Test_2_Resuelto

May 13, 2025

Creado por:

Isabel Maniega

1 Test 2

1.1 Question 1

```
In the context of data analysis with Python, what is the most effective approach for maintaini

a) Write detailed documentation for every change made.
b) Regularly commit changes to a centralized database.
c) Email update script versions to team members.
d) Utilize a version control system like Git for your scripts.
```

1.2 Solution 1

d

Using a version control system (VCS) like Git allows you to track changes, revert to previous versions, and collaborate efficiently, which is crucial for maintaining consistency and history in data processing scripts.

1.3 Question 2

```
In a data analysis project, you are aggregating data from various external web sources using Py

a) Limit data collection to a few sources for consistency.
b) Verify data authenticity and integrity as you ingest it.
c) Use threading to seep up data collection.
d) Collect data in small batches and validate each batch.
```

1.4 Solution 2

b

It ensures that any discrepancies or errors are identified and handled as the data is being collected.

1.5 Question 3

```
When dealing with a numerical feature in a dataset, which technique is most appropriate for va

a) Hashing
b) String Matching
c) Boundary Value Testing
d) Regular Expression matching
```

1.6 Solution 3

c

Boundary Value Testing is the most appropriate technique for validating numerical features in a dataset to ensure that the values fall within a specific range of allowable values. This technique involves testing the minimum, maximum, and boundary values of the range to check for any potential issues or errors.

1.7 Question 4

```
When designing an interactive dashboard, what should be the primary focus to enhance user expe

a) Adding a many widgets as possible
b) Using a wide range of contrasting colors.
c) Using complex and intricate visualizations.
d) Focusing on quick load times and responsive design.
```

1.8 Solution 4

d

Focusing on quick load times and responsive design is crucial for enhancing user experience in an interactive dashboard. Users expect the dashboard to load quickly and be responsive to their interactions, providing a smooth and seamless experience.

1.9 Question 5

Which of the following Python code snippets adheres to PEP 8 guidelines for naming classes, methods, and variables?

a)

```
class DataAnalyser:
    def computeAverage(self,data_list):
        total_sum = sum(data_list)
        count = len(data_list)
        return total_sum / count
```

b)

```
class DataAnalyser:
    def compute_average(self,data_list):
        total_sum = sum(data_list)
        count = len(data_list)
        return total_sum / count
```

c)

```
class data_analyser:
    def Compute_Average(self,data_list):
        total_sum = sum(data_list)
        count = len(data_list)
        return total_sum / count
```

d)

```
class Data_analyser:
    def compute_average(self,data_list):
        totalSum = sum(data_list)
        count = len(data_list)
        return totalSum / count
```

1.10 Solution 5

b

This version correctly uses CamelCase for the class name, snake_case for the method name, and snake_case for variable names, adhering to PEP 8 guidelines.

1.11 Question 6

```
Your organization has a complex dataset that combines sales, customer, and inventory data. You

a) Extract KPIs (Key Performance Indicators) relevant to each domain (sales, customer, invento
b) Perform a linear regression analysis to identify the most important variables, and present 
c) Create a detailed dashboard with multiple graphs and tables covering all variables.
d) Apply a neural network model to automatically summarize the data.
```

1.12 Solution 6

a

Extracting KPIs relevant to each domain (sales, customer, inventory) and summarizing them in a single report would be the most effective approach for synthesizing the information. This method allows for a focused analysis on key metrics that are crucial for decision-making in each area, providing a comprehensive overview of the dataset without overwhelming the audience with unnecessary details.

1.13 Question 7

```
Which of the following is the best way to handle missing values in a Pandas DataFrame for time-

a) Use forward fill to populate missing values.
b) Fill missing values with the mean of the column.
c) Replace missing values with zero.
d) Drop all rows that contain missing values.
```

1.14 Solution 7

a

Using forward fill to populate missing values in a Pandas DataFrame for time-series data is a common practice as it helps maintain the continuity of the time series. By filling missing values with the last known value, the overall trend and patterns in the data are preserved, which is crucial for time-series analysis.

1.15 Question 8

```
You are conducting A/B tests and collecting data via Python scripts. What is the best way to e

a) Rely solely on p-values to interpret result.
b) Collect data only during peak hours to maximize sample size.
c) Ensure the A and B groups are randomly assigned.
d) Use the same metric for A and B groups but change the KPI for each test.
```

1.16 Solution 8

c

Ensuring that the A and B groups are randomly assigned is crucial in A/B testing to eliminate bias and ensure that the groups are comparable. Random assignment helps in controlling for variables that could impact the test results, leading to more accurate and reliable conclusions.

