Python Institute

PCED-30-01

Certified Entry-Level Data Analyst with Python

QUESTION & ANSWERS

QUESTION: 1

In a Pandas DataFrame, what would be the most efficient way to group the data by one column and apply multiple aggregate functions?

Option A : df.groupby('column').agg(['sum', 'mean'])

Option B : df.groupby('column').apply(lambda x: x.sum())

df.groupby('column').apply(lambda x: x.mean())
Option C : df.groupby('column').sum()

df.groupby('column').mean()
Option D : df['column'].sum()

df['column'].mean()

Correct Answer: A

Explanation/Reference:

Correct answer:

df.groupby('column').agg(['sum', 'mean'])

Using .agg() with multiple functions within a list allows you to perform multiple aggregate functions in a single operation, making it the most efficient method.

Incorrect answers:

df.groupby('column').sum()

df.groupby('column').mean()

This approach will group the data twice, once for the sum and once for the mean. This is less efficient than using .agg() to perform both operations in one go.

df['column'].sum()

df['column'].mean()

This will calculate the sum and mean for the entire column, rather than grouping by another column first.

df.groupby('column').apply(lambda x: x.sum())

df.groupby('column').apply(lambda x: x.mean())

While this would work, using .apply() with a lambda function for simple aggregation tasks is generally less efficient than using .agg() with built-in functions.

QUESTION: 2

You're fetching data from an API that returns data in JSON format. After fetching the data using Python's requests library, what should you do next to convert the returned data into a Python dictionary?

Option A : data_dict = response.json()

Option B : import json

data_dict = json.load(response.text)

Option C : import json

data_dict = json.loads(response.content)

Option D : data_dict = eval(response.text)

Correct Answer: A

Explanation/Reference:

Correct answer:

data_dict = response.json()

The json() method on a Response object decodes the JSON data into a Python dictionary.

Incorrect answers:

import json

data_dict = json.load(response.text)

json.load() is used to read from a file-like object, not a string.

import json

data_dict = json.loads(response.content)

json.loads() expects a string, while response.content is bytes.

data_dict = eval(response.text)

Using eval() is unsafe and should not be used to parse JSON data.

QUESTION: 3

You're developing a Python script to collect real-time data from various weather stations. Which approach would best help you implement robust data validation procedures to ensure the reliability and accuracy of the data?

Option A : Use type hints in Python to enforce data types

Option B : Implement checksum validation for incoming data packets

Option C : Employ a schema validation library to enforce structure and type constraints

Option D : Accept data only from weather stations that use the latest sensor technology

Correct Answer: C

Explanation/Reference:

Employ a schema validation library to enforce structure and type constraints -> Correct. Schema validation libraries can enforce data types, acceptable ranges, and data structure, providing robust validation.

Use type hints in Python to enforce data types -> Incorrect. Type hints in Python are not enforceable at runtime and serve mainly for readability and IDE support.

Implement checksum validation for incoming data packets -> Incorrect. Checksum validation ensures the integrity of data during transfer but does not validate its accuracy or reliability.

Accept data only from weather stations that use the latest sensor technology -> Incorrect. The type of technology used in weather stations does not guarantee the accuracy or reliability of the data.

QUESTION: 4

You have a dataset with several missing values. Which technique is generally NOT advisable for dealing with missing values in a dataset intended for machine learning modeling?

Option A : Removing rows with missing values

Option B : Using a machine learning algorithm that can handle missing values

Option C : Filling all missing values with zeros

Option D : Imputing missing values using mean or median

Correct Answer: C

Explanation/Reference:

Filling all missing values with zeros -> Correct. This is generally not advisable unless you're certain that zero is a meaningful value for all the missing fields; otherwise, it can introduce bias.

Imputing missing values using mean or median -> Incorrect. This is a common technique to handle missing numerical values and is generally safe for many machine learning algorithms.

Removing rows with missing values -> Incorrect. This may be appropriate if the number of rows with missing data is small and their removal does not introduce bias.

Using a machine learning algorithm that can handle missing values -> Incorrect. Some algorithms can handle missing values without any issue.

QUESTION: 5

When using Python to collect data from an API, which of the following statements about OAuth (Open Authorization) is true?

Option A : OAuth is only used for basic username and password authentication.

Option B : OAuth is a protocol used to convert JSON data to XML data.

Option C : OAuth encrypts the data payload for each API request.

Option D : OAuth allows users to approve applications to act on their behalf without sharing their password.

