Supervised Learning

Imbalanced Data

Suppose we have two classes in target column (considering Binary Classification) 0 & 1 with 1000 records. Out of one 1000, 200 are with class 0 and 800 with class 1 then the dataset is called as imbalanced dataset.

Techniques to handle Imbalanced Dataset :

1. Choose proper evaluation metrics

2. Resampling

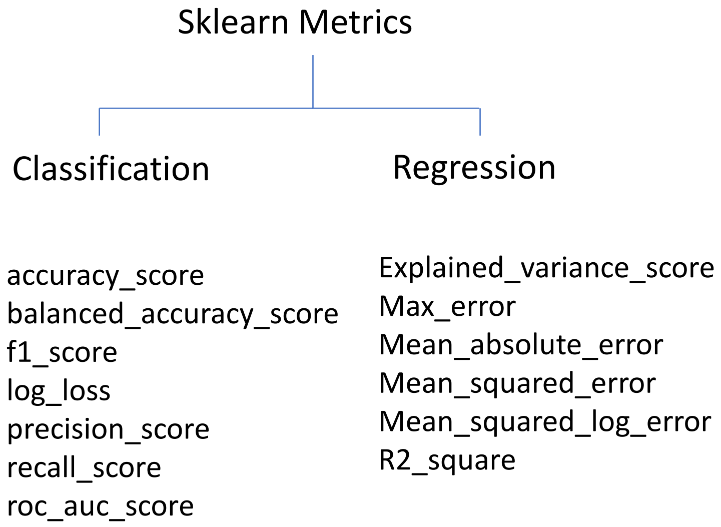
3. Smote

Classification Matrix

Confusion Matrix : The confusion matrix is used to have a more complete picture when assessing the performance of a model.

Metric	Formula	Interpretation
Accuracy	TP+TN / TP + TN + FP + FN	Overall Performance of model
Precision	TP / TP + FP	How accurate the positive predictions are
Recall / Sensitivity	TP / TP + FN	Coverage of actual positive sample
Specificity	TN / TN + FP	Coverage of actual negative sample
F1 Score	2TP / 2TP + FP + FN	Hybrid metric for unbalanced data

		Predicted	
		+	−
Actual	+	TP True Positive	FN False Negative Type II
	−	FP False Positive Type I	TN True Negative



Sklearn Metrics

Classification

accuracy_score
balanced_accuracy_score
f1_score
log_loss
precision_score
recall_score
roc_auc_score

Regression

Explained_variance_score
Max_error
Mean_absolute_error
Mean_squared_error
Mean_squared_log_error
R2_square

ROC : The receiver operating curve is the plot of TPR vs FPR by varying the threshold.

