Data Science Story:

Unleashing the Power of Data

TABLE OF CONTENTS

01 introduction

introduction

- **Welcome!**
- 🚀 Embark on an Exciting Data Science Journey 🚀
- **Key Highlights:**
- Comprehensive Curriculum
- Hands-On Projects
- Real-World Applications
- Expert-Led Learning
- Industry-Recognized Certification
- **Why IBM Professional Data Science?**
- 🔍 Uncover Insights | 📊 Drive Decisions | 🌐 Transform Data

02 Executive Summary

Executive Summary

- **Objective:** Empowering professionals with comprehensive data science skills for real-world applications.
- **Key Components:**
  - **Foundational Knowledge:** Rigorous training in data science fundamentals.
  - **Hands-On Experience:** Practical projects and labs for skill reinforcement.
  - **Advanced Techniques:** Exploration of cutting-edge tools and methodologies.
- **Curriculum Highlights:**
  - **Data Exploration:** Understanding and visualizing data.
  - **Machine Learning:** Building predictive models and algorithms.
  - **Big Data:** Handling large datasets and extracting valuable insights.
- **Benefits:**
  - **Industry-Relevant Skills:** Aligned with current market demands.
  - **Certification:** Recognized credential from IBM.
  - **Networking Opportunities:** Connect with a global community of data professionals.
- **Outcomes:**
  - **Confidence:** Equip yourself with the expertise to tackle real-world data challenges.
  - **Career Advancement:** Unlock new opportunities in data-driven industries.
  - **Impactful Insights:** Translate data into actionable insights for informed decision-making.

03

data collection and data wrangling methodology

Data Collection Methodology

Overview:

- **Sources:**
  - Describe the primary sources of data.
  - Highlight any external databases, APIs, or platforms used.
- **Collection Process:**
  - Outline the step-by-step process for gathering data.
  - Discuss any challenges faced during data acquisition.
- **Data Types:**
  - Specify the types of data collected (structured, unstructured, etc.).
  - Emphasize the relevance of each data type to the project.

Data Wrangling Methodology

Overview:

- **Data Cleaning:**
  - Discuss the techniques employed to handle missing or inconsistent data.
  - Highlight any transformations made to enhance data quality.
- **Feature Engineering:**
  - Explain how new features were created or existing features modified.
  - Showcase the importance of feature selection for model performance.
- **Handling Outliers:**
  - Detail the approach to identify and manage outliers in the dataset.
  - Explain the impact of outlier handling on analysis results.

Data Quality Assurance

Validation Checks:

Describe the validation steps performed on the dataset.
Highlight methods to ensure data integrity.
Data Standardization:

Discuss the process of standardizing data formats.
Illustrate the importance of consistency in data representation.
Documentation:

Emphasize the significance of documenting data transformations.
Showcase any data dictionaries or metadata created.

04

Data Analysis (EDA) and Interactive Visual Analytics Methodology

Exploratory Data Analysis (EDA)

Validation Checks:

- **Purpose of EDA:**
  - Emphasize the role of EDA in understanding data characteristics.
  - Highlight how EDA informs subsequent analysis and model development.
- **Key EDA Techniques:**
  - Showcase statistical summaries, histograms, and descriptive statistics.
  - Discuss the use of visualizations such as scatter plots and correlation matrices.
- **Insights from EDA:**
  - Share specific insights gained during the EDA phase.
  - Demonstrate how initial assumptions were validated or adjusted.

Interactive Visual Analytics

- **Introduction to Visual Analytics:**
  - Define visual analytics and its importance in data exploration.
  - Highlight the value of interactive visualization tools.
- **Tools Used:**
  - List and briefly describe the tools used for interactive visual analytics.
  - Mention any custom visualizations developed for the project.
- **Interactive Features:**
  - Discuss specific interactive features incorporated in visualizations.
  - Emphasize the user-friendly nature for deeper exploration.