Correct Answer: D

Explanation/Reference:

OAuth allows users to approve applications to act on their behalf without sharing their password. -> Correct. OAuth allows for token-based authentication, enabling users to grant specific permissions to applications without exposing their password.

OAuth is a protocol used to convert JSON data to XML data. -> Incorrect. OAuth is an authentication protocol, not a data conversion protocol.

OAuth is only used for basic username and password authentication. -> Incorrect. OAuth is more sophisticated than basic username and password authentication and often uses tokens.

OAuth encrypts the data payload for each API request. -> Incorrect. OAuth is about authentication, not about data encryption.

QUESTION: 6

You are developing a web application with Flask and need to collect user-generated data through web forms. What is the most appropriate way to securely collect this data in Python?

Option A : Use Python's input() function to collect data from users.

Option B : Use Flask's request.form and sanitize the inputs before storing them.

Option C : Use JavaScript to collect the form data and send it to Python using cookies.

Option D : Directly insert form data into a SQL database without modification.

Correct Answer: B

Explanation/Reference:

Use Flask's request.form and sanitize the inputs before storing them. -> Correct. Using Flask's request.form is the most appropriate way to collect POST data from a web form in a Flask application. It's crucial to sanitize these inputs before storing them to prevent SQL injection, cross-site scripting, or other security vulnerabilities. The other methods are either insecure or inappropriate for the context described.

QUESTION: 7

When debugging a complex Python script for data analysis, which of the following practices is least recommended?

Option A : Ignoring warnings as long as the script does not crash

Option B : Writing unit tests to verify each function

Option C : Inserting print statements at different points in the script

Option D : Using a debugger to step through the code

Correct Answer: A

Explanation/Reference:

Ignoring warnings as long as the script does not crash -> Correct. This is not recommended, as warnings often indicate potential issues that could affect the script's performance or correctness.

Using a debugger to step through the code -> Incorrect. This is a good practice, as it allows you to examine the state of variables and the flow of execution.

Inserting print statements at different points in the script -> Incorrect. This is a common debugging technique that helps to trace the execution and identify issues.

Writing unit tests to verify each function -> Incorrect. This is a best practice that helps to ensure each function works as expected, which can aid in debugging.

QUESTION: 8

Suppose you have a DataFrame df with columns Name, Age, Gender, and Salary. Which of the following code snippets will filter the DataFrame to include only rows where Age is more than 25 and Salary is less than 50000, and also sort the resulting DataFrame by Name?

Option A : df[(df['Age'] > 25) & (df['Salary'] < 50000)].sort_values('Name')

Option B : df.query('Age > 25 and Salary < 50000').sort('Name')

Option C : df.sort_values('Name').where(df['Age'] > 25 & df['Salary'] < 50000)

Option D : df.filter('Age' > 25 & 'Salary' < 50000).sort_values(by='Name')

Correct Answer: A

Explanation/Reference:

df[(df['Age'] > 25) & (df['Salary'] < 50000)].sort_values('Name') -> Correct. It performs the filtering and sorting in a single, efficient line of code.

df.query('Age > 25 and Salary < 50000').sort('Name') -> Incorrect. It uses .query correctly but uses .sort which is deprecated; the correct method is .sort_values.

df.filter('Age' > 25 & 'Salary' < 50000).sort_values(by='Name') -> Incorrect. It incorrectly uses the .filter method, which is used for different kinds of filtering.

df.sort_values('Name').where(df['Age'] > 25 & df['Salary'] < 50000) -> Incorrect. It first sorts the data and then filters it, which is less efficient.

QUESTION: 9

In Python, which of the following methods is not a common statistical approach to identify outliers within a dataset?

Option A : Interquartile Range (IQR)

Option B : Z-score

Option C : Tukey's Fences

Option D : Mean Squared Error (MSE)

Correct Answer: D

Explanation/Reference:

Mean Squared Error (MSE) -> Correct. Mean Squared Error (MSE) is commonly used as a loss function for regression models, rather than for identifying outliers in a dataset. The other methods listed (Tukey's Fences, Z-score, MAD, IQR) are all commonly used for detecting outliers statistically.

QUESTION: 10

When dealing with a categorical feature that has high cardinality in a dataset, which of the following approaches is most advisable for one-hot encoding?

