Quiz Accurate DSA-C03 - Reliable SnowPro Advanced: Data Scientist Certification Exam Dumps Files

BONUS!!! Download part of Lead1Pass DSA-C03 dumps for free: https://drive.google.com/open?id=18bBi8aSnT6GCCjYE66PNOtnK3qIwMlGf

The desktop-based practice exam software is the first format that DSA-C03 provides to its customers. It allows candidates to track their progress from start to finish and provides an easily accessible progress report. This Snowflake DSA-C03 Practice Questions is customizable and mimics the real exam's format. It is user-friendly on Windows-based computers, and the product support staff is available to assist with any issues that may arise.

The learning material is available in three different easy-to-use formats. The first one is a DSA-C03 PDF dumps form and it is a printable and portable form. Users can save the notes by taking out prints of Snowflake DSA-C03 PDF questions or can access them via their smartphones, tablets, and laptops. The Snowflake DSA-C03 Pdf Dumps form can be used anywhere anytime and is essential for students who like to learn from their smart devices.

>> Reliable DSA-C03 Dumps Files <<

Reliable DSA-C03 Dumps Files | DSA-C03 100% Free Latest Training

The price for DSA-C03 training materials is quite reasonable, and no matter you are a student at school or an employee in the company, you can afford the expense. You just think that you only need to spend some money, and you can pass the exam and get the certificate, which is quite self-efficient. In addition, DSA-C03 Exam Dumps are edited by the professional experts, who are quite familiar with the professional knowledge and testing center, and the quality and accuracy can be guaranteed. We have 24 hours service stuff, and if you any questions about DSA-C03 training materials, just contact us.

Snowflake SnowPro Advanced: Data Scientist Certification Exam Sample Questions (Q142-Q147):

NEW QUESTION # 142
You are using Snowpark for Python to perform feature engineering on a large dataset stored in a Snowflake table named 'transactions'. You need to create a new feature called 'transaction_size category' based on the 'transaction_amount' column. The categories are defined as follows: Small (amount < 10), Medium (10 <= amount < 100), and Large (amount 100). You want to optimize performance by leveraging Snowflake's parallel processing capabilities. Which of the following Snowpark for Python code snippets is the MOST efficient and Pythonic way to achieve this?

- A.

```
import snowflake.snowpark.functions as F

def categorize_amount(amount):
    if amount < 10:
        return 'Small'
    elif amount < 100:
        return 'Medium'
    else:
        return 'Large'

spark.udf.register(categorize_amount, name='categorize_amount', return_type=StringType())

transactions_df = session.table('transactions')
transactions_df = transactions_df.with_column('transaction_size_category', F.call_udf('categorize_amount', F.col('transaction_amount')))
```

- B.

```
transactions_df = session.table('transactions')

def categorize_amount(row):
    amount = row['transaction_amount']
    if amount < 10:
        return 'Small'
    elif amount < 100:
        return 'Medium'
    else:
        return 'Large'

transactions_df = transactions_df.to_pandas()
transactions_df['transaction_size_category'] = transactions_df.apply(categorize_amount, axis=1)
transactions_df = session.create_dataframe(transactions_df)
```

- C.

```
rom snowflake.snowpark.functions import when, lit

ransactions_df = session.table('transactions')

ransactions_df = transactions_df.with_column(
    'transaction_size_category',
    when(transactions_df.transaction_amount < 10, lit('Small'))
    .when(transactions_df.transaction_amount < 100, lit('Medium')
    .otherwise(lit('Large'))
```

```
transactions_df = session.table('transactions')

transactions_df.createOrReplaceTempView('temp_transactions')

query = """
SELECT  ,
       CASE
           WHEN transaction_amount < 10 THEN 'Small'
           WHEN transaction_amount < 100 THEN 'Medium'
           ELSE 'Large'
       END AS transaction_size_category
FROM temp_transactions
"""

transactions_df = session.sql(query)
```

