# *Project Report*

**PROJECT NAME:** Merchandise Popularity Prediction Using ML

**TIMELINE:** June–July Batch

**Batch:** AI/ML Intern

| | |
|---|---|
| **OBJECTIVE:** | • This ML Model or program predicts merchandise popularity using a machine learning pipeline.<br>• It starts by loading and inspecting the data, removing duplicates, and checking for missing values.<br>• After conducting exploratory data analysis (EDA) to understand feature relationships and distributions, the data is split into training and testing sets.<br>• four classification models—Decision Tree, Random Forest, SVM, and KNN—are trained on the training data.<br>• The models then make predictions on the test data, and their performance is evaluated using accuracy and classification reports.<br>• The results are compared to determine the best model for predicting merchandise popularity. |
| **REVIEWING DATASET:** | • Loading Libraries and Data: Importing necessary libraries like pandas, numpy, seaborn, and matplotlib. Load the dataset from a CSV file into a DataFrame called train.<br>• Inspecting the Data: view the first and last few rows of the dataset to get an overview of content [Dataset= 18208 x 12]<br>• Exploratory Data Analysis (EDA): Descriptive Statistics[calculating and reviewing the statistical summary of each column.], Correlation Analysis[computing the correlation matrix to understand relationships between features and the target variable.], Distribution Plots[used 'sns.distplot' to visualize the distribution of the target variable 'popularity' And used 'sns.histplot' for better results] |
| **DATA PREPROCESSING:** | • Feature and Target Variables: Defining X as the feature variables (excluding popularity) and y as the target variable.<br>• Train-Test Split: The dataset is split into training and testing sets using train_test_split with 20% of the data reserved for testing. |
| **MODELING:** | • Training and Evaluating four different classification models<br>• Decision Tree: [used DecisionTreeClassifier to train the model., accuracy = 75.46%.]<br>• Random Forest:[used RandomForestClassifier with a random state of 46 to train model, accuracy = 83.89%]<br>• Support Vector Machine (SVM): [used SVC with an RBF kernel, accuracy = 84.07% ]<br>• K-Nearest Neighbors (KNN): [used KNeighborsClassifier with 10 neighbors, accuracy = 83.74%] |
| **MODEL EVALUATION:** | • Decision Tree Classifier: [Mean Accuracy= 75.46%, Standard Deviation= 0.0067, The relatively higher standard deviation suggests that the model's performance varied more across different training/test splits<br>• Random Forest Classifier: [Mean Accuracy=83.89%, Standard Deviation= 0.0015, the model consistently performed well across different datasets, making it a reliable choice]<br>• Support Vector Classifier (SVC): [Mean Accuracy= 84.07%', Standard Deviation=0.0000, The zero standard deviation suggests that the model's performance was extremely consistent across all datasets]<br>• K-Nearest Neighbors Classifier (KNN): [Mean Accuracy= 83.74%, Standard Deviation= 0.0011,<br>• The low standard deviation indicates a consistent performance, though slightly less accurate than the SVC and Random Forest models. |