Sales Prediction Documentation

Introduction

Sales prediction is a crucial task for businesses as it helps in strategic decision-making, resource allocation, and revenue forecasting. Leveraging machine learning techniques for sales prediction offers valuable insights into the relationship between advertising expenses and sales, enabling businesses to optimize their marketing strategies effectively. This documentation provides a comprehensive overview of a Python code implemented using Scikit-learn for sales prediction, covering data preprocessing, exploratory data analysis (EDA), model training, prediction, and evaluation.

Importing Libraries

The code initiates by importing essential libraries required for data manipulation, visualization, and machine learning tasks.

- **Pandas:** Used for handling datasets and data manipulation.
- **NumPy:** Essential for numerical operations and array manipulations.
- **Seaborn and Matplotlib:** These libraries are employed for data visualization, facilitating insights into the dataset's structure and relationships.
- **Scikit-learn:** Utilized for implementing machine learning algorithms, model training, and evaluation.

Loading Dataset

The dataset provided by AFAME TECHNOLOGIES is loaded using Pandas' **read_csv** function. The dataset comprises 200 rows of sales data with features including advertising expenses on TV, radio, and newspaper. Loading the dataset is the initial step towards understanding its structure, characteristics, and preparing it for further analysis.

Pre-processing Data

Data pre-processing is a critical step to ensure data quality and enhance model performance. In this code, pre-processing involves:

1. **Data Augmentation:** To increase data counts and enhance model robustness, random noise is added to the existing data. This process is implemented using a custom function **augment_data**, which introduces noise while preserving the correlation between features.
2. **Combining Data:** The augmented data is then combined with the original dataset to form a larger dataset for model training. This step ensures that the model learns from a diverse set of data points, improving its generalization ability.

Exploratory Data Analysis (EDA)

EDA plays a pivotal role in understanding the underlying patterns and relationships within the dataset. Key EDA steps in this code include:

1. **Data Description:** Summary statistics such as mean, standard deviation, and quartiles are computed using the **describe** function to provide insights into the dataset's distribution and variability.
2. **Correlation Analysis:** Correlation matrices and heatmaps are generated to visualize the relationships between features (advertising expenses) and the target variable (sales). This analysis helps identify

which advertising channels have a stronger influence on sales and informs feature selection for model training.

Model Training, Prediction, and Evaluation

The core of the code involves training machine learning models to predict sales based on advertising expenses. Multiple regression algorithms are employed, and their performance is evaluated using various evaluation metrics. The steps include:

1. **Linear Regression:** A simple linear regression model is trained to establish a linear relationship between advertising expenses and sales. Model evaluation is performed using metrics such as Mean Squared Error (MSE), Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and R-squared (R^2) score.

2. **Polynomial Regression:** To capture nonlinear relationships between features and the target variable, polynomial regression is employed. Polynomial features are created, and a linear regression model is trained on these features. The model's performance is evaluated similarly to linear regression.

3. **Gradient Boosting Regressor:** Ensemble methods, particularly gradient boosting, are utilized to build a predictive model. The model's hyperparameters are tuned to optimize performance, and evaluation metrics are computed to assess its effectiveness.

4. **Support Vector Machine (SVM):** A support vector machine model with a polynomial kernel is trained to predict sales. The model's performance is evaluated using standard regression evaluation metrics.

5. **Neural Network:** A feedforward neural network is constructed using TensorFlow-Keras. The model architecture comprises multiple dense layers with rectified linear unit (ReLU) activation functions. The model is trained using the Adam optimizer and evaluated based on MSE, RMSE, MAE, and R^2 score.

By- Abhinav Mishra

8601860227

abhinavmishra@tuta.io