows correlation matrix for selected features for both data set. For the missing values mean of available values were taken. There are no categorical values as all the values are numeric. We also did normalization as well as standardization and compared the results on huge amount of data.

3.3 Algorithm

3.3.1 Regression

We have used linear regression for Regression dataset.

3.3.2 Classification

We have used Logistic Regression, Support Vector Classifier (SVC), Random Forest Classifier, K-NN classifier , Decision trees on Classification dataset. Hyperparameter tuning has been done while training the preprocessed data with different values using GridSearchCV. Based on our performance metric we have selected hyperparameter for each of the algorithm listed below.

3.3.2.1 Logistic Regression

We used different values of C for Logistic regression in a range between 0.0001 to 1000 in an increment of 10X and the best accuracy was achieved at C=10, we also tried between l1 and l2 penalty and discovered that penalty l1 is performing better than l2.

3.3.2.2 Support Vector Classifier

We used different values of C for Support Vector Classifier in a range between 0.0001 to 1000 with an increment of 10X and the best accuracy was achieved at C=1.

3.3.2.4 Random Forest Classifier

We used different values of n\_estimators and max\_depth for Random Forest Classifier in a range between 1 to 100 with an increment of +1 individually and the best accuracy was achieved at n\_estimators=10 and max\_depth=50.

3.3.2.4 K-NN Classifier

We used different values of K for K-NN Classifier in a range between 1 to 100 with an increment of +1 and the best accuracy was achieved at K=10. After that there was minimal increase in accuracy and the processing time increased significantly.

3.3.2.4 Decision Tree Classifier

We used different values of max\_depth for Decision Tree Classifier in a range between 1 to 100 with an increment of +1 and the best accuracy was achieved at max\_depth=50.

3.4 Evaluation

We have divided the given data into two different sets training data and testing data. We are training our model using training data, testing it using testing data. From each dataset we took different ratio of testing and train data ranging from 10:90 to 90:10. We observed that when the ratio of training data was 80%, accuracy was almost stable Figure<>. Hence, we have split the data in the ratio of 80:20 for training and validating respectively. For each of the algorithm we picked data points randomly from 500 to total available data with an increment of 500 data points which we were dividing into testing and training and calculating our performance metrics based on it. For regression we have taken mean of 50 iterations as the results varies slightly in each iteration.

4 Results and Discussion

For both classification and regression, we are using time for training and accuracy of the model as our common performance metric.

4.1 Regression

RMSE

R2

4.2 Classification

ACCURACY

FB

4.3 Discussion

4.3.2 Classification

We started with training of raw data and observed that time taken to train the data for SVM was significantly higher in comparison to other algorithm also it increases exponentially with the increase in data. When we did feature selection the training time reduces for all the algorithms. Considering training time as our performance matric we can observe that normalization worked better for SVM, Decision Tree, and Random Forest and standardization worked better for Logistic Regression and K-NN Classifier. On normalizing and standardizing the data the training time of SVM significantly reduces from 128.02 seconds to 10.618 and 7.85985 sec respectively for dataset4. Were as for dataset3 normalization worked better for KNN where as standardization worked better for SVM, logistic regression and both performed nearly equal for Decision tree and random forest.

When the training data is less than 5k we can see too much variance in accuracy in all the scenarios (raw data, after feature selection, normalized, standardized) after that variance reduces and accuracy becomes stagnant between the two thresholds with very less variance (+1/-1) with the increase in data. We also observed that for dataset4 standardization increases the accuracy of SVM from 78.333 to 81.617 and for KNN from 78.083 to 80.95 and normalization increases the accuracy of KNN from 78.083 to 81.050. For Dataset3 SVM performed best with large amount of data were as Logistic regression performed not so we observed that after normalization the accuracy decrease significantly from 97.17 to 71.50 where as for other algorithms both normalization and standardization reduce the accuracy by few percent but at the same time decreases the training time significantly.

Although accuracy is not a good performance metric for dataset4 as it is highly biased. We used F-beta measure also and observed that for standardized data decision tree performed best with about nearly .4 and logistic regression performed worst with .3 The variance in F-beta is high when training data is less than 5k which reduces as the amount of data increases. For normalized data decision tree performed best with f-beta 0.4 whereas SVM F-beta scores nearly remains 0 which shows it’s not a good algorithm for biased data. For dataset3 f-Beta score was achieved by SVM, Random Forest and KNN in standardized data which is >.9 whereas normalization reduces the F-beta for SVM to .65 and increases the F-beta to .94 for KNN and Random Forest.