PROJECT REPORT

E-Commerce Data Analysis and Sales Prediction using Machine Learning

Submitted by

Ravindra Singh

Acknowledgement

I sincerely thank **YBI Foundation** for giving me the opportunity to work on this internship project titled *"E-Commerce Data Analysis and Sales Prediction using Machine Learning."*

I am especially grateful to my mentors and the training team for their encouragement, valuable insights, and continuous guidance throughout the project. Their support helped me to apply my theoretical knowledge of Python, Data Analysis, and Machine Learning into practical use.

I also acknowledge my peers and family members who motivated me during this journey. This internship has been a great learning experience, and it has improved my skills in data handling, visualization, and predictive modeling.

Ravindra Singh

(19894000003205713)

Declaration

I, **Ravindra Singh**, declare that the internship project report titled *"E-Commerce Data Analysis and Sales Prediction using Machine Learning"* is an original piece of work completed by me during my internship at **YBI Foundation**.

The data analysis, visualizations, and machine learning models presented in this project have been implemented by me using Python and related tools. To the best of my knowledge, this project does not contain any material that has been published or submitted for any degree, diploma, or certificate elsewhere.

Ravindra Singh

(198940000032O5713)

Introduction to Machine Learning

Machine Learning (ML) is a branch of Artificial Intelligence (AI) that enables computer systems to learn patterns from data and make decisions or predictions without being explicitly programmed. Instead of following fixed rules, machine learning models improve automatically through experience and exposure to more data.

In the modern digital world, a massive amount of data is generated every second. Machine Learning provides powerful techniques to analyse this data, uncover hidden patterns, and generate useful insights. It plays a key role in various fields such as e-commerce, healthcare, banking, manufacturing, and social media.

There are mainly three types of machine learning:

1. **Supervised Learning** – The model learns from labeled data to make predictions (e.g., sales forecasting, spam detection).
2. **Unsupervised Learning** – The model finds hidden structures or groups in unlabeled data (e.g., customer segmentation).
3. **Reinforcement Learning** – The model learns by interacting with an environment and receiving rewards or penalties (e.g., robotics, gaming).

In this project, we apply **Supervised Learning** techniques to analyze e-commerce sales data and build predictive models. By using regression algorithms, we aim to predict sales amounts and identify key factors affecting customer purchases.

Conclusion

This project *"E-Commerce Data Analysis and Sales Prediction using Machine Learning"* helped in understanding customer purchase behavior, sales trends, and payment preferences. Through Exploratory Data Analysis (EDA), we identified the most profitable categories, top-performing regions, and customer payment patterns.

Further, Machine Learning models such as Linear Regression, Decision Tree, and Random Forest were applied to predict sales (total_price). The models were evaluated using performance metrics like MAE, RMSE, and R^2 score. Among them, Random Forest performed better in terms of accuracy and prediction capability.

Overall, this project enhanced my skills in Python programming, data handling, visualization, and machine learning. It provided practical exposure to how businesses can use data-driven insights to improve sales strategies and customer satisfaction.

Future Scope

- **Advanced Models:** More advanced machine learning techniques like XGBoost, LightGBM, or Deep Learning models can be applied for better predictions.
- **Recommendation System:** A recommendation engine can be built to suggest products to customers based on past purchases.
- **Real-Time Analysis:** Live dashboards using tools like Power BI or Tableau can be created for real-time e-commerce monitoring.
- **Customer Segmentation:** Unsupervised learning methods (e.g., clustering) can be applied to group customers by behavior and design personalized marketing strategies.
- **Scalability:** The project can be expanded to larger datasets and integrated with cloud platforms for big data handling.

1. Problem Statement

E-commerce businesses generate a large amount of sales and customer data every day. It is important to analyze this data to understand customer behavior, popular products, regional demand, and payment preferences. The main problem is how to use this data effectively to predict sales and support better decision-making.

2. Dataset Description

Dataset source -
https://github.com/YBIFoundation/Pandas/raw/refs/heads/main/EcommerceEDA.csv

The dataset contains order-level details from an e-commerce platform. It includes customer information, product categories, sales transactions, and payment methods.

Dataset variables

- order_id – Unique order identification
- customer_id – Unique customer identification
- order_date – Date of order placed
- category – Product category
- product – Product name
- price – Price per unit
- quantity – Number of items purchased
- total_price – Total sales amount (price × quantity)
- state – State of the customer
- region – Regional location (North, South, East, West)
- payment_method – Mode of payment (e.g., COD, UPI, Credit Card)

3. Objective

- To clean and prepare the dataset for analysis.
- To explore the data and identify important patterns.
- To analyze sales by product, category, state, region, and payment method.
- To build a machine learning model that predicts sales (total_price).
- To provide insights for business growth and customer satisfaction.

4. Methodology

1. **Data Preprocessing** – Handle missing values, encode categorical features, and prepare the dataset.
2. **Exploratory Data Analysis (EDA)** – Use charts and graphs to understand customer and sales behavior.
3. **Feature Engineering** – Create new features to improve prediction.
4. **Model Development** – Apply regression models such as Linear Regression, Decision Tree, and Random Forest.
5. **Model Evaluation** – Compare model performance using MAE, RMSE, and R^2 Score.
6. **Visualization** – Present insights with bar charts, heatmaps, and maps.