- D.
- E.

```
import pandas as pd

transactions_df = session.table('transactions').to_pandas()

def categorize_amount(amount):
    if amount < 10:
        return 'Small'
    elif amount < 100:
        return 'Medium'
    else:
        return 'Large'

transactions_df['transaction_size_category'] = transactions_df['transaction_amount'].apply(categorize_amount)

transactions_df = session.create_dataframe(transactions_df)
```


Answer: C

Explanation:
Option B is the most efficient and Pythonic. It leverages Snowpark's built-in 'when' function for conditional logic, which is optimized for execution within Snowflake's distributed processing environment. It avoids the overhead of creating a UDF (Option A) or transferring the data to Pandas (Options C and E), which would negate the benefits of Snowpark. Option D, while functional, involves creating a temporary view and executing SQL, which is less integrated with the Snowpark Python API than using 'when'. Furthermore, Option C and E converts the Snowpark dataframe to pandas dataframe which is a performance bottleneck.

NEW QUESTION # 143
You've built a complex machine learning model using scikit-learn and deployed it as a Python UDF in Snowflake. The UDF takes a JSON string as input, containing several numerical features, and returns a predicted probability However, you observe significant performance issues, particularly when processing large batches of data'. Which of the following approaches would be MOST effective in optimizing the performance of this UDF in Snowflake?

- A. Rewrite the UDF in Java or Scala to leverage the JVM's performance advantages over Python in Snowflake.
- B. Serialize the scikit-learn model using 'joblib' instead of 'pickle' for potentially faster deserialization within the UDF.
- C. Use Snowflake's vectorized UDF feature to process data in micro-batches, minimizing the overhead of repeated Python interpreter initialization.

- D. Increase the warehouse size to improve the overall compute resources available for UDF execution.
- E. Pre-process the input data outside of the UDF using SQL transformations, reducing the amount of data passed to the UDF and simplifying the Python code.

Answer: C,E

Explanation:
Vectorized UDFs (A) are designed specifically for performance optimization by processing data in batches, reducing overhead. Pre-processing data using SQL (D) can significantly reduce the complexity and data volume handled by the UDF. While 'joblib' (B) might offer a slight improvement, it's less impactful than vectorization. Increasing warehouse size (C) helps but doesn't address the underlying inefficiencies of repeated interpreter initialization. Rewriting in Java/Scala (E) is a viable option but requires significant effort and might not be necessary if vectorization and pre-processing are sufficient.

NEW QUESTION # 144
You are evaluating a binary classification model's performance using the Area Under the ROC Curve (AUC). You have the following predictions and actual values. What steps can you take to reliably calculate this in Snowflake, and which snippet represents a crucial part of that calculation? (Assume tables 'predictions' with columns 'predicted_probability' (FLOAT) and 'actual_value' (BOOLEAN); TRUE indicates positive class, FALSE indicates negative class). Which of the below code snippet should be used to calculate the 'True positive Rate' and 'False positive Rate' for different thresholds

- A. Export the 'predicted_probability' and 'actual_value' columns to a local Python environment and calculate the AUC using scikit-learn.
- B. The best way to calculate AUC is to randomly guess the probabilities and see how it performs.
- C. Calculate AUC directly within a Snowpark Python UDF using scikit-learn's function. This avoids data transfer overhead, making it highly efficient for large datasets. No further SQL is needed beyond querying the predictions data.

```
from snowflake.snowpark.functions import udf
from sklearn.metrics import roc_auc_score

@udf(return_type=FloatType(), input_types=[ArrayType(FloatType()), ArrayType(BooleanType())])
def calculate_auc(predicted_probabilities, actual_values):
 return roc_auc_score(actual_values, predicted_probabilities)
```