1.17 Question 9

You are validating a dataset containing timestamps of customer transactions. Which data validation technique would be most appropriate to ensure the reliability and accuracy of the timestamp data?

a) Completeness validation

b) Format validation.

c) Range validation.

d) Consistency validation

1.18 Solution 9

b

Format validation Format validation checks if the timestamps follow the correct format (e.g., YYYY-MM-DD HH:MM:SS), ensuring the reliability and accuracy of the collected timestamp data.

1.19 Question 10

```
A researcher is collecting data on the daily eating habits of teenagers for a nutritional stud

a) Online surveys completed by the teenagers
b) Direct observation of teenagers during meal times.
c) Analysis of supermarket sales data.
d) Focus group discussions with nutritionists.
```

1.20 Solution 10

a

Online surveys completed by the teenagers Online surveys completed by the teenagers themselves are the most appropriate method for gathering detailed and accurate information on individual dietary patterns. This method allows for direct reporting from the individuals involved, providing specific insights into their daily eating habits.

1.21 Question 11

```
Linda is working with a large dataset of textual customer reviews. She needs to validate that

a) Length validation to ensure reviews meet a minimum word count.
b) Regular expression matching to filter out non-English characters.
c) Natural Language Procesing (NLP) techniques to identify the language used.
d) Format validation to check the text structure.
```

1.22 Solution 11

c

Natural language processing (NLP) techniques to identify the language used. Natural language processing (NLP) techniques are the best choice for Linda to validate that the customer reviews are in English. NLP can accurately identify the language of text entries, distinguishing English from other languages and random characters.

1.23 Question 12

```
Daniel is validating data from multiple sources, including CSV files, JSON files, audio recordi
```

```
a) Ensuring data consistency
b) Performing data format validation
c) Verifying data completeness.
d) Conducting data integrity checks.
```

1.24 Solution 12

b

Performing data format validation Data format validation is crucial for ensuring the reliability and accuracy of data from diverse sources like CSV files, JSON files, audio recordings, and social media posts. It ensures that data follows specific formats or structures, reducing errors and maintaining data quality.

1.25 Question 13

```
You are working on a classification project using a dataset that contains a mix of categorical

a) Ignore categorical variables and focus only on numerical features for modeling.
b) Use label encoding to convert categorcial variables into numerical representations.
c) Drop columns with categorical variables from the dataset to simplify modeling.
d) Utilize one-hot encoding to transform categorical variables into binary columns.
```

1.26 Solution 13

d

Utilize one-hot encoding to transform categorical variables into binary columns. Option D, utilizing one-hot encoding to transform categorical variables into binary columns, is a recommended approach for handling categorical variables in a dataset for modeling. One-hot encoding preserves categorical information without introducing ordinality.

1.27 Question 14

You are required to load data from an Excel file named data.xlsx. The Excel file has multiple sheets, and you are interested in a sheet named 'Employees'. Which of the following code snippets will load this sheet into a Pandas DataFrame?

a)

```
import pandas as pd

df = pd.read_excel('data.xlsx', sheetname='Employees')
```

b)

```
import pandas as pd
```

```
df = pd.read_excel('data.xlsx')['Employees']
```

c)

```
import pandas as pd

df = pd.read_excel('data.xlsx', sheet_name='Employees')
```

d)

```
import pandas as pd

df = pd.read_excel('data.xlsx')
df = df.loc['Employees']
```

1.28 Solution 14

c

This code snippet correctly uses the `pd.read_excel()` function from the Pandas library to read an Excel file named 'data.xlsx' and load the sheet named 'Employees' into a Pandas DataFrame. The `sheet_name` parameter is used to specify the sheet to be loaded.

1.29 Question 15

```
When analyzing a large dataset, a data scientist notices several duplicate records. What is th

a) Keep all duplicates to maintain the size of the dataset.
b) Remove all duplicate records to prevent skewed results.
c) Merge duplicate records by averaging their values.
d) Analyze the duplicates separately to understand their impact.
```

1.30 Solution 15

b

Remove all duplicate records to prevent skewed results. Removing all duplicate records is essential to prevent them from skewing the analysis results. Duplicates can artificially inflate certain statistics or trends, leading to inaccurate conclusions

---

1.31 Question 16

```
When validating a dataset containing URLs of images for a machine learning project, which meth

a) Check the URL format with a regular expression.
b) Dowload each image to verify its existence and integrity.
c) Validate the image dimensions and file size without downloading.
d) Use social media metrics to assess the popularity of images.
```

1.32 Solution 16

b

Download each image to verify its existence and integrity. Downloading each image to verify its existence and integrity is the most reliable method for ensuring that the URLs lead to valid image files. This process confirms not only the URL's correctness but also that the image can be accessed and used for the project.

