**Exploratory Data Analysis:**

1. **Describe the dataset:**

The quartiles, mean, min, max, standard deviation are all described to compare mean and 50th quartile. Here, Insulin has some outliers and to suppress we use normalization.

Generally, normalization is used to supress outliers and when we there is no normal Gaussian distribution (Bell -shaped Curve). Next is describing the datatypes of datasets.

1. **Check for NA values:**

No ‘NA’ values in the dataset. If there were ‘NA’ values,

In continuous inputs, then replace with mean.

In case of categorical variables, replace with mode.

1. **Pair-Plot:**

Defines the relationship between variables and outcomes. From this, it is clear that it is classification problem as outcome is clearly differentiable. It is not a Gaussian Distribution.

1. **Correlation:**

This is used to check if there is correlation among the input variables. Generally, more than 0.7 and less than -0.7 is said to be highly correlated.

1. **Skewness and Kurtosis:**

This is used to find distribution of variables and find outliers.

Normal distribution – Skewness is 0.

**Pre-Processing:**

The train test split is 20% test size and 80% train size.

**Feature Scaling:**

**Normalization:**

This is used to supress outliers. Standardization is used only when there is normal distribution and is sensitive to outliers. So, its better to normalize values here. Min Max scaler is used for normalization.

**Model Building:**

**Logistic Regression:**

This method uses the linear features, applies sigmoid non-linear function. The normalized train data is fit to the logistic regression model. The probabilities are predicted and auc- roc curve is drawn. The value that covers maximum area under the curve is taken to calculate the roc-auc score.

**Random Forest:**

This is ensemble model that uses out of bag score for prediction. Feature importance is used to find important features and those that are less than 2% in feature importance can be removed. Random forest generally does not require sampling as random samples are taken. This method is a collection of decision tree classifiers.

Hyper parameters here are

1. n\_estimators - No. of decision trees
2. min\_samples\_leaf - No. of leaf nodes accepted
3. max\_features - Maximum features to be considered for branching

**SVM:**

The kernel function is used here for classification. Here rbf function separates classes radially.

Hyper-parameters here are

C - Constant

Gamma - gamma function

**Naïve\_Bayes:**

This algorithm uses Naïve Bayes theorem (conditional probability). Basic assumption is that it must be a Gaussian Distribution.

**K – Nearest Neighbours:**

This is a lazy algorithm and has slow learning rate. It maps the value based on number of neighbours.

**Decision Tree:**

Each decision tree is branched based on entropy or Gini index value.

Hyper-parameters are,

1. max\_depth - maximum depth of tree
2. min\_samples\_leaf - No. of leaf nodes accepted
3. max\_features - Maximum features to be considered for branching
4. criterion – either entropy or gini

**Gradient Boosting:**

This is similar to gradient descent method where the value of weights is slowly increased to improve the slow learners.

Hyper-parameters are,

1. max\_depth - maximum depth of tree
2. min\_samples\_leaf - No. of leaf nodes accepted
3. max\_features - Maximum features to be considered for branching
4. n\_estimators - No. of decision trees

**XG-Boost:**

This is most efficient boosting algorithm and faster when compared to other boosting algorithms.

Hyper-parameters are,

1. max\_depth - maximum depth of tree
2. min\_samples\_leaf - No. of leaf nodes accepted
3. max\_features - Maximum features to be considered for branching
4. gamma – constant
5. n\_estimators - No. of decision trees

**Ada -Boost:**

This uses decision-tree classifier for better efficiency and give improved results.

Hyper-parameters are,

1. max\_depth - maximum depth of tree
2. min\_samples\_leaf - No. of leaf nodes accepted
3. max\_features - Maximum features to be considered for branching
4. n\_estimators - No. of decision trees

**Sampling:**

Under-sampling is better than over-sampling as it may increase the number of outliers. Nearmiss algorithm is used for under-sampling. Sample the trained dataset.It can be followed by SVM or logistic regression.

**Deep Learning :**

Input dimensions is given in first layer, Activation function used commonly is Relu and for classification problem the last layer has sigmoid as activation function. Common metrics used is binary cross entropy and adam is the optimizer used. Maximum of 100 epochs can be run and optimum batch size is 32.

**Metrics for ML Algorithms:**

**Commonly used metrics are recall, precision, f1-score , accuracy and log-loss.**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Algorithm** | **Recall** | **Precision** | **F1 score** | **Accuracy** | **Log-loss** |
| **Logistic Regression** | 0.789 | 0.714 | 0.750 | 0.805 | 6.728 |
| **Random Forest** | 0.543 | 0.775 | 0.639 | 0.772 | 7.849 |
| **SVM** | 0.473 | 0.818 | 0.6 | 0.766 | 8.074 |
| **Naïve Bayes** | 0.526 | 0.75 | 0.618 | 0.759 | 8.298 |
| **KNN** | 0.333 | 0.826 | 0.475 | 0.727 | 9.419 |
| **Decision Tree** | 0.491 | 0.7 | 0.577 | 0.733 | 9.195 |
| **Gradient Boosting** | 0.456 | 0.787 | 0.577 | 0.753 | 8.522 |
| **XG Boost** | 0.438 | 0.781 | 0.561 | 0.746 | 8.746 |
| **AdaBoost** | 0.438 | 0.781 | 0.561 | 0.746 | 8.746 |
| **Sampling and SVM** | 0.754 | 0.623 | 0.682 | 0.740 | 8.971 |
| **Neural Networks** | **-** | **-** | **-** | 0.772 | **-** |