Data Storytelling

- **Narrative Building:**
  - Explain the process of building a data-driven narrative.
  - Showcase how insights from EDA contribute to the story.
- **Interactive Dashboards:**
  - Introduce any interactive dashboards created.
  - Illustrate how users can engage with data dynamically.
- **User Engagement:**
  - Discuss the methods used to enhance user engagement with visualizations.
  - Highlight feedback mechanisms and iteration.

05 Predictive Analysis Methodology:

Predictive Analysis Methodology

Objective:

Clearly state the objective of predictive analysis in the context of the project. Emphasize the goal of forecasting or classification.

Data Preparation:

Briefly discuss the steps taken to prepare the data for predictive modeling. Address any missing values, outliers, or feature engineering.

Model Selection

Choice of Models:

Introduce the types of models considered for predictive analysis.
Discuss the rationale behind selecting specific models.
Algorithms Used:

List the algorithms employed for prediction.
- Highlight any ensemble methods or specialized models.

Model Training

Training Process:

Explain the methodology used for training the selected models.
Mention any cross-validation techniques applied.
Hyperparameter Tuning:

Discuss the approach to fine-tuning model hyperparameters.
Emphasize the significance of optimization.

Model Evaluation

- **Performance Metrics:**
  - Introduce the metrics used to evaluate predictive model performance.
  - Include metrics relevant to the project's objectives.
- **Validation Set Results:**
  - Present results on a validation set.
  - Showcase how well the models generalize to new data.

Model Comparison

- **Comparative Analysis:**
  - Provide a comparative analysis of different models.
  - Discuss the strengths and weaknesses of each.
- **Visualization of Results:**
  - Incorporate visualizations to represent model comparison.
  - Use charts or graphs for clarity.

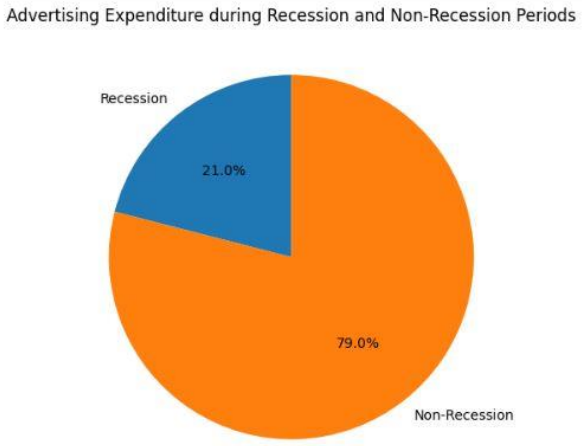
Deployment Strategy

- **Deployment Approach:**
  - Outline the strategy for deploying the predictive model.
  - Discuss any considerations for real-world implementation.
- **Integration with Systems:**
  - Highlight how the model integrates with existing systems.
  - Discuss any API or integration challenges.