---

5. Tools and Libraries

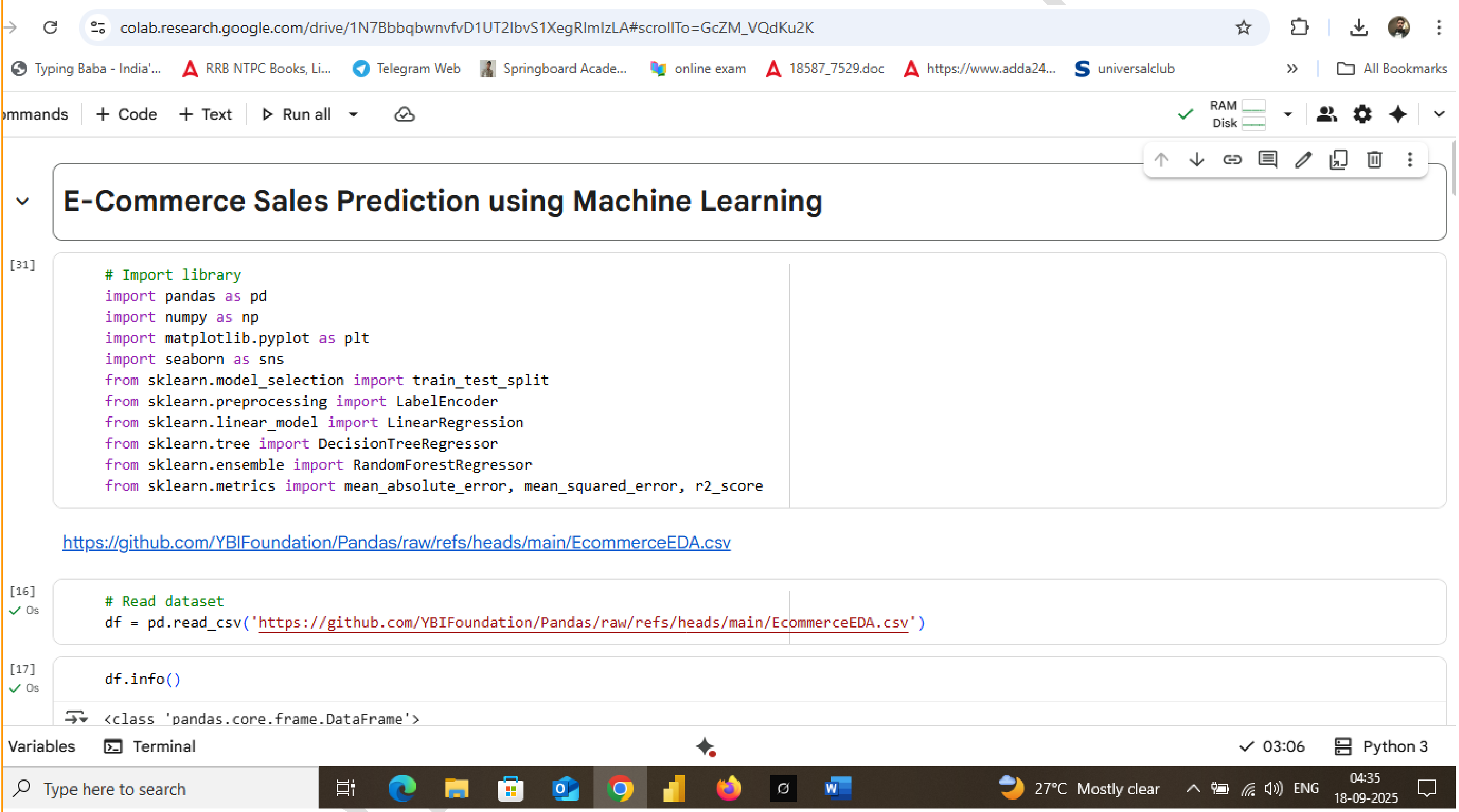
- **Python**
- **Pandas, NumPy** – Data manipulation and analysis
- **Matplotlib, Seaborn** – Visualization
- **Scikit-learn** – Machine learning algorithms

---

6. Expected Outcome

- A clear understanding of e-commerce sales patterns and customer behavior.
- Identification of the most important features influencing sales.
- A machine learning model capable of predicting sales accurately.
- Visual insights through bar charts, heatmaps, and geographical maps.
- Actionable recommendations for business growth and decision-making.

Code and output –



E-Commerce Sales Prediction using Machine Learning

[31]

```
# Import library
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from sklearn.linear_model import LinearRegression
from sklearn.tree import DecisionTreeRegressor
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score
```