Option A : Use label encoding instead of one-hot encoding

Option B : Group less frequent categories into a single "Other" category before one-hot encoding

Option C : Remove the feature from the dataset

Option D : One-hot encode all unique values in the feature

Correct Answer: B

Explanation/Reference:

Group less frequent categories into a single "Other" category before one-hot encoding -> Correct. Grouping less frequent categories into an "Other" category can simplify the feature space while retaining information.

One-hot encode all unique values in the feature -> Incorrect. One-hot encoding all unique values can lead to a very large feature space, potentially causing issues like overfitting.

Use label encoding instead of one-hot encoding -> Incorrect. Label encoding might introduce ordinal relationships that do not exist in the data.

Remove the feature from the dataset -> Incorrect. Removing the feature might lead to loss of valuable information.

QUESTION: 11

You're using a Decision Tree classifier to predict whether a transaction is fraudulent. After training, you find that the classifier has a high accuracy on the training set but performs poorly on the test set. Which of the following techniques is most suitable to improve the model's performance?

Option A : Use a different kernel function.

Option B : Prune the decision tree.

Option C : Apply grid search without cross-validation.

Option D : Change the evaluation metric to accuracy.

Correct Answer: B

Explanation/Reference:

Prune the decision tree. -> Correct. Pruning can help reduce the complexity of the tree, which can be beneficial in reducing overfitting, thus potentially improving test set performance.

Use a different kernel function. -> Incorrect. Decision Trees don't use kernel functions; this is more applicable to Support Vector Machines.

Change the evaluation metric to accuracy. -> Incorrect. Changing the evaluation metric won't solve the issue of poor test set performance.

Apply grid search without cross-validation. -> Incorrect. Grid search is a hyperparameter optimization technique, and using it without cross-validation could lead to more overfitting.

QUESTION: 12

You've built a logistic regression model to predict whether a loan applicant will default on a loan (Default or No Default). After training, the model's AUC-ROC score is 0.95 on the training set but only 0.60 on the test set. What is most likely the reason for this discrepancy?

Option A : The model is overfitting.

Option B : There is multicollinearity among features.

Option C : The model is underfitting.

Option D : The model's hyperparameters are not tuned.

Correct Answer: A

Explanation/Reference:

The model is overfitting. -> Correct. Overfitting is when a model learns the training data too well, including its noise, and performs poorly on new, unseen data.

The model is underfitting. -> Incorrect. Underfitting would typically result in poor performance on both the training

and test datasets.

The model's hyperparameters are not tuned. -> Incorrect. While hyperparameter tuning might improve performance, it wouldn't explain a high training score and a low test score.

There is multicollinearity among features. -> Incorrect. Multicollinearity could be an issue, but it's not likely to be the main cause for the observed discrepancy between training and test scores.

QUESTION: 13

Which of the following best describes the concept of 'State' in a Dash application?

Option A : The values of all current properties of Dash components

Option B : The current user logged into the application

Option C : The geographical location of the server

Option D : The database connection status

Correct Answer: A

Explanation/Reference:

The values of all current properties of Dash components -> Correct. In Dash, 'State' refers to the current values of all the properties of the components, which can be read or updated by callback functions.

The current user logged into the application -> Incorrect. While this could be a state variable, it is not what is generally meant by 'State' in a Dash application.

The geographical location of the server -> Incorrect. This is unrelated to Dash's concept of State.

The database connection status -> Incorrect. This could be considered part of the application state, but it's not the concept of 'State' within Dash.

QUESTION: 14

Which of the following Python Pandas methods is most suitable for removing duplicate rows from a DataFrame?

Option A : df.dropna()

Option B : df.duplicated()

Option C : df.fillna()

Option D : df.drop_duplicates()

Correct Answer: D

Explanation/Reference:

df.drop_duplicates() -> Correct. This is the correct method for removing duplicate rows from a DataFrame.

df.duplicated() -> Incorrect. This method identifies duplicate rows but doesn't remove them.

df.dropna() -> Incorrect. This method is used to remove rows containing missing values.

df.fillna() -> Incorrect. This method is used to replace missing values with a specified value or method.

QUESTION: 15

You are working with a small dataset and want to assess your model's performance. Your colleague recommends using k-fold cross-validation. However, you are concerned about the reliability of the evaluation. What cross-validation technique would be most appropriate for your small dataset?