1.33 Question 17

```
You are analyzing a dataset for predicting housing prices. The dataset includes features like s

a) Convert the "Neighborhood" feature into ordinal categories based on property value.
b) Use one-hot encoding to transform the "Neighborhood" feature into binary columns.
c) Group similar neighborhoods together and create a new categorical feature.
d) Ignore the "Neighborhood" feature as it is not relevant for predicting housing prices.
```

1.34 Solution 17

b

Use one-hot encoding to transform the "Neighborhood" feature into binary columns. Option B is the recommended approach as one-hot encoding preserves the categorical information of the "Neighborhood" feature without introducing ordinality or bias.

1.35 Question 18

```
A data analyst is working with a dataset that includes a 'salary' column with some unusually lo

a) Delete any records with unusually low salaries.
b) Replace the low salary values with the column average.
c) Investigate the low salary entries and correct them if necessary.
d) Convert all salaries to a uniform currency unit without addressing the low values.
```

1.36 Solution 18

c

Investigate the low salary entries and correct them if necessary. Investigating and verifying the accuracy of the unusually low salary values allows the analyst to determine whether they are indeed errors and to make informed decisions on how to correct them, ensuring data quality and reliability.

1.37 Question 19

John has created a line chart showing the trend of website traffic over the past year. How should John effectively communicate the insight gained from this visualization to a technical audience?

a) Use plain language to describe the overall trend in website traffic.

b) Highlight the design and color choices used in the line chart.

c) Explain the data preprocessing steps before creating the chart.

d) Discuss the statistical methods used to analyze the website traffic data.

1.38 Solution 19

d

Discuss the statistical methods used to analyze the website traffic data. When communicating with a technical audience like John, it's important to provide details about the statistical methods used to analyze the data and create the visualization.

1.39 Question 20

```
You are working with a dataset containing high-dimensional features and want to visualize the

a) Principal Component Analysis (PCA)
b) Linear Regression
c) Decision Tree
d) K-Means Clustering
```

1.40 Solution 20

a

Principal Component Analysis (PCA) PCA is a dimensionality reduction technique that transforms high-dimensional data into a lower-dimensional space while preserving as much variance as possible, making it suitable for visualizing high-dimensional data in a more manageable form.

1.41 Question 21

```
You are working on a Python script to parse and extract information from multiple JSON files i

a) Write a single monolithic script that processes all JSON files in the directory.
b) Use modular programming by dividing the script into functions and modules for different tas
c) Hard-code file paths and names in the script for direct file access.
d) Avoid using error handling mechanisms to keep the script concise.
```

1.42 Solution 21

b

Use modular programming by dividing the script into functions and modules for different tasks. Option B, using modular programming by dividing the script into functions and modules, is a recommended best practice for scripting maintainability and scalability. This approach allows for easier maintenance, reusability of code, and scalability for handling new files or tasks.

1.43 Question 22

```
In Python programming, especially in data analysis, what practice should be avoided to maintai

a) Writing extensive documentation for complex functions.
b) Consistently formatting code according to PEP 8 standards.
c) Employing cryptic or non-descriptive variables names.
d) Organizing code into functions or classes based on functionality.
```

1.44 Solution 22

b

Using non-descriptive or cryptic variable names makes the code hard to understand and maintain. It's a practice that should be avoided.

1.45 Question 23

```
In the context of data analysis with Python, what is the most effective approach for maintaini

a) Email updated script versions to team members.
b) Regularly commit changes to a centralized database.
c) Write detailed documentation for every change made.
d) Utilize a version control system like GIT for your scripts.
```

1.46 Solution 23

d

Using a version control system (VCS) like Git allows you to track changes, revert to previous versions, and collaborate efficiently, which is crucial for maintaining consistency and history in data processing scripts.

1.47 Question 24

You're developing a Python program that calculates the square root of a number entered by the user. To ensure the program can handle cases where the user inputs a negative number or a non-numeric value, which of the following code snippets provides robust error handling?

```
a)
import math

try:
    num = float(input("Enter a number: "))
    print(math.sqrt(num))
except TypeError:
    print("Please enter a valid number.")

b)
import math

try:
    num = float(input("Enter a number: "))
    print(math.sqrt(num))
except ValueError:
    print("Please enter a valid number.")
except:
    print("An error occurred.")

c)
import math

try:
    num = int(input("Enter a number: "))
    print(math.sqrt(num))
except ArithmeticError:
    print("Please enter a valid number.")

d)
import math

try:
    num = float(input("Enter a number: "))
    if num < 0:
        raise ValueError("Cannot calculate square root of a negative number.")
    print(math.sqrt(num))
except ValueError:
    print("Please enter a valid positive number.")
```

1.48 Solution 24

b

This code snippet handles both ValueError for non-numeric inputs and negative numbers, and a general exception for any other errors, ensuring the program can handle a wide range of input errors robustly.

