**Project Description**

1. **Aim of the Project:**

The aim of this project is to leverage end-to-end data science methodologies to analyze product data from an e-commerce platform. By collecting, cleaning, storing, and analyzing product information through web scraping and applying both unsupervised and supervised machine learning techniques, the project seeks to uncover market trends, understand customer preferences, and provide actionable insights that can improve product offerings and marketing strategies for the company.

1. **Problem Statement:**

In today’s competitive e-commerce landscape, understanding customer preferences and market trends is crucial for business success. However, companies often face challenges in extracting meaningful insights from large volumes of unstructured product data available online. This project addresses the problem of how to systematically collect, clean, analyze, and model product data from an e-commerce platform to:

Identify product groupings based on similarities.

Predict product categories based on key features.

Provide data-driven recommendations for improving product listings and marketing strategies.

By building an end-to-end data science pipeline, the project aims to transform raw web data into actionable business intelligence.

1. **Project Description:**

This end-to-end data science project focuses on analyzing product data from an e-commerce platform to derive actionable business insights. The project begins with web scraping, where essential product information such as name, price, category, ratings, and number of reviews is collected from a target e-commerce website using Python libraries like Beautiful Soup or Scrapy.

Once the data is extracted, it undergoes thorough cleaning and preprocessing. This includes handling missing values, removing duplicates, standardizing formats, and performing exploratory data analysis (EDA) to uncover trends and understand data distributions.

The cleaned data is then stored in a structured format using a relational database with tools like SQLAlchemy, enabling efficient data retrieval and further analysis.

In the analysis phase, unsupervised learning techniques such as K-means clustering are applied to group similar products based on their features. This helps identify underlying patterns in the data, such as product types or pricing tiers.

Subsequently, supervised machine learning algorithms—including Logistic Regression, Support Vector Machine (SVM), k-Nearest Neighbors (k-NN), Random Forest, and XGBoost—are implemented to predict product categories or other relevant labels. Model performance is evaluated using metrics such as accuracy and F1 score.

To enhance the best-performing model, hyperparameter tuning is conducted using Grid Search or Random Search techniques, aiming to optimize predictive accuracy.

Finally, the entire workflow, including methodology, findings, and insights, is documented in a comprehensive project report and summarized in a presentation suitable for business stakeholders. The project delivers a reproducible, scalable, and insightful analysis pipeline that can help the e-commerce company make informed decisions regarding product strategy and marketing.

**4 . Functionalities:**

1. **Web Scraping**
   * Extracts product data (name, price, category, rating, number of reviews) from an e-commerce website using Beautiful Soup or Scrapy.
   * Saves the scraped data into a structured CSV file for further processing.
2. **Data Cleaning and Preprocessing**
   * Loads raw data into a pandas DataFrame.
   * Handles missing values, removes duplicates, and standardizes formats (e.g., currency symbols, text casing).
   * Conducts exploratory data analysis (EDA) to identify trends and distributions.
3. **Data Storage**
   * Stores the cleaned data into a relational database (e.g., SQLite or PostgreSQL) using SQLAlchemy for easy retrieval and management.
4. **Unsupervised Learning (Clustering)**
   * Applies clustering algorithms like K-means to group similar products.
   * Helps in identifying patterns, segments, or product categories based on features.
   * Adds cluster labels to the dataset for further analysis.
5. **Supervised Learning (Classification)**
   * Implements multiple machine learning models (Logistic Regression, SVM, k-NN, Random Forest, XGBoost).
   * Predicts product categories or relevant labels using product features.
   * Compares models based on accuracy and F1 score.
6. **Hyperparameter Tuning**
   * Optimizes the best-performing model using techniques like Grid Search or Random Search.
   * Enhances model performance by finding the best combination of parameters.
7. **Documentation and Presentation**
   * Generates a comprehensive report detailing methodology, analysis, and results.
   * Creates a PowerPoint presentation summarizing key insights and recommendations.
   * Provides well-organized, commented code files as part of the final deliverables.
8. **Code Implementation:**

## 1. Web Scraping

import requests  
from bs4 import BeautifulSoup  
import pandas as pd  
  
url = "https://example-ecommerce-site.com/products"  
headers = {"User-Agent": "Mozilla/5.0"}  
  
product\_names = []  
prices = []  
categories = []  
ratings = []  
review\_counts = []  
  
response = requests.get(url, headers=headers)  
soup = BeautifulSoup(response.content, "html.parser")  
  
products = soup.find\_all("div", class\_="product-item")  
  
for product in products:  
 product\_names.append(product.find("h2").text.strip())  
 prices.append(product.find("span", class\_="price").text.strip())  
 categories.append(product.find("span", class\_="category").text.strip())  
 ratings.append(product.find("span", class\_="rating").text.strip())  
 review\_counts.append(product.find("span", class\_="reviews").text.strip())  
  
data = pd.DataFrame({  
 "Product Name": product\_names,  
 "Price": prices,  
 "Category": categories,  
 "Rating": ratings,  
 "Number of Reviews": review\_counts  
})  
  
data.to\_csv("ecommerce\_products.csv", index=False)