https://github.com/YBIFoundation/Pandas/raw/refs/heads/main/EcommerceEDA.csv

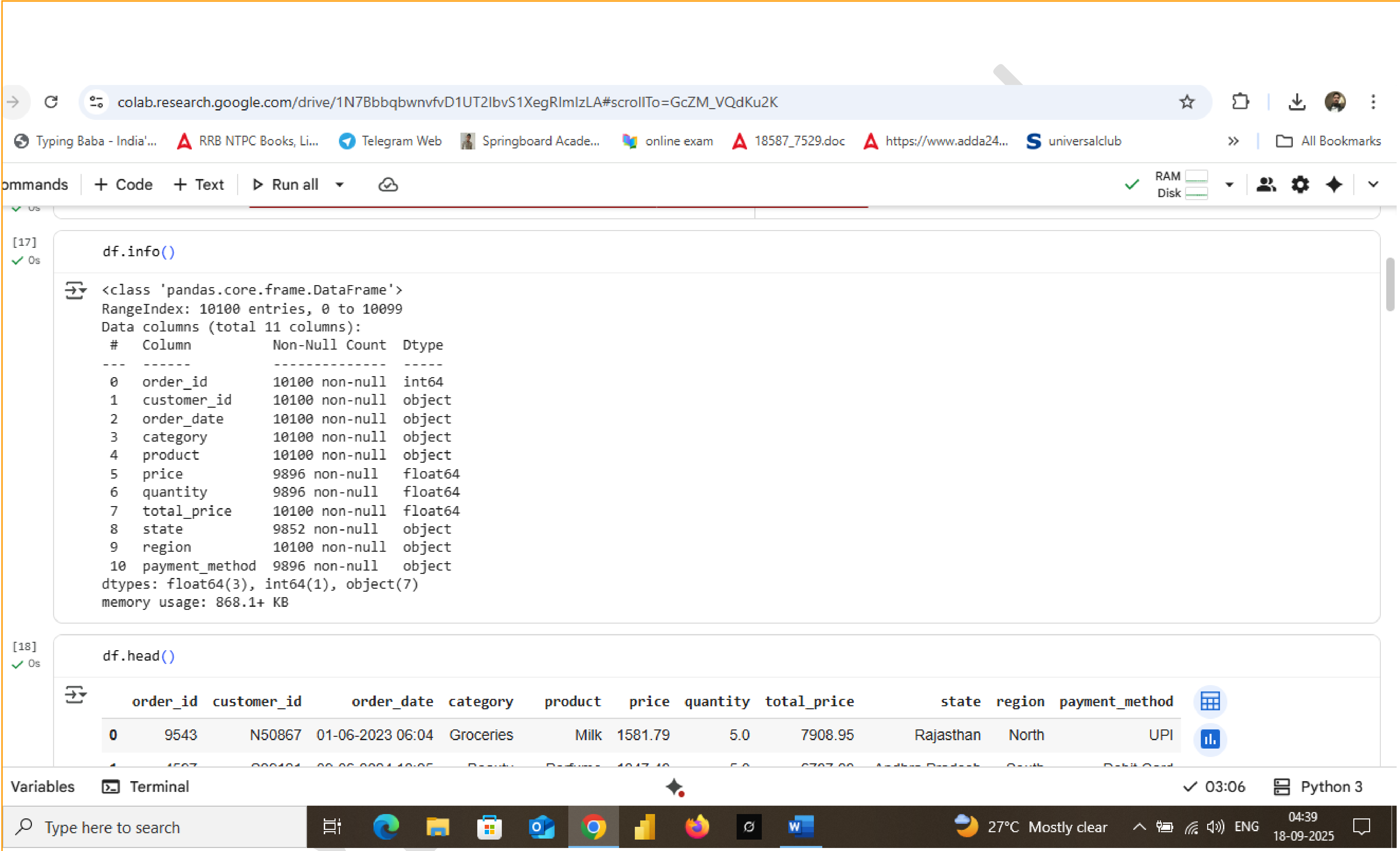
[16] ✓ 0s

```
# Read dataset
df = pd.read_csv('https://github.com/YBIFoundation/Pandas/raw/refs/heads/main/EcommerceEDA.csv')
```

[17] ✓ 0s

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
```

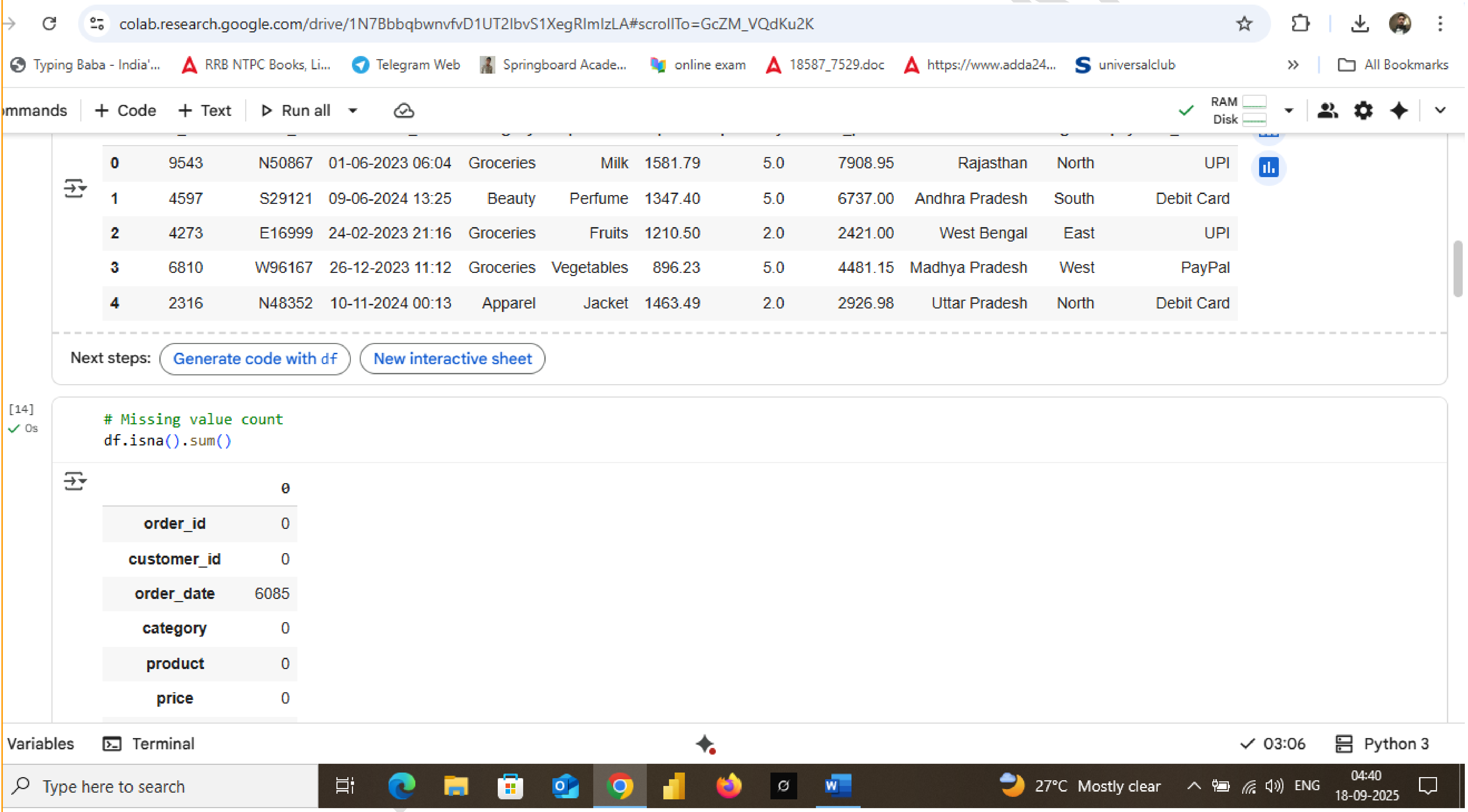


