

## Download DSA-C02 Dumps (2024) - Free PDF Exam Demo [Q29-Q50]



## Download DSA-C02 Dumps (2024) - Free PDF Exam Demo [Q29-Q50]

### Download DSA-C02 Dumps (2024) - Free PDF Exam Demo Enhance your career with DSA-C02 PDF Dumps - True Snowflake Exam Questions NEW QUESTION 29

Which are the following additional Metadata columns Stream contains that could be used for creating Efficient Data science Pipelines & helps in transforming only the New/Modified data only?

- \* METADATA\$ACTION
- \* METADATA\$FILE\_ID
- \* METADATA\$ISUPDATE
- \* METADATA\$DELETE
- \* METADATA\$ROW\_ID

Explanation

A stream stores an offset for the source object and not any actual table columns or data. When que-ried, a stream accesses and returns the historic data in the same shape as the source object (i.e. the same column names and ordering) with the following additional columns:

METADATA\$ACTION

Indicates the DML operation (INSERT, DELETE) recorded.

#### METADATA\$ISUPDATE

Indicates whether the operation was part of an UPDATE statement. Updates to rows in the source object are represented as a pair of DELETE and INSERT records in the stream with a metadata column METADATA\$ISUPDATE values set to TRUE.

Note that streams record the differences between two offsets. If a row is added and then updated in the current offset, the delta change is a new row. The METADATA\$ISUPDATE row records a FALSE value.

#### METADATA\$ROW\_ID

Specifies the unique and immutable ID for the row, which can be used to track changes to specific rows over time.

### NEW QUESTION 30

Which is the visual depiction of data through the use of graphs, plots, and informational graphics?

- \* Data Mining
- \* Data Virtualization
- \* Data visualization
- \* Data Interpretation

Explanation

Data visualization is the visual depiction of data through the use of graphs, plots, and informational graphics.

Its practitioners use statistics and data science to convey the meaning behind data in ethical and accurate ways.

### NEW QUESTION 31

Which tool helps data scientist to manage ML lifecycle & Model versioning?

- \* MLFlow
- \* Pachyderm
- \* Albert
- \* CRUX

Explanation

Model versioning in a way involves tracking the changes made to an ML model that has been previously built.

Put differently, it is the process of making changes to the configurations of an ML Model. From another perspective, we can see model versioning as a feature that helps Machine Learning Engineers, Data Scientists, and related personnel create and keep multiple versions of the same model.

Think of it as a way of taking notes of the changes you make to the model through tweaking hyperparameters, retraining the model with more data, and so on.

In model versioning, a number of things need to be versioned, to help us keep track of important changes. I'll list and explain them below:

Implementation code: From the early days of model building to optimization stages, code or in this case source code of the model plays an important role. This code experiences significant changes during optimization stages which can easily be lost if not tracked

properly. Because of this, code is one of the things that are taken into consideration during the model versioning process.

**Data:** In some cases, training data does improve significantly from its initial state during model optimization phases. This can be as a result of engineering new features from existing ones to train our model on. Also there is metadata (data about your training data and model) to consider versioning. Metadata can change different times over without the training data actually changing. We need to be able to track these changes through versioning **Model:** The model is a product of the two previous entities and as stated in their explanations, an ML model changes at different points of the optimization phases through hyperparameter setting, model artifacts and learning coefficients. Versioning helps take record of the different versions of a Machine Learning model.

MLFlow & Pachyderm are the tools used to manage ML lifecycle & Model versioning.

## NEW QUESTION 32

Which type of Machine learning Data Scientist generally used for solving classification and regression problems?

- \* Supervised
- \* Unsupervised
- \* Reinforcement Learning
- \* Instructor Learning
- \* Regression Learning

Explanation

Supervised Learning

Overview:

Supervised learning is a type of machine learning that uses labeled data to train machine learning models. In labeled data, the output is already known. The model just needs to map the inputs to the respective outputs.

Algorithms:

Some of the most popularly used supervised learning algorithms are:

Linear Regression

Logistic Regression

Support Vector Machine

K Nearest Neighbor

Decision Tree

Random Forest

Naive Bayes

Working:

Supervised learning algorithms take labelled inputs and map them to the known outputs, which means you already know the target variable.

Supervised Learning methods need external supervision to train machine learning models. Hence, the name supervised. They need guidance and additional information to return the desired result.

Applications:

Supervised learning algorithms are generally used for solving classification and regression problems.

Few of the top supervised learning applications are weather prediction, sales forecasting, stock price analysis.