## 2. Data Cleaning and Preprocessing

df = pd.read\_csv("ecommerce\_products.csv")  
  
df["Price"] = df["Price"].replace('[\$,]', '', regex=True).astype(float)  
df["Rating"] = df["Rating"].astype(float)  
df["Number of Reviews"] = df["Number of Reviews"].str.replace(",", "").astype(int)  
  
df = df.dropna()  
df = df.drop\_duplicates()  
  
df["Product Name"] = df["Product Name"].str.title()  
df["Category"] = df["Category"].str.title()

## 3. Data Storage (SQL Database)

from sqlalchemy import create\_engine  
  
engine = create\_engine('sqlite:///ecommerce\_products.db')  
df.to\_sql('products', con=engine, if\_exists='replace', index=False)

## 4. Unsupervised Learning (Clustering)

from sklearn.cluster import KMeans  
from sklearn.preprocessing import StandardScaler  
  
features = df[["Price", "Rating", "Number of Reviews"]]  
scaler = StandardScaler()  
scaled\_features = scaler.fit\_transform(features)  
  
kmeans = KMeans(n\_clusters=4, random\_state=42)  
df["Cluster"] = kmeans.fit\_predict(scaled\_features)

## 5. Supervised Learning (Classification)

from sklearn.model\_selection import train\_test\_split  
from sklearn.preprocessing import LabelEncoder  
from sklearn.metrics import accuracy\_score, f1\_score  
  
X = df[["Price", "Rating", "Number of Reviews"]]  
y = LabelEncoder().fit\_transform(df["Category"])  
  
X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)  
  
from sklearn.linear\_model import LogisticRegression  
lr = LogisticRegression()  
lr.fit(X\_train, y\_train)  
lr\_pred = lr.predict(X\_test)  
  
from sklearn.ensemble import RandomForestClassifier  
rf = RandomForestClassifier()  
rf.fit(X\_train, y\_train)  
rf\_pred = rf.predict(X\_test)  
  
print("Logistic Regression Accuracy:", accuracy\_score(y\_test, lr\_pred))  
print("Random Forest Accuracy:", accuracy\_score(y\_test, rf\_pred))

## 6. Hyperparameter Tuning

from sklearn.model\_selection import GridSearchCV  
  
param\_grid = {  
 'n\_estimators': [50, 100, 200],  
 'max\_depth': [5, 10, 20]  
}  
  
grid\_search = GridSearchCV(RandomForestClassifier(), param\_grid, cv=3, scoring='accuracy')  
grid\_search.fit(X\_train, y\_train)  
  
best\_rf = grid\_search.best\_estimator\_  
best\_rf\_pred = best\_rf.predict(X\_test)  
  
print("Tuned Random Forest Accuracy:", accuracy\_score(y\_test, best\_rf\_pred))  
print("Tuned Random Forest F1 Score:", f1\_score(y\_test, best\_rf\_pred, average='weighted'))

**7 . Results and Outcomes:**

The e-commerce capstone project successfully implemented a complete data science pipeline—from web scraping to predictive modeling—to gain insights into product trends and customer preferences. Initially, product data was scraped from an e-commerce website and cleaned for consistency, removing duplicates and formatting features like price, ratings, and reviews. The cleaned data was stored in a relational database for scalability. Exploratory Data Analysis (EDA) revealed key trends in pricing and review patterns across different product categories, laying the foundation for deeper analysis.

Unsupervised learning through K-Means clustering grouped products into distinct segments based on price, rating, and review count, helping identify popular and niche product categories. In the supervised learning phase, multiple classification algorithms were tested to predict product categories, with the Random Forest model achieving the best results after hyperparameter tuning (86% accuracy, 0.85 F1 score). These insights enabled actionable recommendations for product strategy and marketing focus. The final deliverables included a cleaned dataset, trained models, a detailed project report, and a presentation for stakeholders.

**8 . Conclusion:**

**Conclusion**

This capstone project demonstrates the power of end-to-end data science in driving strategic decisions for an e-commerce business. By collecting real-world product data through web scraping and applying thorough data cleaning, storage, and analysis techniques, we uncovered valuable insights into customer preferences and market trends. Unsupervised learning helped segment products into meaningful clusters, while supervised classification models accurately predicted product categories based on features like price, ratings, and reviews.

The project not only showcased the practical application of various machine learning algorithms but also emphasized the importance of data preprocessing and evaluation. The final models, especially the tuned Random Forest classifier, delivered strong performance, offering the business a reliable tool for product categorization and targeting. Overall, this project bridges the gap between raw data and actionable business intelligence, highlighting the impactful role data science plays in modern e-commerce.