```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10100 entries, 0 to 10099
Data columns (total 11 columns):
 #   Column          Non-Null Count  Dtype  
---  ------          --------------  -----  
 0   order_id        10100 non-null  int64  
 1   customer_id     10100 non-null  object 
 2   order_date      10100 non-null  object 
 3   category        10100 non-null  object 
 4   product         10100 non-null  object 
 5   price           9896 non-null   float64
 6   quantity        9896 non-null   float64
 7   total_price     10100 non-null  float64
 8   state           9852 non-null   object 
 9   region          10100 non-null  object 
 10  payment_method  9896 non-null   object 
dtypes: float64(3), int64(1), object(7)
memory usage: 868.1+ KB
```

```
df.head()
```

	order_id	customer_id	order_date	category	product	price	quantity	total_price	state	region	payment_method
0	9543	N50867	01-06-2023 06:04	Groceries	Milk	1581.79	5.0	7908.95	Rajasthan	North	UPI
1	[illegible]	[illegible]	[illegible]	Beauty	Perfume	[illegible]	5.0	[illegible]	Andhra Pradesh	South	Debit Card



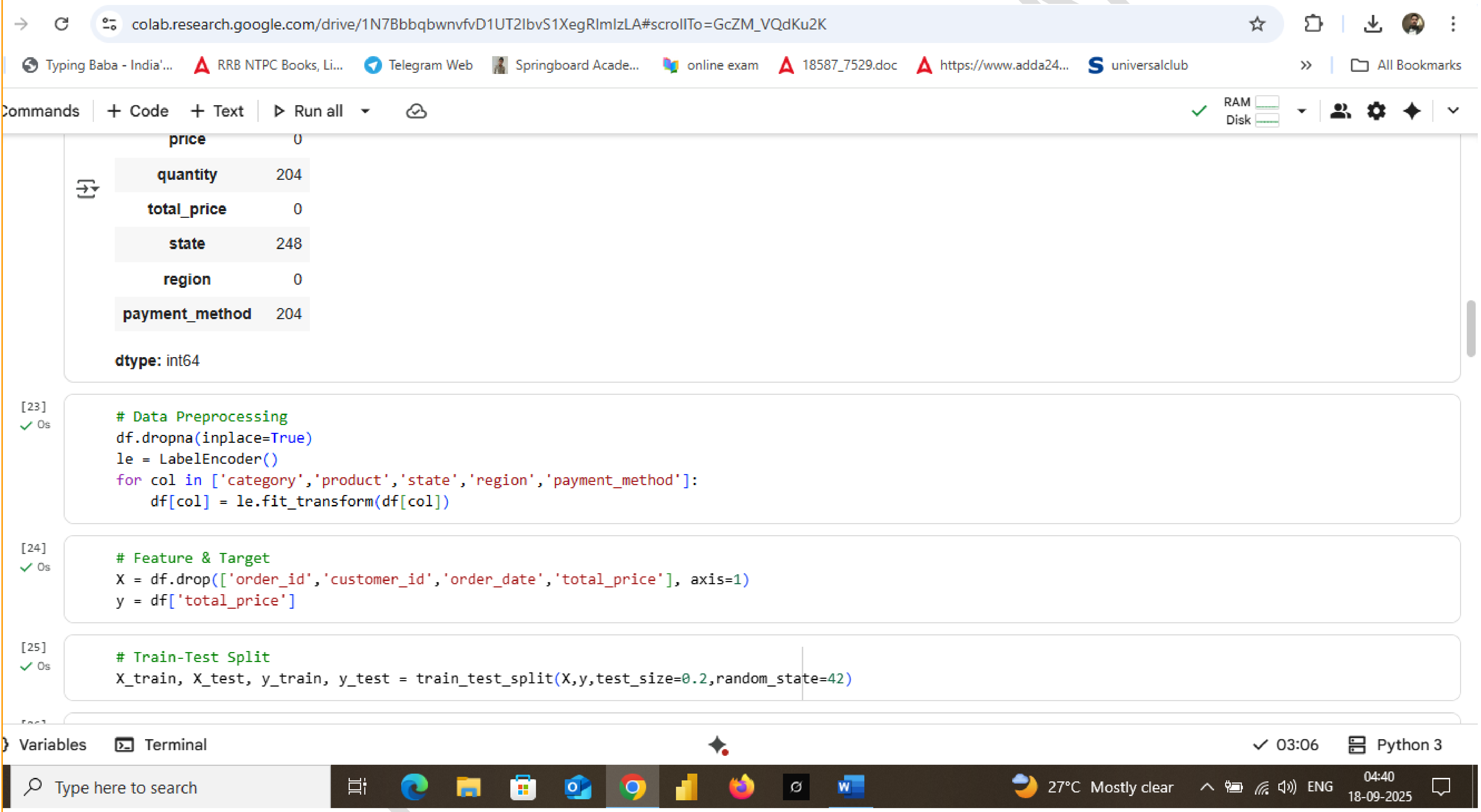
0	9543	N50867	01-06-2023 06:04	Groceries	Milk	1581.79	5.0	7908.95	Rajasthan	North	UPI
1	4597	S29121	09-06-2024 13:25	Beauty	Perfume	1347.40	5.0	6737.00	Andhra Pradesh	South	Debit Card
2	4273	E16999	24-02-2023 21:16	Groceries	Fruits	1210.50	2.0	2421.00	West Bengal	East	UPI
3	6810	W96167	26-12-2023 11:12	Groceries	Vegetables	896.23	5.0	4481.15	Madhya Pradesh	West	PayPal
4	2316	N48352	10-11-2024 00:13	Apparel	Jacket	1463.49	2.0	2926.98	Uttar Pradesh	North	Debit Card

Next steps: Generate code with df New interactive sheet

[14] ✓ 0s

