

Sales Forecast Prediction — Python + XGBoost

1. Imports & Setup pandas, matplotlib, seaborn, xgboost and scikit learn.

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import xgboost as xgb
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error
```

2. load features like Row ID, Order ID, Customer ID, Product ID, etc.

```
file_path = 'train.csv'
data = pd.read_csv(file_path)
data.head()
```

	Row ID	Order ID	Order Date	Ship Date	Ship Mode
Customer ID \					
0	1	CA-2017-152156	08/11/2017	11/11/2017	Second Class
CG-12520					
1	2	CA-2017-152156	08/11/2017	11/11/2017	Second Class
CG-12520					
2	3	CA-2017-138688	12/06/2017	16/06/2017	Second Class
DV-13045					
3	4	US-2016-108966	11/10/2016	18/10/2016	Standard Class
S0-20335					
4	5	US-2016-108966	11/10/2016	18/10/2016	Standard Class
S0-20335					

	Customer Name	Segment	Country	City
State \				
0	Claire Gute	Consumer	United States	Henderson
Kentucky				
1	Claire Gute	Consumer	United States	Henderson
Kentucky				
2	Darrin Van Huff	Corporate	United States	Los Angeles
California				
3	Sean O'Donnell	Consumer	United States	Fort Lauderdale
Florida				
4	Sean O'Donnell	Consumer	United States	Fort Lauderdale
Florida				

	Postal Code	Region	Product ID	Category	Sub-
Category \					
0	42420.0	South	FUR-B0-10001798	Furniture	Bookcases

1	42420.0	South	FUR-CH-10000454	Furniture	Chairs
2	90036.0	West	OFF-LA-10000240	Office Supplies	Labels
3	33311.0	South	FUR-TA-10000577	Furniture	Tables
4	33311.0	South	OFF-ST-10000760	Office Supplies	Storage

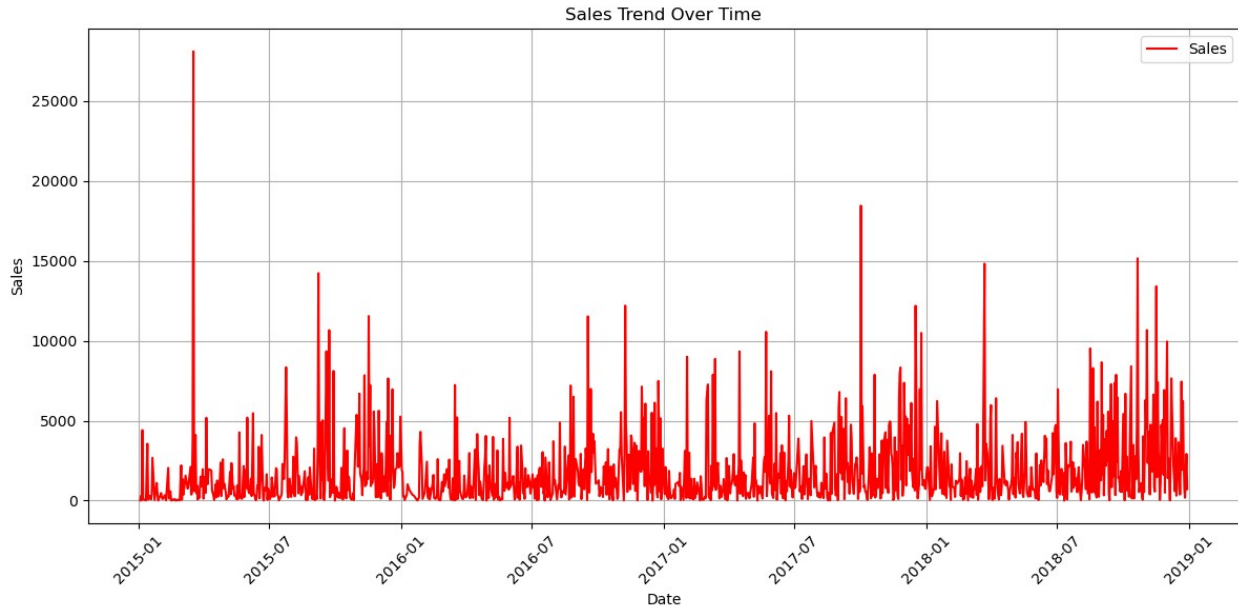
	Product Name	Sales
0	Bush Somerset Collection Bookcase	261.9600
1	Hon Deluxe Fabric Upholstered Stacking Chairs,...	731.9400
2	Self-Adhesive Address Labels for Typewriters b...	14.6200
3	Bretford CR4500 Series Slim Rectangular Table	957.5775
4	Eldon Fold 'N Roll Cart System	22.3680

3. Data Preprocessing and Visualization

```

data['Order Date'] = pd.to_datetime(data['Order
Date'],format='%d/%m/%Y')
sales_by_date = data.groupby('Order Date')
['Sales'].sum().reset_index()
plt.figure(figsize=(12,6))
plt.plot(sales_by_date['Order
Date'],sales_by_date['Sales'],label='Sales',color='red')
plt.title('Sales Trend Over Time')
plt.xlabel('Date')
plt.ylabel('Sales')
plt.grid(True)
plt.legend()
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()

```



4. Feature Engineering - Creating Lagged Features

```
def create_lagged_features(data, lag=1):
    lagged_data = data.copy()
    for i in range(1, lag+1):
        lagged_data[f'lag_{i}'] = lagged_data['Sales'].shift(i)
    return lagged_data

lag = 5
sales_with_lags = create_lagged_features(data[['Order Date', 'Sales']], lag)
sales_with_lags = sales_with_lags.dropna()
```

5. Preparing the Data for Training

```
X = sales_with_lags.drop(columns=['Order Date', 'Sales'])
y = sales_with_lags['Sales']

X_train, X_test, y_train, y_test =
train_test_split(X, y, test_size=0.2, shuffle=False)
```

6. Training the XGBoost Model

```
model =
xgb.XGBRegressor(objective='reg:squarederror', n_estimators=100, learning_rate=0.1, max_depth=5)
model.fit(X_train, y_train)

XGBRegressor(base_score=None, booster=None, callbacks=None,
              colsample_bylevel=None, colsample_bynode=None,
              colsample_bytree=None, device=None,
              early_stopping_rounds=None,
```

```

        enable_categorical=False, eval_metric=None,
feature_types=None,
        feature_weights=None, gamma=None, grow_policy=None,
importance_type=None, interaction_constraints=None,
learning_rate=0.1, max_bin=None, max_cat_threshold=None,
max_cat_to_onehot=None, max_delta_step=None, max_depth=5,
max_leaves=None, min_child_weight=None, missing=nan,
monotone_constraints=None, multi_strategy=None,
n_estimators=100,
        n_jobs=None, num_parallel_tree=None, ...)

```

7. Making Predictions and Evaluating the Model

```

y_pred = model.predict(X_test)
rmse_xgb = np.sqrt(mean_squared_error(y_test,y_pred))
print(f"RMSE: {rmse_xgb:.2f}")

```

RMSE: 734.63

8. Visualizing Results

```

plt.figure(figsize=(12,6))
plt.plot(y_test.index,y_test,label='Actual Sales',color='red')
plt.plot(y_test.index,y_pred,label='Predicted Sales',color='green')
plt.title('Sales Forecasting using XGBoost')
plt.xlabel('Date')
plt.ylabel('Sales')
plt.legend()
plt.grid(True)
plt.tight_layout()
plt.show()

```