1.49 Question 25

A software engineer is tasked with ensuring the integrity of user-submitted data in an online booking form. The form collects user information such as name, contact number, and email address. The engineer writes a Python function to validate this data before storing it in the database. Consider the following code snippet for validation:

```
def check_booking_info(name, contact, email):
    if not name.replace(' ', '').isalpha():
        return "Name must contain only letters and spaces."
    if not contact.isdigit() or len(contact) != 10:
        return "Contact number must be a 10-digit number."
    if "@" not in email or "." not in email:
        return "Email must contain an '@' and a '.' character."
    return "All inputs are valid."

result = check_booking_info("John Doe", "1234567890", "john@example.com")
```

What is the primary purpose of the check_booking_info function as shown in the code snippet?

a) To ensure data integrity by applying specific validation rules to user inputs, ensuring correctness database entry.

b) To reject user inputs that do not follow predefined length restrictions, without considering the actual content or format.

c) To ensure data integrity by applying specific validation rules to user inputs, ensuring correctness database entry.

d) To modify user inputs by removing numeric characters from names, validating contact numbers, and verifying the presence of specal characters in email addresses.

1.50 Solution 25

c

The `check_booking_info` function is performing a series of checks to validate the user input data:

- **Name validation**: Ensures the name contains only letters and spaces.
- **Contact number validation**: Ensures the contact number is a 10-digit number.
- **Email validation**: Ensures the email contains both an “@” symbol and a “.” character.

The primary purpose of this function is to **validate** the integrity of the input data by enforcing these rules before the data is stored in the database.

1.51 Question 26

You are analyzing a dataset containing information about customer satisfaction scores for a product, along with factors such as price, features, and customer demographics. Your goal is to build a linear regression model to predict customer satisfaction based on these features. Which evaluation metric is most appropriate for assessing the performance of your linear regression model?

a) F1 Score

b) Mean Square Error (MSE)

c) Accuracy

d) Confusion Matrix

1.52 Solution 26

b

Mean Squared Error (MSE) Mean Squared Error (MSE) is a suitable evaluation metric for linear regression models, measuring the average squared difference between actual and predicted values.

1.53 Question 27

Suppose you have the following list of temperatures (in Fahrenheit) recorded over a week:

`temperatures = [72, 75, 78, 72, 70, 72, 76, 79]`

Which code snippet will help you find the median temperature using Python?

a)

```
median_temp = sorted(temperatures)[len(temperatures) // 2 if len(temperatures) % 2 == 1 else \
              (sorted(temperatures)[len(temperatures) // 2 - 1] + sorted(temperatures)[len(tem
```

b)

```
median_temp = temperatures.median()
```

c)

```
median_temp = sorted(temperatures)[len(temperatures) // 2]
```

d)

```
median_temp = max(set(temperatures), key=temperatures.count)
```

1.54 Solution 27

[1]:
```
# a
temperatures = [72, 75, 78, 72, 70, 72, 76, 79]

median_temp = sorted(temperatures)[len(temperatures) // 2]\
              if len(temperatures) % 2 == 1 else \
              (sorted(temperatures)[len(temperatures) // 2 - 1]\
               + sorted(temperatures)[len(temperatures) // 2]) / 2
median_temp
```

[1]:
```
73.5
```

[2]:
```
# b

median_temp = temperatures.median()
median_temp
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
Cell In[2], line 3
      1 # b
----> 3 median_temp = temperatures.median()
      4 median_temp

AttributeError: 'list' object has no attribute 'median'
```

[3]:
```
# c
median_temp = sorted(temperatures)[len(temperatures) // 2]
median_temp
```

[3]:
```
75
```

[4]:
```
# d
median_temp = max(set(temperatures), key=temperatures.count)
median_temp
```

[4]:
```
72
```

a

This option checks whether the length of the list is odd or even. It handles both cases correctly by finding the middle element for odd-length lists and averaging the two middle elements for even-length lists. This is the correct solution.

1.55 Question 28

In a Python application that connects to an SQL database, you are responsible for handling various types of data such as user IDs (e.g., 10234), user names (e.g., John Doe), email addresses (e.g., john.doe@example.com), and account balances (e.g., 1500.75). Which practices should be followed to ensure effective and responsible data handling?

a)

```
query = f"SELECT * FROM users WHERE user_id = {user_id}"

try:
    cursor.execute(query)
except Exception as e:
    print("An error occurred")
```

b)

```
query = "SELECT * FROM users WHERE user_id = %s"

cursor.execute(query, (user_id,))
```

c)

```
query = "SELECT * FROM users WHERE user_id = " + user_id
cursor.execute(query)
```

d)

```
import re

def validate_email(email):
    pattern = r'^[a-zA-Z0-9_.+-]+@[a-zA-Z0-9_]+\.[a-zA-Z0-9-.]+$'
    if not re.match(pattern, email):
        raise ValueError("Invalid email address")

query = "SELECT * FROM users WHERE user_id = %s"
cursor.execute(query, (user_id,))
```

1.56 Solution 28

d

This option follows best practices for data handling in a Python application connected to an SQL database:

- **Validation and Sanitization**: The email address is validated using a regular expression to ensure it is in the correct format.
- **Parameterized Queries**: Using parameterized queries helps prevent SQL injection attacks by separating the query logic from the data.
- **Error Handling**: Properly handling potential errors ensures that issues are identified and managed effectively, maintaining application stability and security.