```
# Missing value count
df.isna().sum()
```

	0
order_id	0
customer_id	0
order_date	6085
category	0
product	0
price	0



```
         price    0
      quantity  204
   total_price    0
         state  248
        region    0
payment_method  204

dtype: int64
```

[23] ✓ 0s

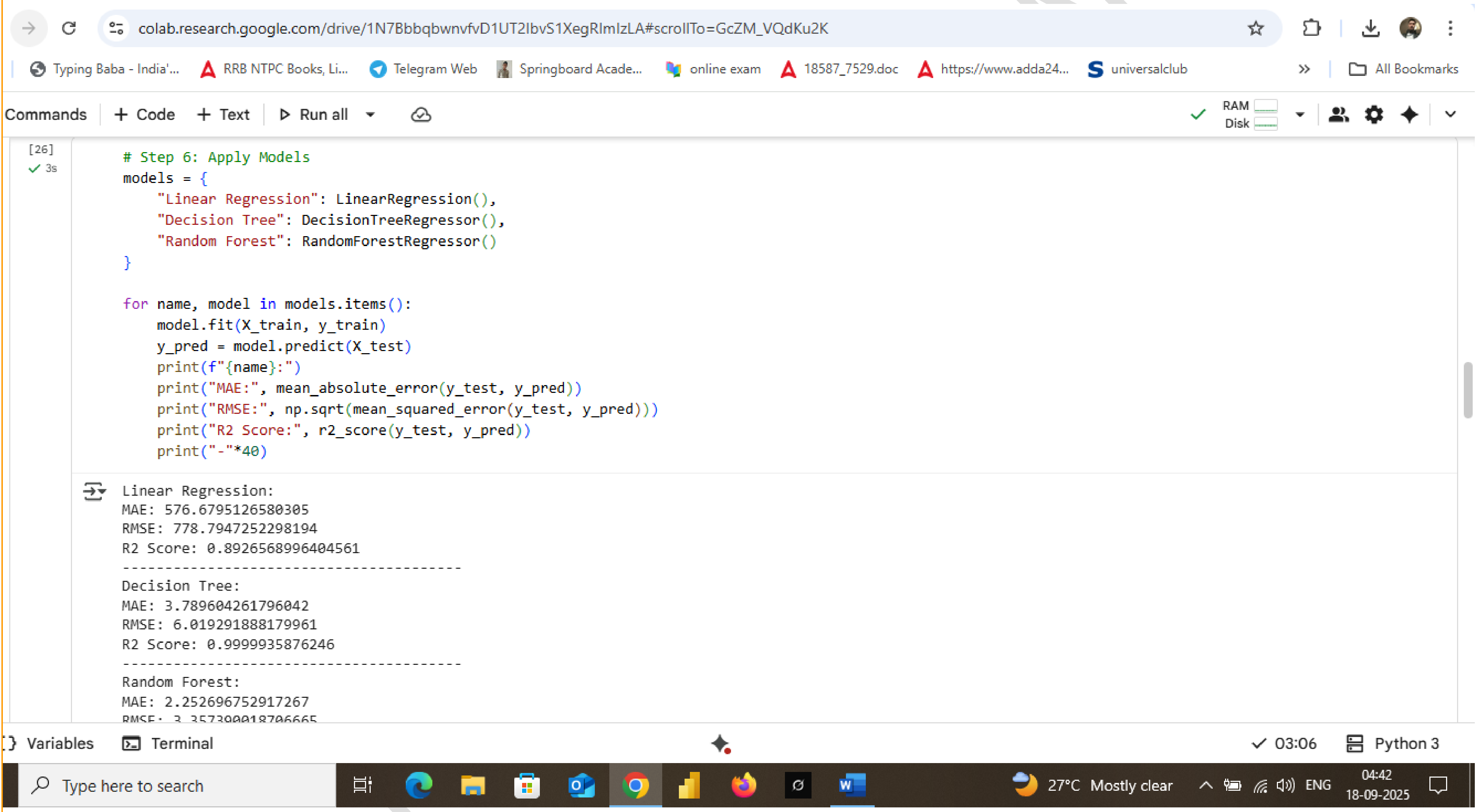
```
# Data Preprocessing
df.dropna(inplace=True)
le = LabelEncoder()
for col in ['category','product','state','region','payment_method']:
    df[col] = le.fit_transform(df[col])
```

[24] ✓ 0s