TPR = TP / TP + FN → Recall, Sensitivity

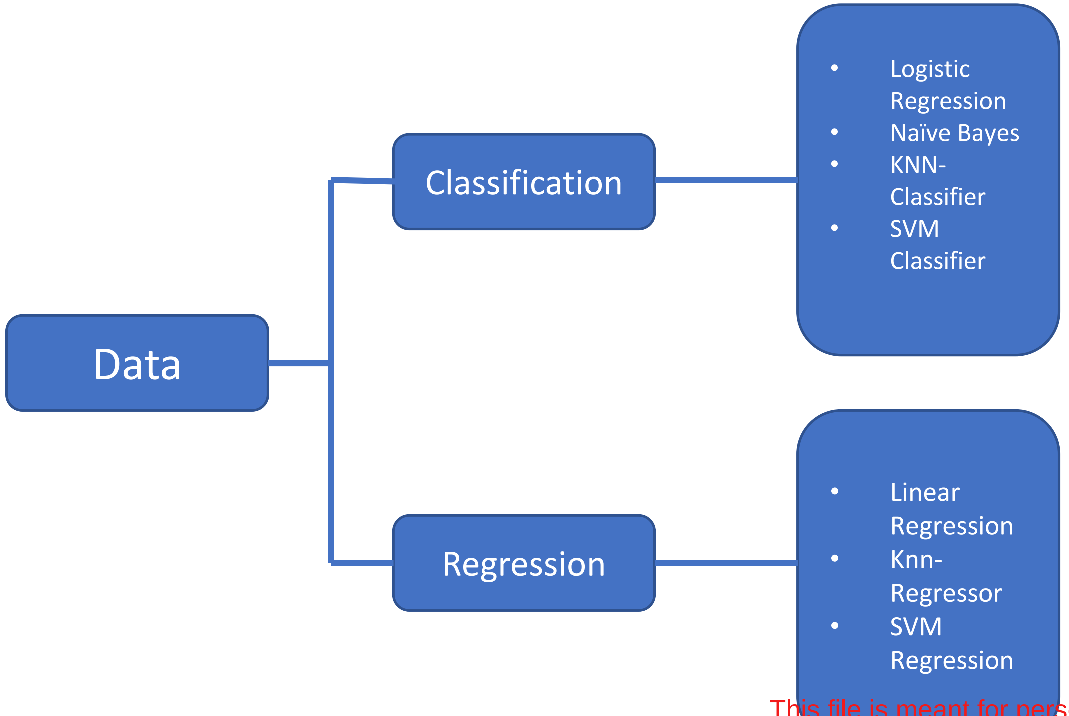
FPR = FP / TN + FP → 1- Specificity

Supervised Learning

Bias and Variance

- Bias - Error in training data
- Variance – Error in testing data
- Generalized model : Low Bias and Low Variance
- Low Bias or / and High Variance - Overfit model
- High Bias or / and low variance - Underfit Model

Machine Learning Algorithms



Data

Classification

- Logistic Regression
- Naïve Bayes
- KNN-Classifier
- SVM Classifier

Regression

- Linear Regression
- Knn-Regressor
- SVM Regression

Data Preprocessing

1. Normalization : Transforms the data in range [0,1]

From sklearn.preprocessing import Normalizer

Norm = normalizer().fit_transform(X_train)

norm_x_test = Norm.transform(x_test)

2. Standardization : Transforms the data with mean = 0 and STD = 1

From sklearn.preprocessing import StandardScaler

SC = StandardScaler().fit_transform(X_train)

scaled_x_test = SC.transform(X_test)

3. Label Encoding :

Datasets that contain multiple labels in one or more than one columns, we need to convert these words to number using label encoding.

From sklearn.preprocessing import LabelEncoder

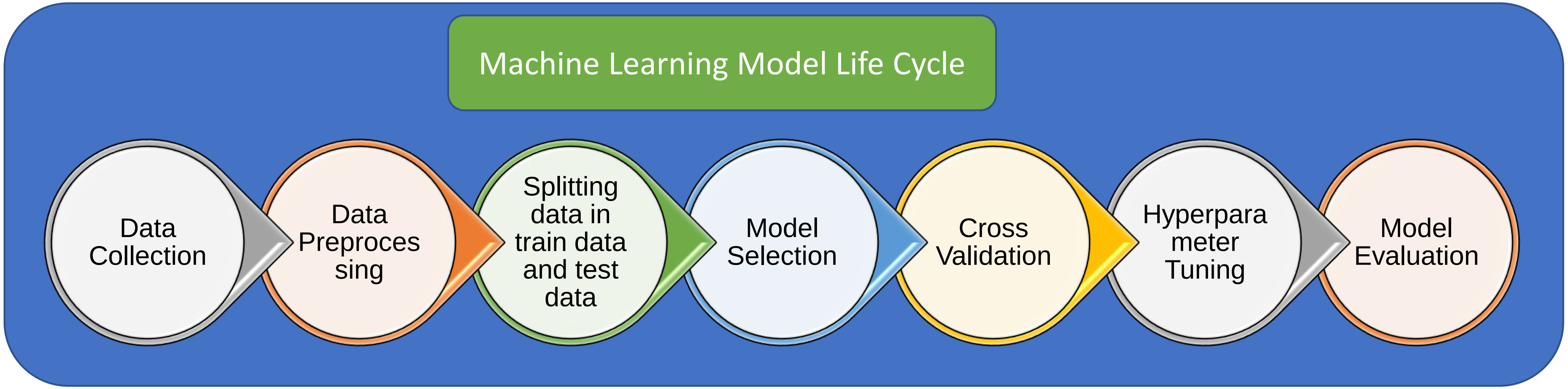
LC = LabelEnconder()

LC.fit_transform(Data)

4. One Hot Encoding : process of converting categorical data variables so they can be provided to machine learning algorithms to improve predictions

Pd.get_dummies()