Option A : Leave-One-Out Cross-Validation (LOOCV)

Option B : Shuffle-Split Cross-Validation

Option C : Regular k-Fold Cross-Validation

Option D : Stratified k-Fold Cross-Validation

Option E : Time Series Cross-Validation

Correct Answer: A

Explanation/Reference:

Leave-One-Out Cross-Validation (LOOCV) -> Correct. Leave-One-Out Cross-Validation (LOOCV) is generally more

reliable for small datasets. In LOOCV, k is set to the number of data points in the dataset, meaning each data point gets its turn as a test set. This maximizes both the training and the test dataset, making it suitable for small datasets.

QUESTION: 16

Your dataset includes several categorical variables. What is the most efficient way to convert these variables into a format that can be fed into a machine learning model?

Option A : Use label encoding indiscriminately

Option B : Apply one-hot encoding selectively

Option C : Convert all categories to their alphabetical rank

Option D : Use frequency encoding for all categorical variables

Correct Answer: B

Explanation/Reference:

Apply one-hot encoding selectively -> Correct. One-hot encoding can effectively convert categorical variables into a numerical format but should be used selectively to avoid the "curse of dimensionality."

Convert all categories to their alphabetical rank -> Incorrect. Alphabetical ranking will impose an ordinal relationship that may not exist in the data.

Use frequency encoding for all categorical variables -> Incorrect. Frequency encoding can introduce collinearity and does not always capture the information effectively.

Use label encoding indiscriminately-> Incorrect. Label encoding can introduce an ordinal relationship between categories that may not actually have such a relationship, affecting the model's performance.

QUESTION: 17

Which of the following cross-validation techniques is most appropriate when you are working with a dataset where the target variable's classes are imbalanced?

Option A : Leave-One-Out Cross-Validation (LOOCV)

Option B : k-Fold Cross-Validation

Option C : Stratified k-Fold Cross-Validation

Option D : Random Split Cross-Validation

Correct Answer: C

Explanation/Reference:

Stratified k-Fold Cross-Validation -> Correct. Stratified k-Fold Cross-Validation ensures each fold is made by preserving the percentage of samples for each class, making it suitable for imbalanced datasets.

k-Fold Cross-Validation -> Incorrect. k-Fold Cross-Validation does not ensure that each fold has a representative ratio of the target class, which is crucial in imbalanced datasets.

Leave-One-Out Cross-Validation (LOOCV) -> Incorrect. LOOCV can be computationally expensive and may also not guarantee a balanced class distribution in each fold.

Random Split Cross-Validation -> Incorrect. Random Split might not guarantee balanced classes in each fold.

QUESTION: 18

Your Python script is giving you unexpected results in data analysis. Which of the following is generally not advisable for debugging this issue?

Option A : Review and validate data inputs and transformations

Option B : Use conditional breakpoints to isolate the problem

Option C : Examine the Python traceback to identify where the error originates

Option D : Suppress all exceptions using a general except block

Correct Answer: D

Explanation/Reference:

Suppress all exceptions using a general except block -> Correct. This is generally not advisable because you may

overlook the root cause of the problem, making it difficult to debug effectively.

Examine the Python traceback to identify where the error originates -> Incorrect. Tracebacks can be invaluable for finding the origin of an issue in your script.

Use conditional breakpoints to isolate the problem -> Incorrect. Conditional breakpoints allow you to halt the script under specific conditions, making it easier to identify problems.

Review and validate data inputs and transformations -> Incorrect. Ensuring that your data is as expected at all stages of your script can help identify where things are going wrong.

QUESTION: 19

Which of the following methods is least likely to be effective when dealing with outliers in a dataset while using Python's Pandas library?

Option A : Imputation

Option B : Deleting the feature column

Option C : Capping

Option D : Transformation

Correct Answer: B

Explanation/Reference:

Deleting the feature column -> Correct. Deleting the feature column to deal with outliers is generally not recommended unless the feature is entirely irrelevant or unreliable. This method removes too much information and can substantially reduce the predictive power of the model.

QUESTION: 20

When would you prefer to use t-SNE over PCA for visualizing high-dimensional data?

Option A : When you need to capture linear relationships in the data.

Option B : When computational efficiency is crucial.

Option C : When you want to visualize clusters or groups in the data.

Option D : When the dataset has fewer than three features.

Correct Answer: C

Explanation/Reference:

When you want to visualize clusters or groups in the data. -> Correct. t-SNE is particularly useful for visualizing high-dimensional data where clusters or groups exist.

When computational efficiency is crucial. -> Incorrect. t-SNE is computationally more intensive than PCA.

When you need to capture linear relationships in the data. -> Incorrect. PCA is better suited for capturing linear relationships.