```
# Feature & Target
X = df.drop(['order_id','customer_id','order_date','total_price'], axis=1)
y = df['total_price']
```

[25] ✓ 0s

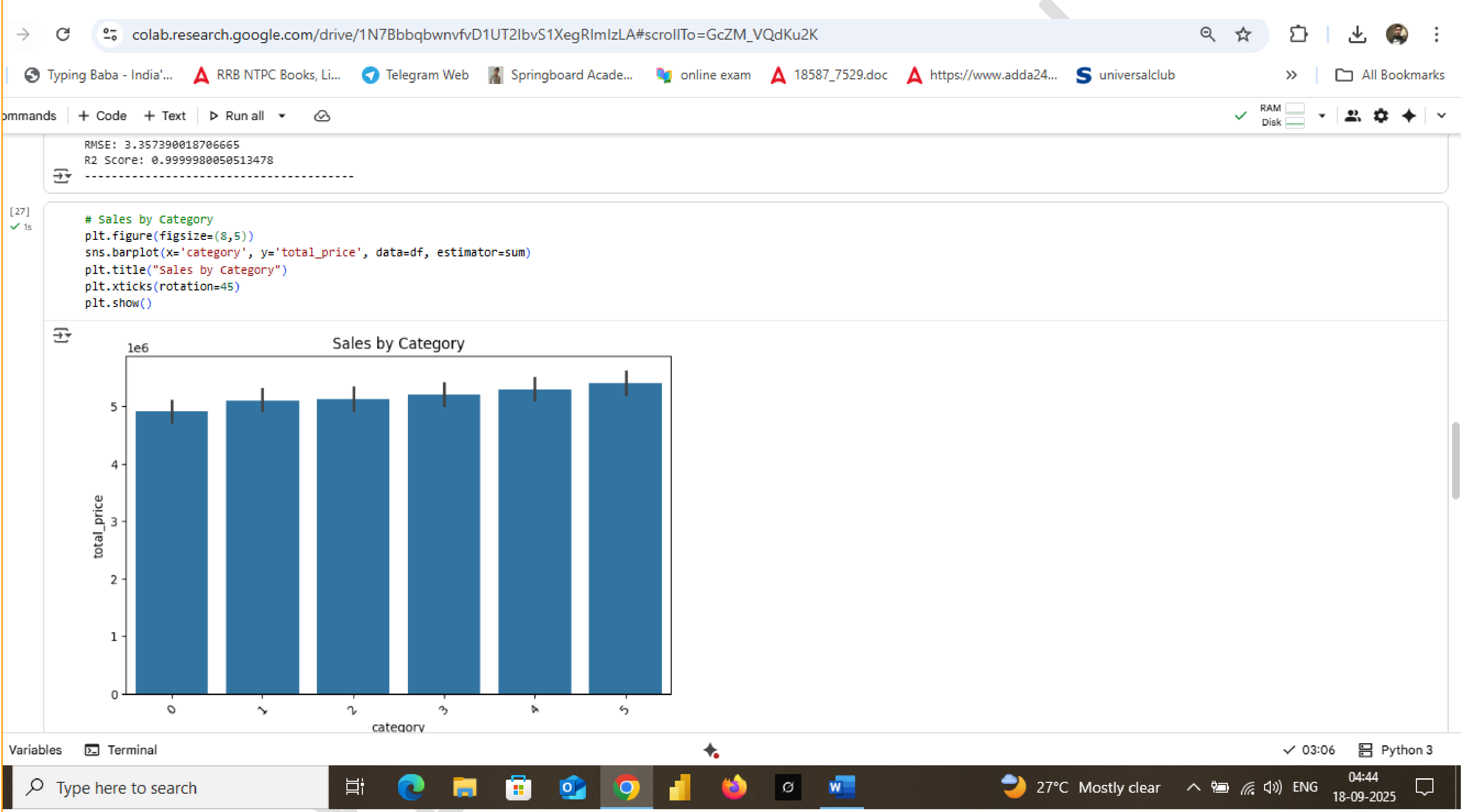
```
# Train-Test Split
X_train, X_test, y_train, y_test = train_test_split(X,y,test_size=0.2,random_state=42)
```