Supervised Learning



Machine Learning Model Life Cycle

Data Collection

Data Preprocessing

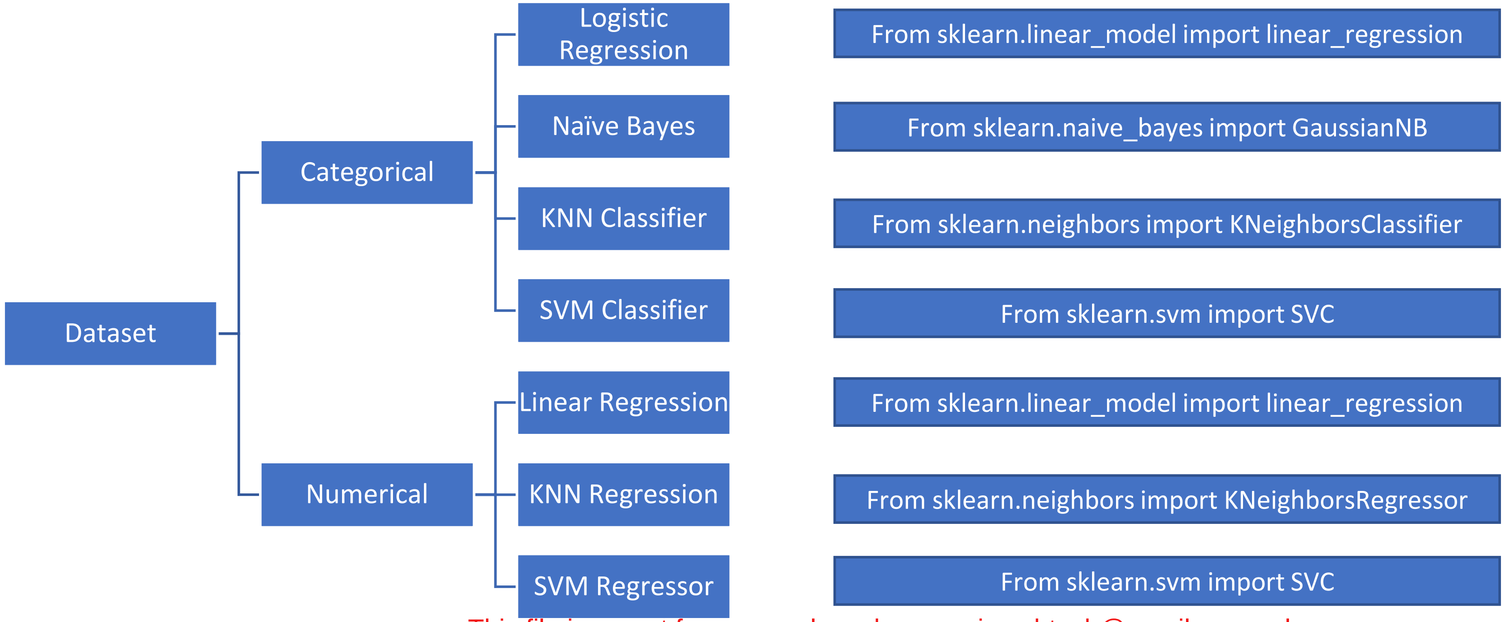
Splitting data in train data and test data

Model Selection

Cross Validation

Hyperparameter Tuning

Model Evaluation



Dataset

Categorical

- Logistic Regression — From sklearn.linear_model import linear_regression
- Naïve Bayes — From sklearn.naive_bayes import GaussianNB
- KNN Classifier — From sklearn.neighbors import KNeighborsClassifier
- SVM Classifier — From sklearn.svm import SVC

Numerical

- Linear Regression — From sklearn.linear_model import linear_regression
- KNN Regression — From sklearn.neighbors import KNeighborsRegressor
- SVM Regressor — From sklearn.svm import SVC

Supervised Learning

Model	Library
Train Test Split	From sklearn.model_selection import train_test_split
Linear Regression	From sklearn.linear_model import LinearRegression
Logistic Regression	From sklearn.linear_model import LogisticRegression
Naïve Bayes	From sklearn.naive_bayes import GaussianNB
K Nearest Neighbor	From sklean.neighbors import KneighborsClassifier From sklean.neighbors import KneighborsRegressor
Support Vector Machine	From sklearn.svm import SVC From sklearn.svm import SVC