When the dataset has fewer than three features. -> Incorrect. For low-dimensional data, other techniques such as basic scatter plots would suffice.

QUESTION: 21

You are trying to improve a linear regression model by adding polynomial features. After adding the polynomial features, you observe that the model performs much better on the training set but worse on the validation set. What is likely happening?

Option A : The model is underfitting the data.

Option B : The model's learning rate has increased.

Option C : The model is overfitting the data.

Option D : The model's complexity has been reduced.

Correct Answer: C

Explanation/Reference:

The model is overfitting the data. -> Correct. Adding polynomial features increases model complexity, making it more likely to overfit the training data, resulting in poor performance on the validation set.

The model is underfitting the data. -> Incorrect. Underfitting would result in poor performance on both training and validation sets, which isn't the case here.

The model's learning rate has increased. -> Incorrect. The learning rate isn't inherently affected by the addition of polynomial features.

The model's complexity has been reduced. -> Incorrect. The model's complexity has increased due to the addition of polynomial features, not reduced.

QUESTION: 22

After performing PCA on a dataset with 10 features, you get two principal components. Which of the following statements is true regarding these components?

Option A : Each principal component represents a cluster in the original dataset.

Option B : Principal components are scaled versions of the original features.

Option C : Each principal component is a linear combination of the original features.

Option D : The first principal component is always the most important feature in the original dataset.

Correct Answer: C

Explanation/Reference:

Each principal component is a linear combination of the original features. -> Correct. PCA works by creating new 'features' through linear combinations of the original features.

The first principal component is always the most important feature in the original dataset. -> Incorrect. The first principal component captures the most variance but isn't equivalent to a single original feature.

Principal components are scaled versions of the original features. -> Incorrect. They are linear combinations, not scaled versions.

Each principal component represents a cluster in the original dataset. -> Incorrect. PCA does not perform clustering.

QUESTION: 23

You have a DataFrame with a 'date' column in string format ('YYYY-MM-DD'). How would you convert this column to Pandas datetime format?

Option A : df['date'].convert_to('datetime')

Option B : df['date'].astype('datetime64')

Option C : df['date'] = df['date'].parse('%Y-%m-%d')

Option D : pd.to_datetime(df['date'], format='%Y-%m-%d')

Correct Answer: D

Explanation/Reference:

pd.to_datetime(df['date'], format='%Y-%m-%d') -> Correct. This is the correct way to convert a 'date' column in string format to Pandas datetime format.

df['date'].astype('datetime64') -> Incorrect. This will create a new Series but won't modify the DataFrame unless assigned back.

df['date'] = df['date'].parse('%Y-%m-%d') -> Incorrect. There's no parse method like this in Pandas for Series or DataFrame.

df['date'].convert_to('datetime') -> Incorrect. The method convert_to doesn't exist in Pandas.

QUESTION: 24

You are building a data pipeline that requires you to load data from an API, transform it, and then save it to a database. Which of the following is the best approach to structure your script for maintainability and reusability?

Option A : Use global variables to hold the data as it moves through the pipeline.

Option B : Implement all operations as nested loops inside a main loop.

Option C : Write the entire workflow in a single function.

Option D : Separate the loading, transforming, and saving operations into individual functions.

Correct Answer: D

Explanation/Reference:

Separate the loading, transforming, and saving operations into individual functions. -> Correct. This approach aligns with the Single Responsibility Principle, making the code more modular, maintainable, and reusable.

Write the entire workflow in a single function. -> Incorrect. This will make the code hard to read, maintain, and debug.

Use global variables to hold the data as it moves through the pipeline. -> Incorrect. Using global variables makes it harder to track the flow of data and debug the script.

Implement all operations as nested loops inside a main loop. -> Incorrect. This can make the script hard to read and maintain, especially as complexity grows.

QUESTION: 25

You are required to present a comparative analysis of the sales data of five different products over the last year. What would be the most effective way to visualize this data?

Option A : Heatmap

Option B : Radar Chart

Option C : 3D Surface Plot

Option D : Multiple Line Graphs on the Same Plot

Correct Answer: D

Explanation/Reference:

Multiple Line Graphs on the Same Plot -> Correct. Multiple Line Graphs on the same plot would effectively show the trends of different products over the same period, making it easier to perform a comparative analysis.

Radar Chart -> Incorrect. Radar Charts are generally used for displaying multivariate data on multiple axes starting from the same point, not ideal for time-series comparative analysis.

