GRT INSTITUTE OF ENGINEERING AND TECHNOLOGY, TIRUTTANI - 631209

Approved by AICTE, New Delhi Affiliated to Anna University, Chennai

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

PROJECT TITLE

Future sales prediction

COLLEGE CODE:1103

Aswini.M

3rd yr, 5th sem

Reg no:110321104004

aswini30012004@gmail.com

Explanation:

Sales prediction is a process of using data and statistical methods to forecast future sales for a product or service. To create a sales prediction model, you typically need a dataset that includes historical sales data along with relevant features. Here's an explanation of the key elements in a sales prediction dataset:

Date/Time:

This is the timestamp when each sale occurred. It helps in capturing seasonality and trends over time.

Sales Amount:

The target variable, which is the actual sales figure for each time period. This is what you want to predict.

Features:

Product Attributes:

Information about the product being sold, such as category, price, brand, and any special promotions or discounts.

Store Information:

Details about the store where the sale took place, such as location, size, and any specific store-level promotions.

Customer Data:

If available, data on customers, including demographics, loyalty programs, or previous purchase history.

External Factors:

These could include economic indicators (e.g., GDP), weather data, and other external factors that might influence sales.

Lagged Variables:

Past sales data for the same or related products, which can help capture dependencies and seasonality.

Categorical Variables:

Variables that aren't numerical, like product category, store location, or day of the week. These need to be encoded for machine learning models.

Holidays and Special Events:

Information about holidays, special promotions, or events that might impact sales.

Competitor Data:

Data on competitors' pricing or promotions can also be relevant if it's available.

Once you have a dataset with these components, you can use various machine learning and statistical techniques to build a sales prediction model. This might include linear regression, time series analysis, or more advanced techniques like neural networks for deep learning.

The goal is to use this data to train a model that can accurately predict future sales based on the chosen features. Keep in mind that data quality and feature selection are crucial for the accuracy of your predictions.

Details about data:

To predict future sales, you can create a dataset with columns for TV advertising spending, radio advertising spending, newspaper advertising spending, and sales.

These columns will help you build a regression model to predict sales based on advertising expenditures. Here are some details about each of these columns:

TV:

Column Name: TV

Data Type: Numeric (continuous)

Description: This column represents the amount of money spent on advertising through television channels. It includes expenses for television commercials, sponsorships, and other TV-related advertising efforts. Measured in dollars.

Radio:

Column Name: Radio

Data Type: Numeric (continuous)

Description: This column represents the amount of money spent on advertising through radio channels. It includes expenses for radio commercials, radio show sponsorships, and other radio-related advertising efforts. Measured in dollars

Newspaper:

Column Name: Newspaper

Data Type: Numeric (continuous)

Description: This column represents the amount of money spent on advertising in newspapers. It includes expenses for print advertisements, classified ads, and other newspaper-related advertising efforts. Measured in dollars.

Sales:

Column Name: Sales

Data Type: Numeric (continuous)

Description: This is the target column you want to predict. It represents the actual sales revenue generated as a result of the advertising expenditures on TV, radio, and newspaper. Measured in dollars

Begin building the project by loading the dataset:

The columns involved in the dataset are:

- Tv
- Radio
- Newspaper
- Sales

Importing necessary libraries:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error,r2_score
from distributed import *
```

loading the dataset:

```
data=pd.read_csv("C:/sales.csv")
print(data)
```

preprocessing the data:

Data Collection:

Gather historical sales data, including variables like time, sales quantity, price, and any relevant features.

Future selection:

Identify the most relevant features and create new ones if needed.

```
data = data.drop(['newspaper'], axis=1)
```

```
scaler = StandardScaler()
data[['TV', 'radio']] = scaler.fit_transform(data[['TV', 'radio']])
```

train-test-split:

Split your data into a training set and a testing set. Train your selected model using the training data.

```
X = data[['TV', 'radio']]
y = data['sales']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42)
```

Modeling:

```
model = LinearRegression()
model.fit(X_train, y_train)
```

Evaluate the Model:

Use evaluation metrics like Mean Absolute Error (MAE), Mean Squared Error (MSE), or Root Mean Squared Error (RMSE) to assess the model's performance.

```
y_pred = model.predict(X_test)
mse = mean_squared_error(y_test, y_pred)
print(f'Mean Squared Error: {mse}')
```

Make Future Sales Predictions:

Use the trained model to predict future sales based on new data.

```
future_data = pd.read_csv('new_data.csv')
future_data[['TV', 'radio']] = scaler.transform(future_data[['TV', 'radio']])
future_sales = model.predict(future_data[['TV', 'radio']])
print(f'Predicted Future Sales: {future_sales}')
```

performing analysis:

```
# Import necessary libraries
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score
import matplotlib.pyplot as plt

# Load your sales data (replace 'sales_data.csv' with your dataset)
data = pd.read_csv('sales_data.csv')

# Data preprocessing
X = data[['Feature1', 'Feature2']]  # Use relevant features
y = data['Sales']  # Sales column to predict

# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
```

```
random_state=42)

# Create and train a linear regression model
model = LinearRegression()
model.fit(X_train, y_train)

# Make predictions on the test set
y_pred = model.predict(X_test)

# Evaluate the model
mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)
print(f"Mean Squared Error: {mse}")
print(f"R-squared: {r2}")

# Plot the predictions
plt.scatter(y_test, y_pred)
plt.xlabel("Actual Sales")
plt.ylabel("Predicted Sales")
plt.title("Actual Sales vs. Predicted Sales")
plt.show()

# Predict future sales (replace the features with your input values)
future_sales = model.predict(np.array([[value1, value2]]))

print(f"Predicted Future Sales: {future_sales[0]}")
```