1.57 Question 29

Consider the following Python program that calculates the total hours worked in a week from a list containing daily work hours:

```
daily_hours = [8, 9, 7, 5, 6, 8, 10]
total_hours = 0
overtime = None

for hours in daily_hours:
    total_hours += hours

if total_hours > 40:
    overtime = True
else:
    overtime = False

print("Total Hours:", total_hours)
print("Overtime Achieved:", overtime)
```

Given the daily work hours `[8, 9, 7, 5, 6, 8, 10]`, what is the expected output of the program?

a)

```
Total Hours: 53
Overime Achieved: False
```

b)

```
Total Hours: 53
Overime Achieved: Invalid
```

c)

```
Total Hours: 53
Overime Achieved: Not Applicable
```

d)

```
Total Hours: 53
Overime Achieved: True
```

1.58 Solution 29

[5]:
```
daily_hours = [8, 9, 7, 5, 6, 8, 10]
total_hours = 0
overtime = None

for hours in daily_hours:
    total_hours += hours

if total_hours > 40:
    overtime = True
else:
    overtime = False

print("Total Hours:", total_hours)
print("Overtime Achieved:", overtime)
```

```
Total Hours: 53
Overtime Achieved: True
```

d

The sum of the daily work hours is 53, which exceeds the 40-hour threshold, resulting in overtime being set to True.

1.59 Question 30

Sarah is analyzing temperature data over a week and wants to create a histogram to show the frequency distribution of temperatures. Which function from the Matplotlib library would Sarah use to plot this histogram?

a) plt.scatter()

b) plt.bar()

c) plt.plot()

d) plt.hist()

1.60 Solution 30

c

The plt.hist() function in Matplotlib is specifically designed for creating histograms, making it the appropriate choice for visualizing frequency distributions like temperature data over a week.

1.61 Question 31

A data scientist needs to validate a dataset containing weather records before using it for a climate study. The dataset includes fields like RecordID, Date, Temperature, and Precipitation. Consider the following code snippet for validation:

```
def validate_weather_data(records):
    for record in records:
        if not isinstance(record['RecordID'], int) or not isinstance(record['Temperature'], (i
            return False
        if record['Temperature'] < -50 or record['Temperature'] > 50:
            return False
        if record['Precipitation'] < 0:
            return False
    return True

records = [
    {'RecordID': 101, 'Date': '2023-01-01', 'Temperature': 25.0, 'Precipitation': 5.0},
    {'RecordID': 102, 'Date': '2023-01-02', 'Temperature': -55.0, 'Precipitation': 2.0},
    # ... more records
]

validation_result = validate_weather_data(records)
```

What does the validation function ensure?

a) Checks that Date is in the correct format, Temperature is standardized to Celsius, and every RecordID is linked to a Precipitation value.

b) Verifies that RecordID is a number, Temperature is within a realistic range, and Precipitation is non-negative.

c) Validates that Date falls, within the current year, RecordID and temperature are sequential, and Precipitation reflects accurate measurements.

d) Ensures RecordID is unique, converts Date into numeric format, and adjusts negative Precipitation values.

1.62 Solution 31

b

The function checks that RecordID is an integer, Temperature is within a plausible range, and Precipitation is non-negative, ensuring data accuracy and integrity

---

1.63 Question 32

Consider the following Python code snippet:

```
fruit = "banana"
numbers1 = (8, 3, 9, 2)
numbers2 = (5, 7, 6, 1)
combined_numbers = numbers1 + numbers2
unique_set = set([ord(fruit[2]), combined_numbers[7]])

print("Third Character:", fruit[2])
print("Eighth Number:", combined_numbers[7])
print("Unique Set:", unique_set)
```

Knowing that the ASCII value for the character ‘n’ is 110, what will be the output of this code?

a)

```
Third Character: n
Eighth Number: 1
Unique Set: {1, 110}
```

b)

```
Third Character: n
Eighth Number: 1
Unique Set: {n, 49}
```

c)

```
Third Character: n
Eighth Number: 1
Unique Set: {n, 1}
```

d)

```
Third Character: n
Eighth Number: 1
Unique Set: {110, 49}
```

1.64 Solution 32

```
[6]: fruit = "banana"
numbers1 = (8, 3, 9, 2)
numbers2 = (5, 7, 6, 1)
combined_numbers = numbers1 + numbers2
unique_set = set([ord(fruit[2]), combined_numbers[7]])

print("Third Character:", fruit[2])
print("Eighth Number:", combined_numbers[7])
print("Unique Set:", unique_set)
```

```
Third Character: n
Eighth Number: 1
Unique Set: {1, 110}
```

a

The ASCII value for ‘n’ is 110, and the eighth number in the combined tuple is 1. Therefore, the unique set will contain 110 and 1. The combined tuple combined_numbers is created by concatenating numbers1 and numbers2, making the eighth element of combined_numbers the last element of numbers2, which is 1.