- D. The AUC cannot be reliably calculated within Snowflake due to limitations in SQL functionality for statistical analysis.
- E. Using only SQL, Create a temporary table with calculated True Positive Rate (TPR) and False Positive Rate (FPR) at different probability thresholds. Then, approximate the AUC using the trapezoidal rule.

```
CREATE OR REPLACE TEMP TABLE roc_curve AS
SELECT
 predicted_probability,
 SUM(CASE WHEN actual_value = TRUE THEN 1 ELSE 0 END) OVER (ORDER BY predicted_probability DESC) AS true_positives,
 SUM(CASE WHEN actual_value = FALSE THEN 1 ELSE 0 END) OVER (ORDER BY predicted_probability DESC) AS false_positives,
 (true_positives / (SELECT COUNT( ) FROM predictions WHERE actual_value = TRUE)) AS true_positive_rate,
 (false_positives / (SELECT COUNT( ) FROM predictions WHERE actual_value = FALSE)) AS false_positive_rate
FROM
 predictions
ORDER BY
 predicted_probability DESC;

SELECT
 SUM((false_positive_rate - LAG(false_positive_rate, 1, 0) OVER (ORDER BY predicted_probability))
 (true_positive_rate + LAG(true_positive_rate, 1, 0) OVER (ORDER BY predicted_probability)) / 2)
AS approximate_auc
FROM
 roc_curve;
```

Answer: C,E

Explanation:
Options A and C are correct. Option A demonstrates calculating AUC directly within Snowflake using a Snowpark Python UDF and scikit-learn's . This is efficient for large datasets as it avoids data transfer. Option C correctly outlines the process of calculating TPR and FPR using SQL and approximating AUC using the trapezoidal rule, another viable approach within Snowflake. Option B is incorrect; AUC can be calculated reliably within Snowflake. Option D is inefficient due to data transfer. Option E is blatantly incorrect.

NEW QUESTION # 145
You are working with a Snowflake table 'CUSTOMER TRANSACTIONS containing customer IDs, transaction dates, and transaction amounts. You need to identify customers who are likely to churn (stop making transactions) in the next month using a supervised learning model. Which of the following strategies would be MOST appropriate to define the target variable (churned vs. not churned) and create features for this churn prediction problem, suitable for a Snowflake-based machine learning pipeline?

- A. Define churn as customers with zero transactions in the last month. Create features like average transaction amount over the past year, number of transactions in the past month, and recency (time since the last transaction).
- B. Define churn based on a fixed threshold of total transaction value over a predefined period. Feature Engineering should purely consist of time series decomposition using Snowflake's built-in functions.
- C. Define churn as customers who haven't made a transaction in the past 6 months. Create a single feature representing the total number of transactions the customer has ever made.
- D. Define churn as customers with a significant decrease (e.g., 50%) in transaction amounts compared to the previous month. Create features based on demographic data and customer segmentation information, joined from other Snowflake tables.
- E. Define churn as customers with no transactions in the next month (the prediction target). Create features including: Recency (days since last transaction), Frequency (number of transactions in the past 3 months), Monetary Value (average transaction amount over the past 3 months), and trend of transaction amounts (using linear regression slope over the past 6 months).

Answer: E

Explanation:
Option E is the most appropriate strategy. Defining churn as the absence of transactions in the next month allows for building a predictive model. The features Recency, Frequency, Monetary Value (RFM), and the trend of transaction amounts provide a comprehensive view of the customer's transaction behavior, capturing both the current activity and the recent trend. Option A's definition of churn is based on the past month, which is not predictive. Option B's definition of churn is too sensitive to temporary fluctuations. Option C's approach limits the value of featurization. Option D lacks depth in featurization.