```python
# Step 6: Apply Models
models = {
    "Linear Regression": LinearRegression(),
    "Decision Tree": DecisionTreeRegressor(),
    "Random Forest": RandomForestRegressor()
}

for name, model in models.items():
    model.fit(X_train, y_train)
    y_pred = model.predict(X_test)
    print(f"{name}:")
    print("MAE:", mean_absolute_error(y_test, y_pred))
    print("RMSE:", np.sqrt(mean_squared_error(y_test, y_pred)))
    print("R2 Score:", r2_score(y_test, y_pred))
    print("-"*40)
```

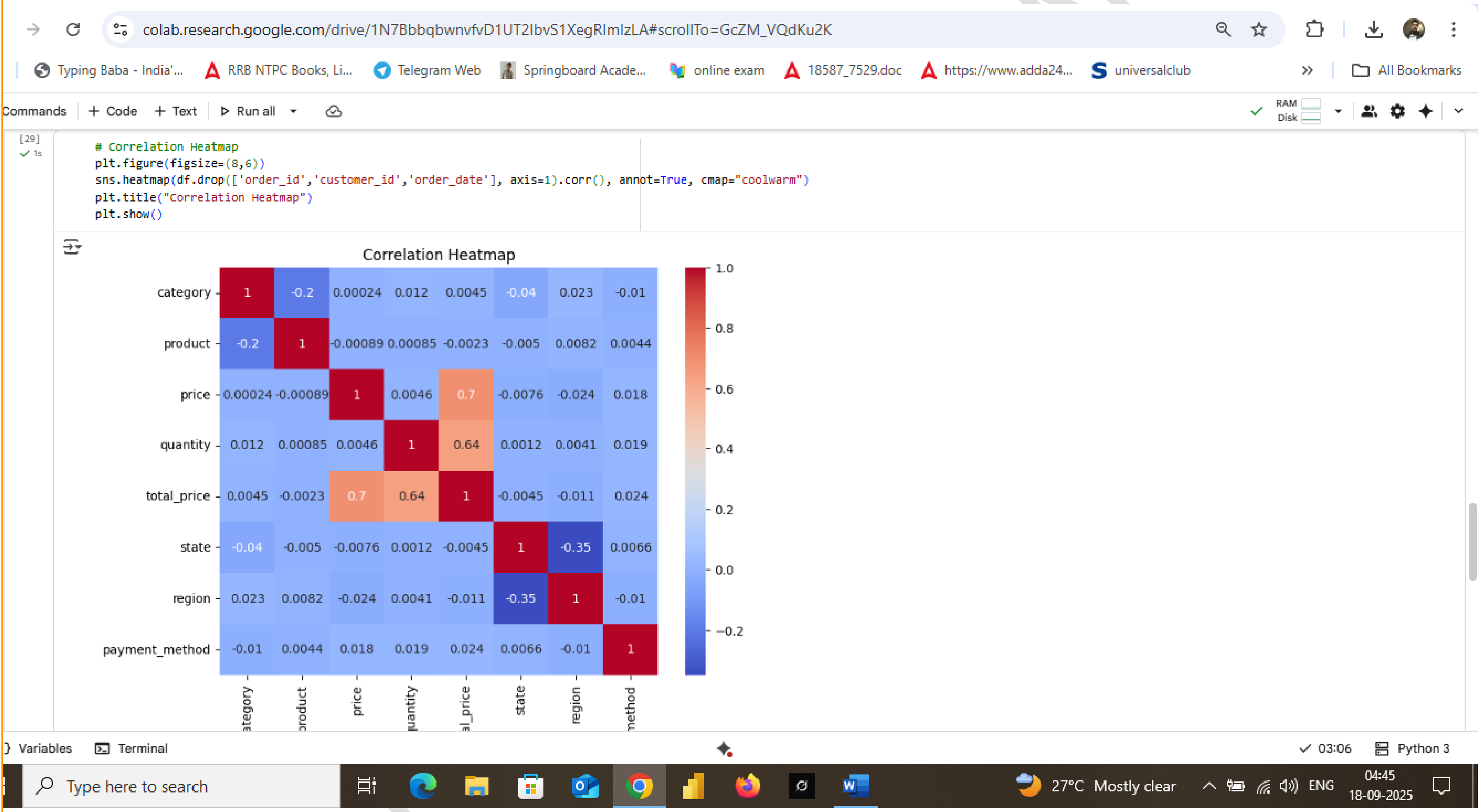
```
Linear Regression:
MAE: 576.6795126580305
RMSE: 778.7947252298194
R2 Score: 0.8926568996404561
----------------------------------------
Decision Tree:
MAE: 3.789604261796042
RMSE: 6.019291888179961
R2 Score: 0.9999935876246
----------------------------------------
Random Forest:
MAE: 2.252696752917267
RMSE: 3.357390018706665
```