06 EDA with visualization

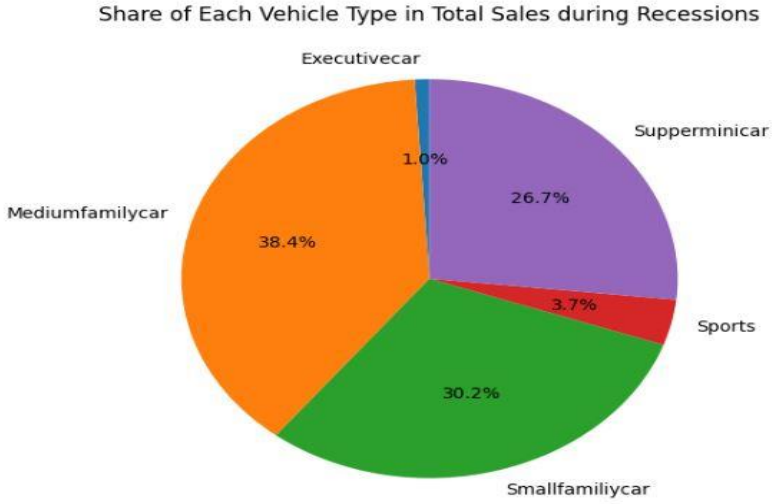
Pie1

[32]:
```
    # Filter the data
Rdata = df[df['Recession'] == 1]
NRdata = df[df['Recession'] == 0]

    # Calculate the total advertising expenditure for both periods
RAtotal = Rdata['Advertising_Expenditure'].sum()
NRAtotal = NRdata['Advertising_Expenditure'].sum()

    # Create a pie chart for the advertising expenditure
plt.figure(figsize=(8, 6))

labels = ['Recession', 'Non-Recession']
sizes = [RAtotal, NRAtotal]
plt.pie(sizes, labels=labels, autopct='%1.1f%%', startangle=90)

plt.title('Advertising Expenditure during Recession and Non-Recession Periods')

plt.show()
```



pie2

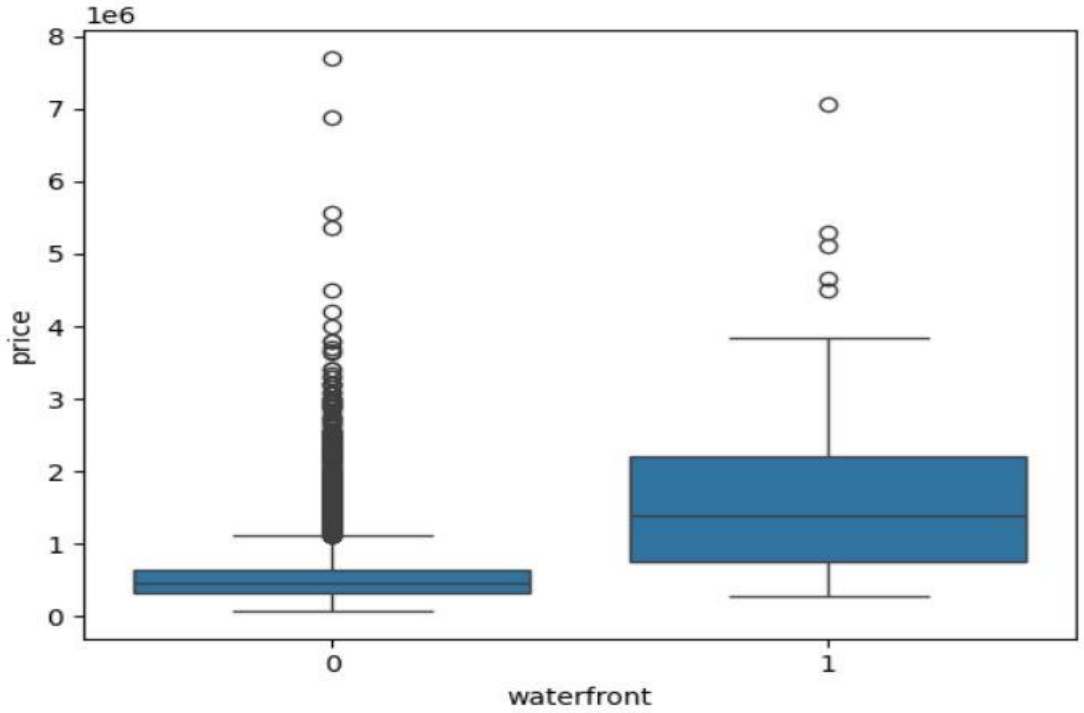
[33]:
```
    # Filter the data
Rdata = df[df['Recession'] == 1]

    # Calculate the sales volume by vehicle type during recessions
VTsales = Rdata.groupby('Vehicle_Type')['Advertising_Expenditure'].sum()

    # Create a pie chart for the share of each vehicle type in total sales during recessions
plt.figure(figsize=(8, 6))

labels = VTsales.index
sizes = VTsales.values
plt.pie(sizes, labels=labels, autopct='%1.1f%%', startangle=90)

plt.title('Share of Each Vehicle Type in Total Sales during Recessions')

plt.show()
```