3D Surface Plot -> Incorrect. 3D Surface Plots can be complex and are generally not the best choice for

straightforward comparative analysis.

Heatmap -> Incorrect. Heatmaps are better for showing density or intensity of variables rather than a comparative analysis over time.

QUESTION: 26

Consider the Python code snippet below:

```
numbers = [1, 2, 3, 4, 5]
filtered_numbers = list(filter(lambda x: x % 2 == 1, numbers))
result = len(filtered_numbers)
```

What will be the value of result after executing the code?

Option A : 3

Option B : 4

Option C : 1

Option D : 5

Option E : 2

Correct Answer: A

Explanation/Reference:

3 -> Correct.

The filter() function filters the elements of numbers based on the lambda function, which keeps only the odd numbers.

filtered_numbers will then be [1, 3, 5].

The length of [1, 3, 5] is 3, which will be stored in the variable result.

QUESTION: 27

You notice that the transformation phase of the ETL process is extremely slow. You suspect that the issue lies in the inefficient handling of a large DataFrame in Pandas. Which of the following should be your first step to troubleshoot?

Option A : Break the data into smaller chunks and process them sequentially.

Option B : Switch to a different data processing library like Dask or PySpark.

Option C : Upgrade the hardware to speed up the process.

Option D : Review and optimize the Pandas code for performance.

Correct Answer: D

Explanation/Reference:

Review and optimize the Pandas code for performance. -> Correct. The first step should be to review the existing code for any inefficiencies or bottlenecks that could be causing the slowdown.

Upgrade the hardware to speed up the process. -> Incorrect. While hardware upgrades can provide some improvements, they are often a costly and temporary fix.

Switch to a different data processing library like Dask or PySpark. -> Incorrect. Switching libraries is a drastic measure that should only be considered if optimizing the current code doesn't help.

Break the data into smaller chunks and process them sequentially. -> Incorrect. Breaking the data into chunks can improve performance but doesn't address potential inefficiencies in the code.

QUESTION: 28

You have the following DataFrame named sales_data:

```
sales_data = pd.DataFrame({
    'Year': [2021, 2021, 2022, 2022],
    'Quarter': [1, 2, 1, 2],
    'Revenue': [1000, 1100, 1200, 1300]
})
```

You want to reshape the DataFrame such that the Revenue for each Quarter becomes its own column, and the Year remains as the index. Which of the following code snippets accomplishes this?

Option A : sales_data.pivot(index='Year', columns='Quarter', values='Revenue')

Option B : sales_data.melt(id_vars=['Year'], value_vars=['Revenue'])

Option C : pd.crosstab(index=sales_data['Year'], columns=sales_data['Quarter'], values=sales_data['Revenue'], aggfunc='sum')

Option D : sales_data.groupby(['Year', 'Quarter']).Revenue.sum().unstack()

Correct Answer: A

Explanation/Reference:

sales_data.pivot(index='Year', columns='Quarter', values='Revenue') -> Correct. The pivot function is ideal for this straightforward reshaping. Setting the index to 'Year', columns to 'Quarter', and values to 'Revenue' will accomplish the task.

sales_data.melt(id_vars=['Year'], value_vars=['Revenue']) -> Incorrect. melt is used for transforming a DataFrame from a wide to a long format, not suitable for the desired output.

sales_data.groupby(['Year', 'Quarter']).Revenue.sum().unstack() -> Incorrect. This will also pivot the DataFrame but is more complex than needed for this task.

pd.crosstab(index=sales_data['Year'], columns=sales_data['Quarter'], values=sales_data['Revenue'], aggfunc='sum') -> Incorrect. pd.crosstab is more suited for frequency tables and not the most direct way to achieve this.

QUESTION: 29

You are tasked with preparing a dataset that has a mix of continuous, ordinal, and nominal variables for a machine learning model. What would be the most appropriate encoding method for ordinal variables?

Option A : Label encoding

Option B : Binary encoding

Option C : One-hot encoding

Option D : Frequency encoding

Correct Answer: A

Explanation/Reference:

Label encoding -> Correct. Label encoding is usually the most appropriate for ordinal variables as it maintains the ordinal relationship between the values.

One-hot encoding -> Incorrect. One-hot encoding is generally not suitable for ordinal variables as it doesn't capture the inherent order of the categories.

Binary encoding -> Incorrect. Binary encoding is generally used for categorical variables but fails to capture the ordinal nature.

