DUMPSARENA

SnowPro Advanced: Data Scientist Certification Exam

Snowflake DSA-C02

Version Demo

Total Demo Questions: 8

Total Premium Questions: 86

Buy Premium PDF

https://dumpsarena.co

sales@dumpsarena.co

sales@dumpsarena.co
dumpsarena.co

Topic Break Down

Topic	No. of Questions
Topic 1, Data Science Fundamentals	21
Topic 2, Data Preparation	34
Topic 3, Model Development	19
Topic 4, Model Deployment	12
Total	86

QUESTION NO: 1

Select the Correct Statements regarding Normalization?

A. Normalization technique uses minimum and max values for scaling of model.

B. Normalization technique uses mean and standard deviation for scaling of model.

C. Scikit-Learn provides a transformer RecommendedScaler for Normalization.

D. Normalization got affected by outliers.

ANSWER: A D

Explanation:

Normalization commonly refers to min-max feature scaling, which transforms each numeric feature using its observed minimum and maximum values. The transformation maps values into a specified interval, with the commonly used default interval being 0 to 1. In scikit-learn, this approach is implemented by `MinMaxScaler`, which learns the minimum and maximum from the training data and applies the corresponding linear transformation to later data. Because the learned extrema determine the scale, normalization is sensitive to outliers: an unusually large or small observation can greatly expand the feature range and compress the transformed values of most ordinary observations into a narrow portion of the target interval. This can reduce the practical usefulness of the scaled feature for many machine-learning algorithms. Therefore, the statements that normalization uses minimum and maximum values for scaling and that normalization is affected by outliers are correct. See the scikit-learn MinMaxScaler API documentation and its preprocessing guide for the scaling formula, behavior, and implementation details.

QUESTION NO: 2

What Can Snowflake Data Scientist do in the Snowflake Marketplace as Consumer?

A. Discover and test third-party data sources.

B. Receive frictionless access to raw data products from vendors.

C. Combine new datasets with your existing data in Snowflake to derive new business in-sights.

D. Use the business intelligence (BI)/ML/Deep learning tools of her choice.

ANSWER: A B C D

Explanation:

Snowflake Marketplace consumers can discover data products offered by providers and, where available, request or obtain access so they can evaluate third-party data for their analytical and data-science needs. Shared listings provide access within Snowflake rather than requiring consumers to build and operate separate ingestion processes for the provider’s data. After access is granted, consumers can use the shared data alongside data already held in their Snowflake account. This enables joins, feature engineering, analysis, and the development of richer business insights without first copying the provider’s data into an external platform.

Marketplace-delivered data is designed to be governed and maintained by the provider, with updates made available to the consumer through the sharing model. Consumers can then query and analyze those data products using their chosen Snowflake-compatible analytics environment, including BI and data-science workflows. Snowflake supports a broad ecosystem of partner tools as well as native capabilities for SQL, Python, machine learning, and related analytical work. The Marketplace documentation describes how consumers discover listings and access data products, while the Snowflake ecosystem documentation describes using partner technologies with Snowflake. See Snowflake Marketplace consumer documentation and the Snowflake ecosystem.

QUESTION NO: 3

Select the Data Science Tools which are known to provide native connectivity to Snowflake?

A. Denodo

B. DvSUM

C. DiYotta

D. HEX

ANSWER: D

Explanation:

HEX is correct because Hex is a collaborative analytics and data science workspace with a native Snowflake data connection. A Hex project can connect directly to a Snowflake account, use Snowflake credentials and connection parameters, browse available database objects, and execute SQL queries against Snowflake. This enables analysts and data scientists to combine SQL-based exploration with Python- and notebook-oriented workflows while keeping the underlying data in Snowflake rather than requiring an extract to a separate analytical store. Hex also documents Snowflake-specific configuration options, including supported authentication approaches and settings for establishing the connection. In Snowflake's ecosystem, this type of direct connection supports governed, scalable data science and analytics workflows: computation can be pushed to Snowflake warehouses, and access remains subject to the organization's Snowflake roles and permissions. The relevant product documentation is available in the Hex Snowflake connection guide. For the Snowflake side of the integration model, see Snowflake's access control overview, which describes the role-based authorization applied when tools connect and query Snowflake data.

QUESTION NO: 4

Which one of the following is not the key component while designing External functions within Snowflake?

A. Remote Service

B. API Integration

C. UDF Service

D. Proxy Service

ANSWER: C

Explanation:

UDF Service is correct because it is not a defined architectural component or Snowflake object used to implement an external function. Snowflake external functions are UDFs whose executable logic resides outside Snowflake, but Snowflake does not use the term "UDF Service" for a required service in this architecture. Instead, the external-function design connects Snowflake to externally executed code through defined components: a remote service that processes requests, a proxy service that mediates the communication, and an API integration that stores the authorization and security configuration Snowflake needs to work with the endpoint. The external function itself is created in Snowflake and contains metadata used to invoke the external endpoint, including its API integration and proxy-service URL. The external code can be implemented with a variety of HTTP-based technologies, including serverless platforms, but its implementation does not create a Snowflake component named "UDF Service." This distinction is important when configuring external functions: use Snowflake's API integration object to establish authenticated access and expose the remote logic through a supported proxy/API management layer. Snowflake documents this architecture and the required objects in its external-functions overview and creation reference: Introduction to external functions and CREATE EXTERNAL FUNCTION.