NEW QUESTION # 146
You are working with a Snowflake table 'CUSTOMER DATA containing customer information for a marketing campaign. The table includes columns like 'CUSTOMER ID', 'FIRST NAME', 'LAST NAME, 'EMAIL', 'PHONE NUMBER, 'ADDRESS, 'CITY, 'STATE, ZIP CODE, 'COUNTRY, 'PURCHASE HISTORY, 'CLICKSTREAM DATA, and 'OBSOLETE COLUMN'. You need to prepare this data for a machine learning model focused on predicting customer churn. Which of the following strategies and Snowpark Python code snippets would be MOST efficient and appropriate for removing irrelevant fields and handling potentially sensitive personal information while adhering to data governance policies? Assume data governance requires removing personally identifiable information (PII) that isn't strictly necessary for the churn model.

- A. Drop 'OBSOLETE_COLUMN'. For columns like and 'LAST_NAME' , consider aggregating into a single 'FULL_NAME feature if needed for some downstream task. Apply hashing or tokenization techniques to sensitive PII columns like and 'PHONE NUMBER using Snowpark UDFs, depending on the model's requirements. Drop columns like 'ADDRESS, 'CITY, 'STATE, ZIP_CODE, 'COUNTRY as they likely do not contribute to churn prediction. Example hashing function:

```
import hashlib

def hash_email(email: str) -> str:
    return hashlib.sha256(email.encode('utf-8')).hexdigest()
session.udf.register(hash_email, return_type=StringType(), name='hash_email_udf', replace=True)
```

- B. Dropping columns 'OBSOLETE_COLUMN' directly. Then, for PII columns ('FIRST_NAME, 'LAST_NAME, 'EMAIL', 'PHONE_NUMBER, 'ADDRESS', 'CITY', 'STATE' , , 'COUNTRY), create a separate table with anonymized or aggregated data for analysis unrelated to the churn model. Use Keep all PII columns but encrypt them using Snowflake's built-in encryption features to comply with data governance before building the model. Drop 'OBSOLETE COLUMN'.
- C. Keeping all columns as is and providing access to Data Scientists without any changes, relying on role based security access controls only.
- D. Dropping 'FIRST NAME, UST NAME, 'EMAIL', 'PHONE NUMBER, 'ADDRESS', 'CITY, 'STATE', ZIP CODE, 'COUNTRY and 'OBSOLETE_COLUMN' columns directly using 'LAST_NAME', 'EMAIL', 'PHONE_NUMBER', 'ADDRESS', 'CITY', 'STATE', 'ZIP_CODE', 'COUNTRY', without any further consideration.

Answer: C

Explanation:
Option D is the most comprehensive and adheres to best practices. It identifies and removes truly irrelevant columns ('OBSOLETE_COLUMN', and location details), handles PII appropriately using hashing and tokenization (or aggregation), and leverages Snowpark UDFs for custom data transformations. Options A is too simplistic and doesn't consider data governance. Option B is better than A, but more complex than needed if the data is not needed elsewhere. Option C doesn't address the principle of minimizing data exposure. Option E is unacceptable from a data governance and security perspective. The example code demonstrates how to register a UDF for hashing email addresses.

NEW QUESTION # 147
.....

DSA-C03 practice materials stand the test of time and harsh market, convey their sense of proficiency with passing rate up to 98 to 100 percent. They are 100 percent guaranteed DSA-C03 learning quiz. And our content of the DSA-C03 Exam Questions are based on real exam by whittling down superfluous knowledge without delinquent mistakes. At the same time, we always keep updating the DSA-C03 training guide to the most accurate and the latest.

DSA-C03 Latest Training: https://www.lead1pass.com/Snowflake/DSA-C03-practice-exam-dumps.html

Snowflake Reliable DSA-C03 Dumps Files Just go to the login page after successful login you can easily download your button by simple clicking on a download button, Lead1Pass is the ideal alternative for your SnowPro Advanced: Data Scientist Certification Exam (DSA-C03) test preparation because it combines all of these elements, Snowflake Reliable DSA-C03 Dumps Files You can easily understand how you have to prepare, Snowflake Reliable DSA-C03 Dumps Files How to identify the most helpful one from them?