Frequency encoding -> Incorrect. Frequency encoding also replaces each category with its frequency in the dataset, which doesn't capture the ordinality.

QUESTION: 30

In a large dataset, you notice that one column has some entries recorded in uppercase and others in lowercase. This inconsistency could potentially cause errors in downstream analyses. What is the best way to handle this issue?

Option A : Replace all text entries with 'NaN'

Option B : Delete the column to avoid inconsistencies

Option C : Convert all entries to lowercase for uniformity

Option D : Convert all entries to numerical format

Correct Answer: C

Explanation/Reference:

Convert all entries to lowercase for uniformity -> Correct. Converting all entries to lowercase (or uppercase) ensures that the data in the column is consistent and can be accurately analyzed.

Convert all entries to numerical format -> Incorrect. Converting to numerical format wouldn't make sense for text data and doesn't solve the problem.

Delete the column to avoid inconsistencies -> Incorrect. Deleting the column could lead to loss of potentially valuable

data.

Replace all text entries with 'NaN' -> Incorrect. Replacing with 'NaN' would mean losing all information in the column, which is not advisable.

QUESTION: 31

You are using Python to analyze data from IoT sensors. You notice that some data points are far off from what you'd expect. What is the most appropriate course of action to ensure data accuracy and reliability?

Option A : Consult with domain experts to interpret the outliers

Option B : Ignore the outliers as they add variance to your dataset

Option C : Replace the outliers with the median value of the dataset

Option D : Remove the outliers and proceed with the analysis

Correct Answer: A

Explanation/Reference:

Consult with domain experts to interpret the outliers -> Correct. Consulting with domain experts can provide valuable context and help you decide whether the outliers are errors or valuable data.

Remove the outliers and proceed with the analysis -> Incorrect. Simply removing outliers may compromise the integrity of the dataset if those outliers are valid data points.

Replace the outliers with the median value of the dataset -> Incorrect. Replacing outliers with median values could distort the dataset and make it less representative of the real-world conditions.

Ignore the outliers as they add variance to your dataset -> Incorrect. Ignoring outliers without investigating can compromise the reliability and accuracy of your analysis.

QUESTION: 32

You have a dataset with 15 features and you want to visualize it on a 2D plot for exploratory data analysis. What would be the first step in using PCA (Principal Component Analysis) to achieve this?

Option A : Perform clustering on the dataset

Option B : Apply a linear regression model to reduce dimensions

Option C : Use a scatter plot to visualize all the features

Option D : Standardize the dataset

Correct Answer: D

Explanation/Reference:

Standardize the dataset -> Correct. The first step in PCA is often to standardize the dataset so that each feature has a mean of 0 and a standard deviation of 1.

Perform clustering on the dataset -> Incorrect. Clustering is not the first step in PCA.

Use a scatter plot to visualize all the features -> Incorrect. A scatter plot is not useful for visualizing high-dimensional data directly.

Apply a linear regression model to reduce dimensions -> Incorrect. Linear regression is not used for dimensionality reduction in the context of PCA.

QUESTION: 33

In a given dataset, you have a categorical feature "State" that includes the 50 U.S. states. You decide to one-hot encode this feature. What potential issue should you be cautious about?

Option A : Introducing multicollinearity

Option B : Reducing the dimensionality of the dataset

Option C : Creating an imbalanced dataset

Option D : Inducing underfitting in the model

Correct Answer: A

Explanation/Reference:

Introducing multicollinearity -> Correct. One-hot encoding can introduce multicollinearity, especially if you include all dummy variables in the model. To avoid this issue, you can drop one of the one-hot encoded columns (the "drop-first" strategy).

Creating an imbalanced dataset -> Incorrect. One-hot encoding does not inherently create an imbalanced dataset; it only changes the representation of categorical variables.

Reducing the dimensionality of the dataset -> Incorrect. One-hot encoding actually increases the dimensionality of the dataset.

Inducing underfitting in the model -> Incorrect. One-hot encoding usually doesn't induce underfitting but can sometimes lead to overfitting due to the increased dimensionality.

QUESTION: 34

Which of the following techniques should be avoided to ensure that your Python data scripting code is easily understandable and follows best practices?

Option A : Using descriptive variable names

Option B : Refactoring repetitive code into reusable functions

Option C : Documenting functions with docstrings

Option D : Using single-letter variable names for main variables

Correct Answer: D

Explanation/Reference:

Using single-letter variable names for main variables -> Correct. This should be avoided, as it can make the code hard to understand and maintain.

