**Data Science Project on BigMart Sales Prediction Project Methodology**

**Overview**

Sales forecasting is critical for businesses to effectively allocate resources, manage cash flow, and optimize inventory. By accurately predicting product sales across various stores, businesses can better meet customer demand and optimize stock levels. This **BigMart Sales Prediction Project** aims to leverage machine learning to predict sales, enabling BigMart to understand the underlying product and store attributes that drive sales. With a diverse dataset, this project will explore advanced data processing, feature engineering, and machine learning techniques to improve sales forecasting accuracy.

**Objectives**

The primary objective of this project is to develop, compare, and evaluate predictive models that forecast product sales at individual stores. The project will provide actionable insights into which product and store attributes influence sales, enabling BigMart to make data-driven decisions to enhance business strategies.

**Technical Stack**

* **Programming Language**: Python
* **Database**: Amazon Redshift
* **Python Libraries**:
* **Data Manipulation**: Pandas, NumPy
* **Data Visualization**: Matplotlib, Seaborn
* **Machine Learning**: Scikit-Learn
* **Database Connectivity**: Redshift Connector
* **Cloud Infrastructure**: AWS (Amazon Web Services)
* **Storage**: S3 for storing raw data
* **Compute**: Redshift Cluster for database storage and processing

**Methodology**

**1. Data Exploration with Amazon Redshift**

The data was initially stored in **AWS S3** and then loaded into the **Amazon Redshift cluster** database, enabling efficient querying and storage within a scalable cloud environment. Data exploration includes basic SQL queries to retrieve the dataset and assess data quality, understanding the distribution of key features, and identifying any data imbalances.

**2. Data Cleaning and Imputation**

Data cleaning is crucial for handling missing or inconsistent values. This project addresses missing data using SQL-based imputation techniques:

* **Imputation** of missing values for continuous variables (e.g., item\_weight).
* **Normalization** of categorical variables like item\_fat\_content.

SQL queries are used to streamline this process directly within Amazon Redshift, improving performance when dealing with large datasets.

**3. Exploratory Data Analysis (EDA)**

Exploratory Data Analysis (EDA) is essential for understanding relationships within the dataset. This includes:

* **Categorical Data Analysis**: Examines variables like item\_type and outlet\_type to understand how product categories and store types impact sales.
* **Continuous Data Analysis**: Reviews variables such as item\_mrp and item\_visibility for trends.
* **Correlation Analysis**:
  + **Pearson Correlation** for continuous variables.
  + **Chi-squared Test** and **Cramer’s V** for categorical correlations.
  + **ANOVA (Analysis of Variance)** to assess the relationship between categorical predictors and the target variable.

**4. Feature Engineering**

Feature engineering involves creating new features that enhance the predictive power of the dataset:

* **Outlet Age**: Calculated as the difference between the current year and outlet\_establishment\_year to understand how store age affects sales.
* **Label Encoding**: Converts categorical features into numerical format, ensuring compatibility with machine learning algorithms.

**5. Data Splitting**

The dataset is split into training and test sets, ensuring a balanced representation for model training and validation.

**6. Model Building and Evaluation**

This project evaluates multiple machine learning models, focusing on both traditional statistical models and more advanced ensemble techniques. Models include:

* **Linear Regressor**: Establishes a baseline for predictive accuracy.
* **Elastic Net Regressor**: A hybrid approach combining Ridge and Lasso regularization for a balanced model.
* **Random Forest and Extra Trees Regressors**: Ensemble methods that improve accuracy by reducing overfitting.
* **Gradient Boosting Regressor**: A powerful algorithm for minimizing errors through iterative boosting.
* **Multilayer Perceptron (MLP) Regressor**: A deep learning approach for capturing complex patterns.
* **Spline Regressor**: Capture non-linear relationships in the data.

**Advanced Ensemble Techniques**:

* **Voting Regressor**: Combines predictions from multiple models to improve accuracy.
* **Stacking Regressor**: Trains multiple models and then combines their outputs in a final model.
* **Model Blending**: Integrates predictions from several models with a weighted approach to improve overall prediction robustness.

**7. Evaluation Metrics**

The models are evaluated using **R-squared** metrics to assess prediction accuracy. The chosen metric reflects the degree to which the model explains variability in sales data, which is critical for effective forecasting.

**Modular Code Structure**

The project code is organized into several modular components for scalability and maintainability:

1. **data**: Contains the dataset files.
2. **docs**: Contains files with information related to data description & project methodology.
3. **notebook**: Houses reference scripts, including Jupyter Notebooks, for initial development.
4. **src**: Includes functionally organized scripts that handle data loading, data preprocessing, feature engineering, data visualization, model building, hyperparameter tuning & model evaluation.
5. **main.py**: Main script that calls functions from the ML pipeline, enabling a streamlined model training and evaluation process.
6. **output**: Contains the pkl files of two best performing models i.e., Stacking model & Bleanding model.
7. **requirements.txt**: Lists all required libraries and versions, installable via pip install -r requirements.txt.

**Project Takeaways**

1. Analyzing sales prediction data and defining a problem statement.
2. Conducting data exploration and cleaning using Amazon Redshift and SQL.
3. Performing comprehensive EDA, including correlation analyses and hypothesis testing.
4. Building and evaluating machine learning models, from basic linear regression to complex ensemble models.
5. Using ensemble techniques like stacking and blending to enhance predictive accuracy.
6. Understanding and applying regression metrics for model evaluation.

**Conclusion**

The BigMart Sales Prediction project provides a comprehensive framework for predicting product sales in a multi-store environment. By employing SQL for data extraction, feature engineering, and a robust set of machine learning models, this project delivers insights into which product and store attributes influence sales the most. This knowledge empowers BigMart to make data-driven business decisions and enhances predictive capabilities for better inventory and marketing strategies.