Recall that in a set notaion {1,110} is the same as {110,1}, i.e. the order in which the set elements are listed is not important

---

1.65 Question 33

```
You have to retrieve data from a database table named employees. The table contains columns em

a) SELECT first_name, salary FROM employees WHERE department NOT 'HR';
b) SELECT first_name, salary FROM employees WHERE department = 'HR';
c) SELECT first_name, last_name, salary FROM employees WHERE department IS 'HR';
d) SELECT first_name, salary FROM employees WHERE department = 'HR' AND salary;
```

1.66 Solution 33

b

This query correctly selects the first_name and salary columns from the employees table where the department is ‘HR’.

1.67 Question 34

You are analyzing a dataset containing numeric data from various sensors. Which data validation technique should you use to ensure the reliability and accuracy of the numeric data?

a) Data type validation

b) Completeness validation.

c) Range validation.

d) Consistency validation

1.68 Solution 34

c

Range validation Range validation checks if the numeric data falls within expected ranges, ensuring the reliability and accuracy of the data collected from sensors

---

1.69 Question 35

You are tasked with consolidating and validating data from three different datasets: a CSV file containing customer information, an Excel sheet with product details, and a JSON file with transaction records. Which approach is most suitable for consolidating and validating this diverse data?

a) Use pandas to read each file and concatenate them into a single Dataframe, then perform data type validation and cross-reference validation.

b) Write custom script to read and merge the data from each file manually, then apply data type validation and range validation.

c) Use thrid-party software to import and integer the data from each file separately, then apply range validation and cross-reference validation.

d) Convert all data to a common format (e.g., CSV), merge them using Excel, and then validate using formulas and conditional formatting

1.70 Solution 35

a

Use pandas to read each file and concatenate them into a single DataFrame, then perform data type validation and cross-reference validation. Using pandas to read and concatenate different file formats into a single DataFrame allows for efficient data consolidation. Data type validation ensures the accuracy of data types, while cross-reference validation checks for consistency and relationships between different datasets.

1.71 Question 36

You have a DataFrame df that consolidates data from multiple sources. Your next task is to validate the "Age" column to ensure all ages are between 0 and 120. Which of the following code snippets will correctly filter out the invalid entries?

a) `valid_df = df[df['Age'].between(0, 120, inclusive=False)]`

b) `valid_df = df[(df['Age'] >= 0) & (df['Age'] <= 120)]`

c) `valid_df = df.query('Age > 0) and Age < 120')`

d) `valid_df = df[(df['Age'] > 0) & (df['Age'] < 120)]`

1.72 Solution 36

b

This code snippet correctly filters out the invalid entries by selecting rows where the "Age" column values are greater than or equal to 0 and less than or equal to 120, ensuring that all ages fall within the specified range.

1.73 Question 37

You need to collect data from an Excel file with multiple sheets. One of the sheets is named "Sales" and another is named "Inventory". You want to load the "Sales" sheet into a DataFrame named df_sales and the "Inventory" sheet into a DataFrame named df_inventory. Which of the following code snippets accomplishes this?

a)

```
df_sales = pd.read_csv('file.xlsx', sheet_name='Sales')
df_inventory = pd.read_csv('file.xlsx', sheet_name='Inventory')
```

b)

```
df_sales = pd.read_excel('file.xlsx').parse('Sales')
df_inventory = pd.read_excel('file.xlsx').parse('Inventory')
```

c)

```
xls = pd.Excel('file.xlsx')
df_sales = xls.parse('Sales')
df_inventory = xls.parse('Inventory')
```

d)

```
df_sales = pd.read_excel('file.xlsx', sheet_name='Sales')
df_inventory = pd.read_excel('file.xlsx', sheet_name='Inventory')
```

1.74 Solution 37

d

This code snippet correctly uses the `pd.read_excel()` function to read an Excel file and load specific sheets into DataFrames. By specifying the `sheet_name` parameter for each DataFrame, it ensures that the "Sales" sheet is loaded into `df_sales` and the "Inventory" sheet is loaded into `df_inventory`.

1.75 Question 38

You have developed a logistic regression model for a classification task. After training, you notice that both the training and validation errors are high. Which of the following might be the problem with your model?

a) The model is underfitting the data.

b) The learning rate is too high.

c) The model is overfitting the data.

d) The model is too complex.