Using descriptive variable names -> Incorrect. This is a best practice, as it makes the code self-explanatory.

Documenting functions with docstrings -> Incorrect. This is also a best practice, as it provides additional information about the functionâ€™s purpose, parameters, and return values.

Refactoring repetitive code into reusable functions -> Incorrect. This is considered a best practice, as it makes the code more modular and easier to maintain.

QUESTION: 35

Your dataset has multiple variables with different scales and units, making it challenging to compare them directly. You need to synthesize this data into a digestible report for decision-makers who are not data scientists. What should be your first step in preprocessing the data?

Option A : Perform natural language processing (NLP) on any text variables

Option B : Transform all variables into Z-scores

Option C : Conduct a chi-squared test for each variable

Option D : Apply hierarchical clustering to group similar variables

Correct Answer: B

Explanation/Reference:

Transform all variables into Z-scores -> Correct. Transforming all variables into Z-scores standardizes them, making it easier to compare variables that originally had different scales and units. This standardized data can be more easily interpreted and visualized, thereby making it more digestible for decision-makers. Other options like chi-squared tests or hierarchical clustering are useful for other purposes but won't address the immediate need to compare variables on a common scale.

QUESTION: 36

In a regression analysis of employee job satisfaction against years of experience, the p-value for the 'years of experience' variable is found to be 0.12. How should you interpret this result at a 0.05 significance level?

Option A : A p-value of 0.12 confirms that years of experience and job satisfaction are positively correlated.

Option B : Years of experience is not a statistically significant predictor of job satisfaction.

Option C : Years of experience is a statistically significant predictor of job satisfaction.

Option D : A p-value of 0.12 means that years of experience has a 12% impact on job satisfaction.

Correct Answer: B

Explanation/Reference:

Years of experience is not a statistically significant predictor of job satisfaction. -> Correct. Because the p-value is greater than the 0.05 significance level, we fail to reject the null hypothesis.

Years of experience is a statistically significant predictor of job satisfaction. -> Incorrect. A p-value greater than 0.05 does not indicate statistical significance.

A p-value of 0.12 means that years of experience has a 12% impact on job satisfaction. -> Incorrect. A p-value is not a measure of impact but the probability of observing a statistic as extreme as the one observed under the null hypothesis.

A p-value of 0.12 confirms that years of experience and job satisfaction are positively correlated. -> Incorrect. A p-value does not establish the direction of the relationship, only its statistical significance.

QUESTION: 37

You're working with a DataFrame df that contains the exam scores of students in three different classes: Math, Science, and History. You're interested in determining if the mean exam score significantly differs between these classes. What would be the most appropriate statistical test to use?

Option A : One-way ANOVA

Option B : Pearson Correlation Test

Option C : Paired t-test

Option D : Independent Samples t-test

Correct Answer: A

Explanation/Reference:

One-way ANOVA -> Correct. One-way ANOVA is used to compare the means of more than two independent groups, making it the appropriate test for this scenario.

Paired t-test -> Incorrect. A Paired t-test is used to compare means of the same group at different times or conditions, not applicable here.

Pearson Correlation Test -> Incorrect. Pearson Correlation Test is for measuring the linear relationship between two sets of data, not for comparing means across multiple groups.

Independent Samples t-test -> Incorrect. An Independent Samples t-test is used for comparing the means of two independent groups, not more than two.

QUESTION: 38

You discover that a text-based column in your DataFrame supposed to contain only alphanumeric characters has special characters in some entries. This constitutes "bad data" for your specific use-case. How can you most effectively clean this column?

Option A : Convert the entire DataFrame to a NumPy array and handle text cleaning

Option B : Use Regular Expressions with df.str.replace() to remove all special characters at once

Option C : Use df.replace() to replace each special character individually

Option D : Drop the column to avoid any complications

Correct Answer: B

Explanation/Reference:

Use Regular Expressions with df.str.replace() to remove all special characters at once -> Correct. Utilizing Regular Expressions with df.str.replace() allows for the removal of all special characters in an efficient manner, best suited for the task.

Use df.replace() to replace each special character individually -> Incorrect. Using df.replace() for each special character can be inefficient and time-consuming.

Convert the entire DataFrame to a NumPy array and handle text cleaning -> Incorrect. Converting to a NumPy array would lose the DataFrame functionalities that make the cleaning process more straightforward.

Drop the column to avoid any complications -> Incorrect. Dropping the column could result in loss of useful data.