The Cost of Healthcare and Six Sigma, We will DSA-C03 discuss the different object menus in detail in the chapters related to each objecttype, Just go to the login page after successful DSA-C03 Latest Training login you can easily download your button by simple clicking on a download button.

Avail [Updated 2025]! Snowflake DSA-C03 Exam Questions | Alleviate Exam Stress

Lead1Pass is the ideal alternative for your SnowPro Advanced: Data Scientist Certification Exam (DSA-C03) test preparation because it combines all of these elements, You can easily understand how you have to prepare.

How to identify the most helpful one from them, You will not feel confused.

- Reliable DSA-C03 Dumps Files - 100% Pass Quiz 2025 First-grade DSA-C03: SnowPro Advanced: Data Scientist Certification Exam Latest Training ◂ Easily obtain free download of ➥ DSA-C03 □ by searching on " www.free4dump.com " □DSA-C03 Downloadable PDF
- 100% Pass Snowflake - DSA-C03 –High Hit-Rate Reliable Dumps Files □ Open ➤ www.pdfvce.com □ enter [DSA-C03] and obtain a free download □Official DSA-C03 Practice Test
- DSA-C03 Latest Exam Duration □ Exam DSA-C03 Cram Review □ Practice DSA-C03 Test Online □ Open website ➥ www.pdfdumps.com □ and search for ▹ DSA-C03 ◃ for free download □Official DSA-C03 Practice Test
- 100% Pass Snowflake - DSA-C03 –High Hit-Rate Reliable Dumps Files ❣ Simply search for ▹ DSA-C03 ◃ for free download on ➡ www.pdfvce.com □□□ □Valid Exam DSA-C03 Registration
- DSA-C03 Trustworthy Exam Torrent □ DSA-C03 Exam Price □ Valid Test DSA-C03 Testking □ Search for ➤ DSA-C03 □ and download it for free on " www.passtestking.com " website □Real DSA-C03 Exams
- 2025 Reliable DSA-C03 Dumps Files | Latest 100% Free DSA-C03 Latest Training □ Search for ➠ DSA-C03 □ and download it for free on { www.pdfvce.com } website □Valid Test DSA-C03 Tutorial
- DSA-C03 testing engine training online | DSA-C03 test dumps □ Search for ✔ DSA-C03 □✔□ and download it for free on { www.exam4pdf.com } website □Real DSA-C03 Exams
- DSA-C03 Online Training Materials □ Real DSA-C03 Exams □ Valid Exam DSA-C03 Registration □ Search for 【 DSA-C03 】 and download it for free immediately on ⇛ www.pdfvce.com ⇚ □Latest DSA-C03 Test Cram
- DSA-C03 Exams Torrent □ DSA-C03 Valid Braindumps Pdf □ Valid DSA-C03 Exam Experience □ Download ▸ DSA-C03 ◂ for free by simply entering ➤ www.pdfdumps.com □ website □DSA-C03 Exams Torrent
- DSA-C03 Exams Torrent □ DSA-C03 Trustworthy Exam Torrent □ DSA-C03 Valid Braindumps Pdf □ Easily obtain free download of ➠ DSA-C03 □ by searching on 「 www.pdfvce.com 」 □DSA-C03 Exams Torrent
- DSA-C03 Exam Price □ Exam DSA-C03 Cram Review □ Valid Test DSA-C03 Testking ❤□ Go to website □ www.prep4pass.com □ open and search for 《 DSA-C03 》 to download for free □DSA-C03 Trustworthy Exam Torrent

- academy.hypemagazine.co.za, www.stes.tyc.edu.tw, ncon.edu.sa, indianagriexam.com, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, daotao.wisebusiness.edu.vn, freestudy247.com, sarahmdash.com, easystartupit.com, Disposable vapes

P.S. Free & New DSA-C03 dumps are available on Google Drive shared by Lead1Pass: https://drive.google.com/open?id=18bBi8aSnT6GCCjYE66PNOtnK3qIwMlGf