```
RMSE: 3.357390018706665
R2 Score: 0.9999980050513478
----------------------------------------
```

[27]

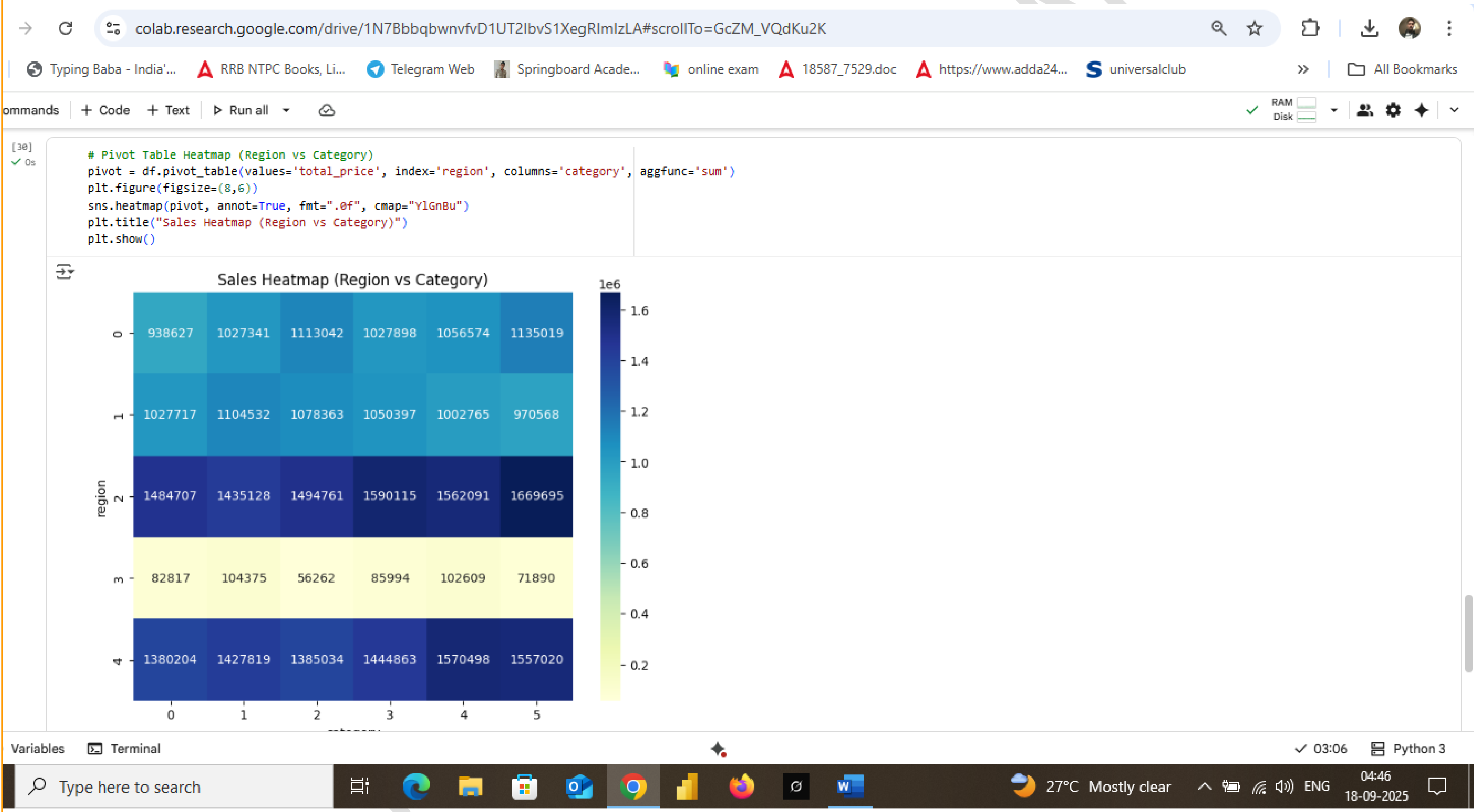
```
# Sales by Category
plt.figure(figsize=(8,5))
sns.barplot(x='category', y='total_price', data=df, estimator=sum)
plt.title("Sales by Category")
plt.xticks(rotation=45)
plt.show()
```



```
# Correlation Heatmap
plt.figure(figsize=(8,6))
sns.heatmap(df.drop(['order_id','customer_id','order_date'], axis=1).corr(), annot=True, cmap="coolwarm")
plt.title("Correlation Heatmap")
plt.show()
```

Correlation Heatmap

	ategory	roduct	price	uantity	al_price	state	region	method
category	1	-0.2	0.00024	0.012	0.0045	-0.04	0.023	-0.01
product	-0.2	1	-0.00089	0.00085	-0.0023	-0.005	0.0082	0.0044
price	0.00024	-0.00089	1	0.0046	0.7	-0.0076	-0.024	0.018
quantity	0.012	0.00085	0.0046	1	0.64	0.0012	0.0041	0.019
total_price	0.0045	-0.0023	0.7	0.64	1	-0.0045	-0.011	0.024
state	-0.04	-0.005	-0.0076	0.0012	-0.0045	1	-0.35	0.0066
region	0.023	0.0082	-0.024	0.0041	-0.011	-0.35	1	-0.01
payment_method	-0.01	0.0044	0.018	0.019	0.024	0.0066	-0.01	1



```
# Pivot Table Heatmap (Region vs Category)
pivot = df.pivot_table(values='total_price', index='region', columns='category', aggfunc='sum')
plt.figure(figsize=(8,6))
sns.heatmap(pivot, annot=True, fmt=".0f", cmap="YlGnBu")
plt.title("Sales Heatmap (Region vs Category)")
plt.show()
```

Sales Heatmap (Region vs Category)

region	0	1	2	3	4	5
0	938627	1027341	1113042	1027898	1056574	1135019
1	1027717	1104532	1078363	1050397	1002765	970568
2	1484707	1435128	1494761	1590115	1562091	1669695
3	82817	104375	56262	85994	102609	71890
4	1380204	1427819	1385034	1444863	1570498	1557020