**UCI Supermarket Sales Dataset: Project Requirements Document**

**Dataset Overview**

The UCI Supermarket Sales dataset includes information on individual sales transactions at a supermarket. Each record represents a customer purchase and includes details on branch location, product type, and sales metrics. The goal of this project is to predict the Total Sales value based on customer and product attributes by implementing data cleanup, exploratory data analysis (EDA), hypothesis testing, and machine learning modeling.

**Dataset Description**

**Features Overview:**

1. **Invoice ID** - Unique identifier for each transaction.
2. **Branch** - Store branch where the sale was made (A, B, or C).
3. **City** - City in which the branch is located.
4. **Customer Type** - Type of customer (e.g., Member or Normal).
5. **Gender** - Gender of the customer (Male or Female).
6. **Product Line** - Category of product sold (e.g., Health and Beauty, Food and Beverages).
7. **Unit Price** - Price per unit of the product.
8. **Quantity** - Quantity of units purchased in the transaction.
9. **Tax** - Tax applied to the purchase.

**Project Objectives**

1. **Data Preparation and Cleanup**
   * Handle missing values in relevant columns (if any).
   * Convert categorical variables (e.g., Branch, City) to numeric representations as needed.
   * Identify and treat outliers in numerical columns like Unit Price, Quantity, and Tax to prevent skewed analysis or model performance.
2. **Feature Engineering**
   * Create a new feature Total Sales by calculating Unit Price \* Quantity + Tax for each transaction.
   * Consider additional aggregations, such as average sales per product line or branch.
3. **Exploratory Data Analysis (EDA)**
   * Generate descriptive statistics for each feature to understand data distribution.
   * Visualize key features (e.g., Product Line, Total Sales) with bar charts, histograms, and box plots.
   * Create a correlation heatmap to explore relationships between numerical features like Quantity, Tax, and Total Sales.
4. **Hypothesis Testing**
   * Perform statistical hypothesis tests to investigate:
     + Whether there is a significant difference in Total Sales between customer types (Member vs. Normal) using a t-test.
     + The correlation between Quantity and Total Sales using Pearson’s correlation test.
     + The association between Product Line and Gender using a chi-square test.
5. **Machine Learning Modeling**
   * Split the dataset into training and testing sets (e.g., 80-20 split).
   * Build and evaluate multiple machine learning models to predict Total Sales:
     + **Linear Regression** – Establish a baseline model for sales prediction.
     + **Decision Tree Regressor** – Tune hyperparameters to optimize model performance.
   * Evaluate models based on Mean Absolute Error (MAE), Mean Squared Error (MSE), and R-squared metrics.
   * Identify the best-performing model and analyze its results, do a cross validation if necessary
6. **Model Interpretation and Insights**
   * Examine feature importance to understand the most influential attributes on Total Sales.
   * Create partial dependence plots for top features to interpret their effect on predicted sales.
   * Summarize findings and insights in a report format, highlighting patterns in sales across branches, product lines, or customer types.
7. **Python API using Flask**
   * Develop a simple Flask REST API server for predicting the values
   * Use POSTMAN or similar technique to test it

**Project Timeline**

**Total Duration**: 4 hours

| **Task** | **Description** | **Estimated Time** |
| --- | --- | --- |
| **Data Preparation and Cleanup** | Clean missing data, handle data types and outliers | 45 minutes |
| **Exploratory Data Analysis** | Visualize and analyze feature relationships | 45 minutes |
| **Hypothesis Testing** | Conduct statistical tests on selected features | 30 minutes |
| **Machine Learning Modeling** | Build, train, and evaluate ML models | 1 hour |
| **Model Interpretation and Insights** | Analyze model importance and create final report | 30 minutes |
| **API Development** | Python FLASK API | 30 mins |

**Submission Requirements**

1. **Python code and Jupyter Notebook** containing all code, visualizations, and comments explaining each step.
2. **Summary Report** detailing:
   * Key findings from EDA and hypothesis tests.
   * Comparison of model performance metrics.
   * Interpretation of the best model and insights into influential features.
   * Presentation for 10 minutes