Box plot 1

```
[46]: sns.boxplot(x=df["waterfront"],y=df["price"])
```

```
[46]: <AxesSubplot:xlabel='waterfront', ylabel='price'>
```



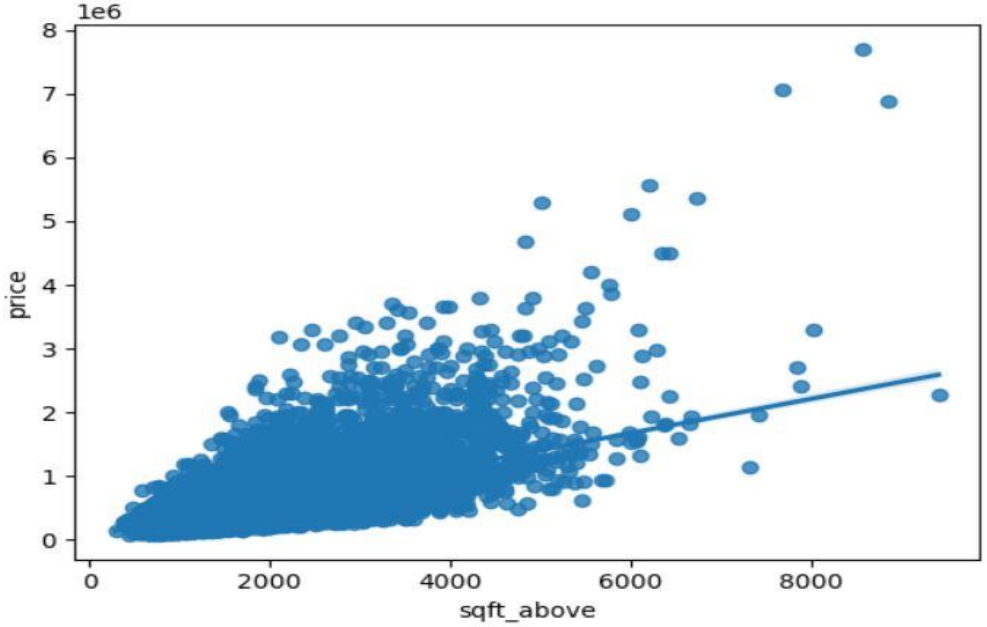
Scatter 1

Question 5 ¶

Use the function `regplot` in the seaborn library to determine if the feature `sqft_above` is negatively or positively correlated with price.

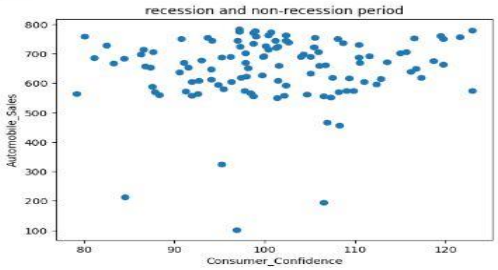
47]:
```
sns.regplot(x=df["sqft_above"],y=df["price"])
```

47]:
```
<AxesSubplot:xlabel='sqft_above', ylabel='price'>
```



Scatter 2

```
[24]: #Create dataframes for recession and non-recession period
rec_data = df[df['Recession'] == 1]
plt.scatter(recession_data['Consumer_Confidence'], rec_data['Automobile_Sales'])

plt.xlabel('Consumer_Confidence')
plt.ylabel('Automobile_Sales')
plt.title('recession and non-recession period')
plt.show()
```

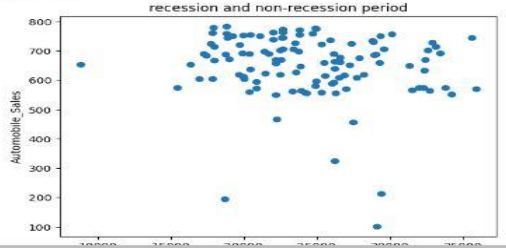


► Click here for Solution template

How does the average vehicle price relate to the sales volume during recessions?