QUESTION NO: 5

Mark the Incorrect understanding of Data Scientist about Streams?

A. Streams on views support both local views and views shared using Snowflake Secure Data Sharing, including secure views.

B. Streams can track changes in materialized views.

C. Streams itself does not contain any table data.

D. Streams do not support repeatable read isolation.

ANSWER: B D

Explanation:

The incorrect understandings are “Streams can track changes in materialized views.” and “Streams do not support repeatable read isolation.” Snowflake streams provide change data capture for supported source objects, including tables and certain views, but a stream cannot be created on a materialized view. Materialized views therefore cannot serve as a stream source for tracking changes. Snowflake’s Streams documentation explicitly identifies this limitation while describing supported stream source objects and view-stream behavior.

Streams do support repeatable-read isolation semantics. Within a transaction, multiple statements that query the same stream see a consistent, repeatable set of change records. The returned delta is determined using the stream’s current offset and the transaction start time. If the transaction commits after consuming the stream in a data-manipulation operation, the stream offset advances to that transaction’s start time; if the transaction does not commit, the position remains unchanged. This behavior lets a transactional workload process a stable CDC window reliably, even when concurrent changes occur on the source object. See Snowflake’s Introduction to streams and Transactions documentation for the supported source-object limitations and stream isolation behavior.

QUESTION NO: 6

There are a couple of different types of classification tasks in machine learning, Choose the Correct Classification which best categorized the below Application Tasks in Machine learning?

· To detect whether email is spam or not

· To determine whether or not a patient has a certain disease in medicine.

· To determine whether or not quality specifications were met when it comes to QA (Quality Assurance).

A. Multi-Label Classification

B. Multi-Class Classification

C. Binary Classification

D. Logistic Regression

ANSWER: C

Explanation:

Binary Classification is correct because every listed application requires choosing between exactly two mutually exclusive target classes. An email is classified as spam or not spam; a medical prediction assesses whether a specified disease is present or absent; and a quality-control decision determines whether a product meets specifications or does not meet them. In each case, the model uses labeled historical examples to learn a mapping from input features to one of two discrete outcomes.

Binary classification is a supervised-learning task. Its labels are commonly represented as values such as 0 and 1, or as meaningful categories such as “pass” and “fail.” A classifier can produce either a predicted class directly or a probability for one class, after which a decision threshold is applied. For instance, a spam model may score the probability that an email is spam and label it accordingly. The same fundamental setup applies to disease screening and quality assurance, although the appropriate decision threshold may differ depending on the business cost of false positives and false negatives.

Snowflake ML supports supervised classification workflows, including model training and prediction with data stored in Snowflake. For further background, see Snowflake's ML-based classification documentation and scikit-learn's LogisticRegression API documentation.

QUESTION NO: 7

What Can Snowflake Data Scientist do in the Snowflake Marketplace as Provider?

A. Publish listings for free-to-use datasets to generate interest and new opportunities among the Snowflake customer base.

B. Publish listings for datasets that can be customized for the consumer.

C. Share live datasets securely and in real-time without creating copies of the data or im-posing data integration tasks on the consumer.

D. Eliminate the costs of building and maintaining APIs and data pipelines to deliver data to customers.

ANSWER: A B C D

Explanation:

A Snowflake Marketplace provider can publish free-to-use listings to make data products discoverable across the Snowflake customer base. Free listings are a practical way to demonstrate the value of a data product, build awareness, and develop commercial opportunities without requiring an initial purchase. Providers can also support consumer-specific needs through listings and direct engagement, including offering data products or access arrangements tailored to a consumer's requirements.

Marketplace distribution is based on Snowflake Secure Data Sharing. This enables providers to share live, governed data with consumers without copying or moving the underlying data into the consumer's account. Because consumers access the shared data product directly through Snowflake, the provider does not need to build, operate, or maintain separate delivery APIs, file-export processes, or customer-specific data pipelines for this distribution model. The provider retains control of the shared object and can update the live dataset, with those updates becoming available to entitled consumers. Snowflake describes Marketplace as a channel for providers to publish and distribute data products, while Secure Data Sharing provides the underlying no-copy sharing capability. See Snowflake Marketplace provider documentation and Snowflake Secure Data Sharing documentation.

QUESTION NO: 8

Which Python method can be used to Remove duplicates by Data scientist?

A. remove_duplicates()

B. duplicates()

C. drop_duplicates()

D. clean_duplicates()

ANSWER: C

Explanation:

The correct method is `drop_duplicates()`. In pandas, `DataFrame.drop_duplicates()` returns a DataFrame with duplicate rows removed, making it a standard operation when a data scientist needs to prepare tabular data for analysis or modeling. By default, pandas compares all columns and retains the first occurrence of each duplicated row. The method can also be tailored to a data-quality requirement: use the `subset` parameter to identify duplicates based only on selected columns, use `keep` to control whether the first or last matching row is retained, or specify `keep=False` to remove every occurrence of duplicated values. For example, `df.drop_duplicates(subset=["customer_id"])` produces one row per customer identifier while preserving the first row encountered for each customer. The operation returns a new DataFrame unless it is explicitly applied in place. This behavior makes `drop_duplicates()` useful in reproducible data-

preparation workflows, where the original dataset may need to remain available for validation. The pandas API reference documents its parameters and return behavior at pandas.DataFrame.drop_duplicates. Additional examples of removing duplicate rows are available in the pandas indexing guide.