

▼ Task_2

Install Required Libraries

```
import pandas as pd
from sklearn.preprocessing import OneHotEncoder
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_absolute_error, mean_squared_error
import numpy as np

# Load the training and testing datasets
train_data = pd.read_csv('train.csv')
test_data = pd.read_csv('test.csv')

# Remove rows with missing target variable in the training data
train_data_cleaned = train_data.dropna(subset=['units'])

# Extract features and target variable for training (excluding ad_spend)
train_features = train_data_cleaned[['Item Id', 'date']]
train_target = train_data_cleaned['units']

# Extract features from the test data
test_features = test_data[['Item Id', 'date']]

# Feature Engineering: Extract date-related features
train_features['date'] = pd.to_datetime(train_features['date'])
train_features['day_of_week'] = train_features['date'].dt.dayofweek
train_features['month'] = train_features['date'].dt.month
train_features.drop(columns=['date'], inplace=True)

test_features['date'] = pd.to_datetime(test_features['date'])
test_features['day_of_week'] = test_features['date'].dt.dayofweek
test_features['month'] = test_features['date'].dt.month
test_features.drop(columns=['date'], inplace=True)

# Convert categorical variables to numerical using OneHotEncoding
encoder = OneHotEncoder(handle_unknown='ignore', sparse=False)
encoded_train_item_id = encoder.fit_transform(train_features[['Item Id']])
encoded_test_item_id = encoder.transform(test_features[['Item Id']])

encoded_train_item_id_df = pd.DataFrame(encoded_train_item_id, columns=encoder.get_feature_names_out(['Item Id']))
encoded_test_item_id_df = pd.DataFrame(encoded_test_item_id, columns=encoder.get_feature_names_out(['Item Id']))

# Concatenate the encoded columns with the original data
train_features = pd.concat([train_features.reset_index(drop=True).drop(columns=['Item Id']), encoded_train_item_id_df], axis=1)
test_features = pd.concat([test_features.reset_index(drop=True).drop(columns=['Item Id']), encoded_test_item_id_df], axis=1)

# Handle potential NaN values after one-hot encoding - Impute missing values
train_features.fillna(0, inplace=True)
test_features.fillna(0, inplace=True)

# Check shapes to verify consistency
print("Shape of train_features:", train_features.shape)
print("Shape of train_target:", train_target.shape)

# Train a simple linear regression model
model = LinearRegression()
model.fit(train_features, train_target)

# Predict on the test set
test_predictions = model.predict(test_features)

# If you have actual units in the test set, evaluate the model
if 'units' in test_data.columns:
    test_actual = test_data['units']
    mae = mean_absolute_error(test_actual, test_predictions)
    rmse = np.sqrt(mean_squared_error(test_actual, test_predictions))
    mse = mean_squared_error(test_actual, test_predictions) # Calculate MSE
    print("MAE:", mae)
    print("RMSE:", rmse)
    print("MSE:", mse) # Print MSE value
else:
    test_data['predicted_units'] = test_predictions
```

```
test_data[ predicted_units ] = test_predictions
test_data.to_csv('path_to_save_predictions.csv', index=False)
print("Predictions saved to path_to_save_predictions.csv")
```

 <ipython-input-10-355c83606e64>:16: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-c
train_features['date'] = pd.to_datetime(train_features['date'])

<ipython-input-10-355c83606e64>:17: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-c
train_features['day_of_week'] = train_features['date'].dt.dayofweek

<ipython-input-10-355c83606e64>:18: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-c
train_features['month'] = train_features['date'].dt.month

<ipython-input-10-355c83606e64>:19: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-c
train_features.drop(columns=['date'], inplace=True)

<ipython-input-10-355c83606e64>:21: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-c
test_features['date'] = pd.to_datetime(test_features['date'])

<ipython-input-10-355c83606e64>:22: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-c
test_features['day_of_week'] = test_features['date'].dt.dayofweek

<ipython-input-10-355c83606e64>:23: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-c
test_features['month'] = test_features['date'].dt.month

<ipython-input-10-355c83606e64>:24: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-c
test_features.drop(columns=['date'], inplace=True)

/usr/local/lib/python3.10/dist-packages/sklearn/preprocessing/_encoders.py:975: FutureWarning: `sparse` was renamed to `sparse_output`
warnings.warn(
Shape of train_features: (83592, 217)
Shape of train_target: (83592,)
Predictions saved to path_to_save_predictions.csv