Plot another scatter plot and title it as 'Relationship between Average Vehicle Price and Sales during Recessions'

```
[25]: #Create dataframes for recession and non-recession period
rec_data = df[df['Recession'] == 1]
plt.scatter(recession_data['Price'], rec_data['Automobile_Sales'])

plt.xlabel('Consumer_Confidence')
plt.ylabel('Automobile_Sales')
plt.title('recession and non-recession period')
plt.show()
```



07 EDA with SQL

```python
import mysql.connector
import pandas as pd

# Replace these values with your MySQL server details
db_config = {
    "host": "localhost",
    "user": "root",
    "password": "mena Fci 4321",
    "database": "coursea_data",
}

# Establish a connection to the MySQL server
connection = mysql.connector.connect(**db_config)

# Create a cursor object to interact with the database
cursor = connection.cursor()

# Execute a SELECT query with a subquery to find the Community Area Name with the most number of crimes
cursor.execute("""SELECT ps.NAME_OF_SCHOOL AS School_Name, cd.COMMUNITY_AREA_NAME AS Community_Name, \
ps.AVERAGE_STUDENT_ATTENDANCE
FROM publicschools ps
JOIN censusdata cd ON ps.COMMUNITY_AREA_NUMBER = cd.COMMUNITY_AREA_NUMBER
WHERE cd.HARDSHIP_INDEX = 98;
""")

# Fetch the result
result = cursor.fetchone()

# Display the result
print("Community Area Name with the most number of crimes:", result[0] if result else "No data")

# Close the cursor and connection
cursor.close()
connection.close()
```

```
Community Area Name with the most number of crimes: George Washington Carver Military Academy High School
```

Question 1

Display the data types of each column using the function dtypes, then take a screenshot and subm

[11]:
```
df.dtypes
```

[11]:
```
Unnamed: 0         int64
id                 int64
date              object
price            float64
bedrooms         float64
bathrooms        float64
sqft_living        int64
sqft_lot           int64
floors           float64
waterfront         int64
view               int64
condition          int64
grade              int64
sqft_above         int64
sqft_basement      int64
yr_built           int64
yr_renovated       int64
zipcode            int64
lat              float64
long             float64
sqft_living15      int64
sqft_lot15         int64
dtype: object
```

```python
import mysql.connector
import pandas as pd

# Replace these values with your MySQL server details
db_config = {
    "host": "localhost",
    "user": "root",
    "password": "mena Fci 4321",
    "database": "coursea_data",
}

# Establish a connection to the MySQL server
connection = mysql.connector.connect(**db_config)

# Create a cursor object to interact with the database
cursor = connection.cursor()

# Execute a SELECT query with a subquery to find the Community Area Name with the most number of crimes
cursor.execute("""SELECT c.CASE_NUMBER, c.PRIMARY_TYPE, p.COMMUNITY_AREA_NAME
FROM crimedata c
JOIN publicschools p ON c.COMMUNITY_AREA_NUMBER = p.COMMUNITY_AREA_NUMBER;
""")

# Fetch the result
result = cursor.fetchone()

# Display the result
print("Community Area Name with the most number of crimes:", result[0] if result else "No data")

# Close the cursor and connection
cursor.close()
connection.close()
```

```
Community Area Name with the most number of crimes: JA261898
```

Question 2

Drop the columns `"id"` and `"Unnamed: 0"` from axis 1 using the method `drop()`, then use the method `describe()` to obtain a statistical summary of the data. Take a screenshot and submit it, make sure the `inplace` parameter is set to `True`