1.76 Solution 38

a

If both the training and validation errors are high, it indicates that the model is not capturing the underlying patterns in the data adequately. This situation is known as underfitting, where the model is too simple to represent the complexity of the data, resulting in poor performance on both training and validation sets

---

1.77 Question 39

You're asked to write a Python script to clean a dataset by dropping columns that have more than 50% missing values. Which of the following code snippets accomplishes this task using the Pandas library?

a) `df = df.loc[:, df.isnull().mean() <= 0.5]`

b) `df = df.drop(df.columns[df.isnull().sum() > (df.shape[0] // 2], axis=1)`

c) `df.dropna(thresh=df.shape[0] * 0.5, axis=1, inplace=True)`

d) `df = df.drop(df.columns[df.isnull().mean() > 0.5, axis=1)`

1.78 Solution 39

d

This code effectively drops the columns where the mean of missing values is greater than 0.5.

1.79 Question 40

While evaluating a logistic regression model using a confusion matrix, you notice that the True Positive rate is significantly higher than the True Negative rate. What does this observation imply?

a) The model performs equally well for both classes.

b) The model is better at identifying the positive class than the negative class.

c) The model is overfitting

d) The model has a high level of bias.

1.80 Solution 40

b

A significantly higher True Positive rate compared to the True Negative rate indicates that the model is better at correctly identifying instances of the positive class (True Positives) than instances of the negative class (True Negatives). This suggests that the model may have a bias towards predicting the positive class.

1.81 Question 41

When validating a dataset with datetime entries, you notice that some timestamps are incorrectly formatted. Which Python library and method are best suited to enforce a uniform datetime format?

a) `calendar.timegm()` from Python's `calendar` library.

b) `pd.to_datetime()` from the Pandas library.

c) `re.sub()` from the `re` library.

d) `datetime.strptime()` from Python's `datetime` library.

1.82 Solution 41

a

The pd.to_datetime() method from the Pandas library is best suited for enforcing a uniform datetime format in a dataset. It can convert a wide variety of date and time formats into a consistent datetime format, making it ideal for standardizing timestamps in a dataset.

1.83 Question 42

You are working on a predictive model to classify emails as either Spam or Not Spam. You have built a model and it has an accuracy of 99% on your training dataset. However, when you test it on a new set of emails, the accuracy drops to 70%. What is most likely happening?

a) Class Imbalance

b) Overfitting

c) Underfitting

d) High Bias

1.84 Solution 42

b

Overfitting occurs when a model performs extremely well on the training data but fails to generalize to new, unseen data. In this case, the high accuracy on the training dataset (99%) suggests that the model may have memorized the training data instead of learning the underlying patterns, leading to a drop in accuracy when tested on new emails.

