Machine Learning

Steps

- ***Problem statement***
- ***Data collection***
- ***Data cleaning***
    - Check categorical values of columns if they have **error in names** using count.value(). If any, replace their names
    - Check the **missing value** and separate them in dataset for easiness
        - Separete numerical columns and categorical columns in two dataframe
        - Check a column has <=25 unique value will be considered as discrete data, otherwise it will be numerical data
        - Replace missing value with mode for categorical & discrete data and median for numerical data.
    - Describe general statistics
    - Reducing **Number of colulmn**
        - Drop any unnecessary categorical column
        - Combine similar numerical columns if it is possible. And dorp their original columns
    -
- ***Model Training***
    - **Train test splitting of data**
        - Separete input as X and output as y using drop function
        - Split X into **X_train & X_test** and **y_train & y_test** using *train_test_split* function from *sklearn.model_selection*
            - mention test data will be 20% or you can adjust its amount based on test performance
    - **Encoding** using ColumnTransfer
        - Separate Categorical and Numerical feature of X
            - Apply OneHotEncoder for categorical feature of X if not many features, Itherwise use LabelEncoding
            - Apply StandardScaler for numerical feature of X
    - **Transformation and Dataframe**
        - Apply transformation on X_train usinf fit.transformer # reason to masking
        - Convert X_train in Dataframe
        - Apply transformer on X_test
        - Conver X_test into DataFrame
    - **Model Training Algorithms**
        - Run RandomForestClassifier on X_train & and y_train import from sklearn.ensemble
        - GIve X_test and X_train data to make prediction using model.prediction # we already know their prediction/output
    - **Performance metrics for both training and testing data**
        - Then we check training performance by giving y_train & predict y_train data to performance metrics

    - Then we check test performance by giving y_test & predict y_test data to performance metrics
    - For Classification Algorithms
      - accuracy_score
      - classification_report
      - precision_score,
      - recall_score
      - f1_score
      - roc_auc_score
      - roc_curve
  - **Hypertuning**
    - Enter parameters into algortthims using model list
    - andomizedSearchCV
    - Then run algorithms
    - Plotting with the ROC AUC Curve