**Capstone Project: E-Commerce Product Data Analysis**

**1.Project Objective:**

The primary objective of this capstone project is to analyse e-commerce product data to uncover insights that can help an online retailer improve product offerings and marketing strategies. The project involves a complete end-to-end data science workflow, including data collection through web scraping, cleaning, storage, clustering, classification, and hyperparameter tuning.

**2. Web Scraping:**

* A Python-based web scraping script was developed using the BeautifulSoup library.
* Data was collected from an e-commerce website under the category "Refrigerators."
* Extracted fields included Product Name, Price, Category, Ratings, and Number of Reviews.
* The scraped data was stored in a CSV file named refridgerator\_cleaned.csv.
* Ethical scraping practices were followed by adhering to the website's terms of service.

**3. Data Cleaning:**

* The dataset was loaded into a panda DataFrame for pre-processing.
* Handled missing values using imputation or row deletion depending on significance.
* Removed duplicate and irrelevant entries to ensure data quality.
* Standardized text (e.g., casing) and numerical fields (e.g., currency conversion).
* Conducted exploratory data analysis (EDA) to understand the data distribution and identify trends.

**4. Data Storage:**

* The cleaned data was stored in a relational database using MySQL.
* A database schema was designed to hold product related attributes.
* This ensured data integrity and enabled efficient querying for subsequent analysis.

**5. Unsupervised Learning:**

* Applied K-Means clustering to group similar products based on price, ratings, and reviews.
* Used the Elbow Method to determine the optimal number of clusters.
* Each product was labelled with its respective cluster for further interpretation.
* Clusters revealed insights into product segmentation and pricing strategies.

**6. Supervised Learning:**

* **Objective:** Predict product categories using machine learning models.
* **Algorithms used:**

**1. Logistic Regression**

* **Description**: A linear model used for binary and multiclass classification. It estimates probabilities using the logistic function.
* **Implementation**: Logistic Regression (random state=42)
* **Accuracy Score**: **0.82** (82%)

**2. Support Vector Machine (SVM)**

* **Description**: A powerful classifier that finds the optimal hyperplane to separate classes. Linear kernel was used here.
* **Implementation**: SVC (kernel='linear', random state=42)
* **Accuracy Score**: **0.85** (85%)

**3. k-Nearest Neighbours (k-NN)**

* **Description**: A non-parametric method that classifies a data point based on the majority vote of its nearest neighbours.
* **Implementation**: KNeighborsClassifier(neighbours=5)
* **Accuracy Score**: **0.79** (79%)

**4. Random Forest**

* **Description**: An ensemble model using multiple decision trees to improve accuracy and reduce overfitting.
* **Implementation**: RandomForestClassifier(random\_state=42)
* **Accuracy Score**: **0.88** (88%)

**5. XGBoost**

* **Description**: A gradient boosting framework that builds models in a stage-wise fashion and generalizes them through optimization.
* **Implementation**: XGBClassifier(random\_state=42, use\_label\_encoder=False, eval\_metric="logloss")
* **Accuracy Score**: **0.90** (90%)

**7. Hyperparameter Tuning:**

1. Performed **Grid Search** to explore multiple combinations of hyperparameters for the Random Forest model.
2. Tuned parameters included:

* n\_estimators: number of trees in the forest
* max\_depth: maximum depth of the tree
* min\_samples\_split: minimum number of samples required to split an internal node
* min\_samples\_leaf: minimum number of samples required to be at a leaf node

1. Identified the optimal parameter combination:  
   {'n\_estimators': 150, 'max\_depth': 30, 'min\_samples\_split': 2, 'min\_samples\_leaf': 1}
2. The final tuned **Random Forest model** achieved a **test set accuracy of 92%**, showing notable improvement over the baseline.
3. This process enhanced the model's robustness and helped avoid overfitting by ensuring it generalized well to unseen data.

**8. Key Insights and Results:**

* Clustering helped identify low-rated and high-rated product groups.
* Supervised models could accurately predict categories with up to 90%+ accuracy.
* Price and ratings were among the most influential features in clustering and classification.
* Recommendations were made to promote specific product groups based on popularity and review volume.

**9. Tools and Technologies Used:**

* Python (pandas, numpy, scikit-learn, xgboost, matplotlib, seaborn, BeautifulSoup)
* Jupiter Notebook
* SQL (MySQL)
* CSV and Relational Databases

**10. Conclusion:**

This capstone project demonstrates a complete application of the data science lifecycle, from data acquisition to insight generation. By integrating web scraping, data pre-processing, machine learning, and hyperparameter tuning, actionable insights were provided for real-world business decisions in the e-commerce domain.