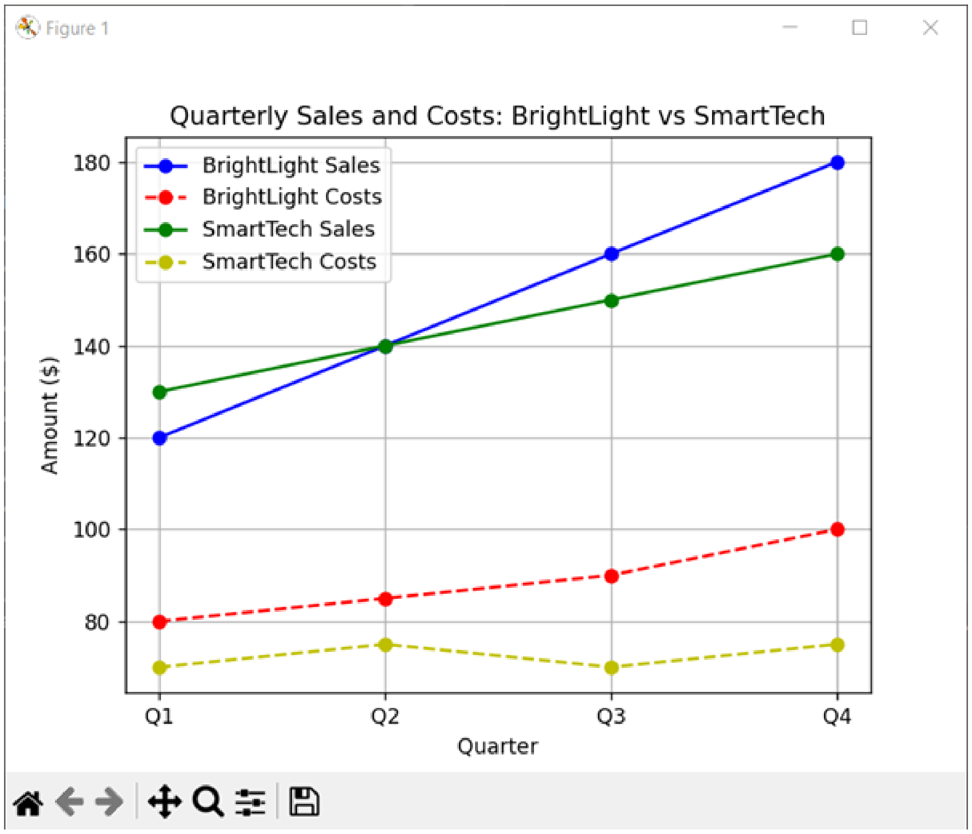
1.85 Question 43

How can the financial performance and cost management of BrightLight and SmartTech be compared over the four quarters? Select the best answer.

[7]:
```python
from IPython import display

display.Image("figure_1.png")
```

[7]:



a) Both BrightLight and SmartTech exhibit decreasing sales and decreasing operational expenses, indicating that both products are cutting cost effectively but may be facing declining demand.

b) SmartTech's sales are significantly higher that BrightLight's but its operational expenses are also proportionally higher, indicating that both products have similar efficiency levels and profit margins.

c) The data shows that BrightLight's operational expenses have remained constant while its sales have increased sharply, suggesting superior efficiency and cost management compared to SmartTech, which has increasing expenses and stable sales.

d) While both BrightLight and SmartTech show indresing sales over the four quarters, BrightLight incurs higer operational expenes relative to its sales compared to SmarthTech, suggesting SmartTech operates more efficiently with potentially higher profit margins.

1.86 Solution 43

d

The graph shows that although both products' sales are growing, BrightLight's operational expenses are increasing at a higher rate compared to its sales. This indicates less efficient cost management compared to SmartTech, which maintains relatively stable and lower operational expenses. Consequently, SmartTech operates more efficiently with potentially higher profit margins.

1.87 Question 44

Consider a database table named orders with the following columns: `order_id` (primary key), `customer_name`, `product_id`, and `order_date`. You are tasked with performing a complete CRUD cycle for an order with the following operations:

Create: Add a new order with `order_id` "1001" for customer "John Doe" with `product_id` "101" and `order_date` "2023-01-01".

Update: Change the product_id for the order with `order_id` "1001" to "102".

Read: Retrieve the full record for the order with `order_id` "1001".

Delete: Remove the order with `order_id` "1001" from the database.

Read: Attempt to retrieve the full record for the order with `order_id` "1001" again.

Select the sequence of SQL statements that accurately represent these operations.

a)

```
INSERT INTO orders (order_id, customer_name, product_id, order_date) VALUES ('1001', 'John Doe
UPDATE orders SET product_id = '102' WHERE customer_name = 'John Doe';
SELECT * FROM orders WHERE customer_name = 'John Doe';
DELETE FROM orders WHERE customer_name = 'John Doe';
SELECT * FROM orders WHERE customer_name = 'John Doe';
```

b)

```
INSERT INTO orders (order_id, customer_name, product_id, order_date) VALUES (John Doe', 'Doe',
UPDATE orders SET product_id = '102' WHERE order_date = '2023-01-01';
SELECT * FROM orders WHERE order_date = '2023-01-01';
DELETE FROM orders WHERE order_date = '2023-01-01';
SELECT * FROM orders WHERE order_date = '2023-01-01';
```

c)

```
INSERT INTO orders (order_id, customer_name, product_id, order_date) VALUES ('1001', 'John Doe
UPDATE orders SET product_id = '102' WHERE order_id = '1001';
SELECT * FROM orders WHERE order_id = '1001';
DELETE FROM orders WHERE order_id = '1001';
SELECT * FROM orders WHERE order_id = '1001';
```

d)

```
INSERT INTO orders (order_id, customer_name, product_id, order_date) VALUES ('John Doe', '101'
UPDATE orders SET product_id = '102' WHERE product_id = '101';
SELECT * FROM orders WHERE product_id = '101';
DELETE FROM orders WHERE product_id = '101';
SELECT * FROM orders WHERE product_id = '101';
```

1.88 Solution 44

c

This sequence correctly performs the CRUD operations using the order_id as the identifier. Using the order_id ensures that the operations are accurately targeted to the specific order. After the deletion, attempting to read the order again should return no results, indicating the order has been successfully deleted.

1.89 Question 45

James is analyzing customer satisfaction ratings for a company's products across different demographics. Which type of visualization would be most appropriate for James to use in representing this data?

a) Line Chart.

b) Scatter plot

c) Stacked bar chat.

d) Pie chart

1.90 Solution 45

c

Stacked bar chart allows James to compare customer satisfaction ratings across different demographics within each product category, making it suitable for visualizing this type of categorical data.

Creado por:

Isabel Maniega