[32]:
```
df.drop(["id", "Unnamed: 0"], axis=1, inplace=True)
df.describe()
```

[32]:

	price	bedrooms	bathrooms	sqft_living	sqft_lot	floors	waterfront	view	condition	grade	sqft_above	s
count	2.161300e+04	21600.000000	21603.000000	21613.000000	2.161300e+04	21613.000000	21613.000000	21613.000000	21613.000000	21613.000000	21613.000000	
mean	5.400881e+05	3.372870	2.115736	2079.899736	1.510697e+04	1.494309	0.007542	0.234303	3.409430	7.656873	1788.390691	
std	3.671272e+05	0.926657	0.768996	918.440897	4.142051e+04	0.539989	0.086517	0.766318	0.650743	1.175459	828.090978	
min	7.500000e+04	1.000000	0.500000	290.000000	5.200000e+02	1.000000	0.000000	0.000000	1.000000	1.000000	290.000000	
25%	3.219500e+05	3.000000	1.750000	1427.000000	5.040000e+03	1.000000	0.000000	0.000000	3.000000	7.000000	1190.000000	
50%	4.500000e+05	3.000000	2.250000	1910.000000	7.618000e+03	1.500000	0.000000	0.000000	3.000000	7.000000	1560.000000	
75%	6.450000e+05	4.000000	2.500000	2550.000000	1.068800e+04	2.000000	0.000000	0.000000	4.000000	8.000000	2210.000000	
max	7.700000e+06	33.000000	8.000000	13540.000000	1.651359e+06	3.500000	1.000000	4.000000	5.000000	13.000000	9410.000000	

We can see we have missing values for the columns `bedrooms` and `bathrooms`

```python
import mysql.connector
import pandas as pd
# Replace these values with your MySQL server details
db_config = {
    "host": "localhost",
    "user": "root",
    "password": "mena Fci 4321",
    "database": "coursea_data",
}

# Establish a connection to the MySQL server
connection = mysql.connector.connect(**db_config)

# Create a cursor object to interact with the database
cursor = connection.cursor()
create_procedure_query = """
-- Create or replace the stored procedure
CREATE OR REPLACE PROCEDURE UPDATE_LEADERS_SCORE (
    IN in_School_ID INT,
    IN in_Leader_Score INT
)
BEGIN
    -- Your SQL statements for the stored procedure go here

    -- Example: Update the leaders' score for the specified school
    UPDATE your_table
    SET Leader_Score = in_Leader_Score
    WHERE School_ID = in_School_ID;

    -- End of SQL statements
END;
"""
# Execute a SELECT query with a subquery to find the Community Area Name with the most number of crimes
cursor.execute(create_procedure_query)

# Fetch the result
result = cursor.fetchone()

# Display the result
print("Community Area Name with the most number of crimes:", result[0] if result else "No data")

# Close the cursor and connection
cursor.close()
connection.close()
```

08 Conclusion

Conclusion

- **Insights Gained:**
  - Summarize the major insights and discoveries made during the entire data analysis process.
- **Impact on Decision-Making:**
  - Discuss how the findings from the analysis have influenced decision-making or project direction.
- **Successes and Challenges:**
  - Reflect on the successes achieved in the analysis.
  - Acknowledge any challenges encountered and how they were addressed.

Project Significance:

- **Relevance to Objectives:**
  - Highlight how the analysis aligns with the initial project objectives.
  - Emphasize the relevance of the results to the broader goals.
- **Value Added:**
  - Discuss the value added to the project through the data analysis process.
  - Mention any unexpected or particularly insightful outcomes.

Cont..

Future Directions:

- **Areas for Further Exploration:**
  - Identify any areas within the dataset that warrant further exploration.
  - Suggest potential avenues for future analysis.
- **Improvements and Iterations:**
  - Discuss ways the analysis or methodology could be improved in future iterations.
  - Consider feedback and lessons learned for continuous improvement.

Closing Remarks:

- **Acknowledgments:**
  - Thank any collaborators, team members, or stakeholders who contributed to the analysis.
- **Gratitude:**
  - Express gratitude for the opportunity to conduct the analysis and present the findings.

Thank You!