### NEW QUESTION 33

Which ones are the key actions in the data collection phase of Machine learning included?

- \* Label
- \* Ingest and Aggregate
- \* Probability
- \* Measure

Explanation

The key actions in the data collection phase include:

**Label:** Labeled data is the raw data that was processed by adding one or more meaningful tags so that a model can learn from it. It will take some work to label it if such information is missing (manually or automatically).

**Ingest and Aggregate:** Incorporating and combining data from many data sources is part of data collection in AI.

Data collection

Collecting data for training the ML model is the basic step in the machine learning pipeline. The predictions made by ML systems can only be as good as the data on which they have been trained. Following are some of the problems that can arise in data collection:

**Inaccurate data.** The collected data could be unrelated to the problem statement.

**Missing data.** Sub-data could be missing. That could take the form of empty values in columns or missing images for some class of prediction.

**Data imbalance.** Some classes or categories in the data may have a disproportionately high or low number of corresponding samples. As a result, they risk being under-represented in the model.

**Data bias.** Depending on how the data, subjects and labels themselves are chosen, the model could propagate inherent biases on gender, politics, age or region, for example. Data bias is difficult to detect and remove.

Several techniques can be applied to address those problems:

**Pre-cleaned, freely available datasets.** If the problem statement (for example, image classification, object recognition) aligns with a clean, pre-existing, properly formulated dataset, then take advantage of existing, open-source expertise.

**Web crawling and scraping.** Automated tools, bots and headless browsers can crawl and scrape websites for data.

Private data. ML engineers can create their own data. This is helpful when the amount of data required to train the model is small and the problem statement is too specific to generalize over an open-source dataset.

Custom data. Agencies can create or crowdsource the data for a fee.

### NEW QUESTION 34

Mark the incorrect statement regarding Python UDF?

- \* Python UDFs can contain both new code and calls to existing packages
- \* For each row passed to a UDF, the UDF returns either a scalar (i.e. single) value or, if defined as a table function, a set of rows.
- \* A UDF also gives you a way to encapsulate functionality so that you can call it repeatedly from multiple places in code
- \* A scalar function (UDF) returns a tabular value for each input row

Explanation

A scalar function (UDF) returns one output row for each input row. The returned row consists of a single column/value

### NEW QUESTION 35

Which object records data manipulation language (DML) changes made to tables, including inserts, updates, and deletes, as well as metadata about each change, so that actions can be taken using the changed data of Data Science Pipelines?

- \* Task
- \* Dynamic tables
- \* Stream
- \* Tags
- \* Delta
- \* OFFSET

Explanation

A stream object records data manipulation language (DML) changes made to tables, including inserts, updates, and deletes, as well as metadata about each change, so that actions can be taken using the changed data. This process is referred to as change data capture (CDC). An individual table stream tracks the changes made to rows in a source table. A table stream (also referred to as simply a `stream`;) makes a `change table`; available of what changed, at the row level, between two transactional points of time in a table. This allows querying and consuming a sequence of change records in a transactional fashion.

Streams can be created to query change data on the following objects:

Standard tables, including shared tables.

Views, including secure views

Directory tables

Event tables

### NEW QUESTION 36

The most widely used metrics and tools to assess a classification model are:

- \* Confusion matrix
- \* Cost-sensitive accuracy
- \* Area under the ROC curve

- \* All of the above

### NEW QUESTION 37

Consider a data frame `df` with 10 rows and index [ 'r1', 'r2', 'r3', 'row4', 'row5', 'row6', 'r7', 'r8', 'r9', 'row10' ].

What does the expression `g = df.groupby(df.index.str.len())` do?

- \* Groups `df` based on index values
- \* Groups `df` based on length of each index value
- \* Groups `df` based on index strings
- \* Data frames cannot be grouped by index values. Hence it results in Error.

Explanation

Data frames cannot be grouped by index values. Hence it results in Error.

### NEW QUESTION 38

Which one is not Types of Feature Scaling?

- \* Economy Scaling
- \* Min-Max Scaling
- \* Standard Scaling
- \* Robust Scaling

Explanation Feature Scaling

Feature Scaling is the process of transforming the features so that they have a similar scale. This is important in machine learning because the scale of the features can affect the performance of the model.

Types of Feature Scaling:

**Min-Max Scaling:** Rescaling the features to a specific range, such as between 0 and 1, by subtracting the minimum value and dividing by the range.

**Standard Scaling:** Rescaling the features to have a mean of 0 and a standard deviation of 1 by subtracting the mean and dividing by the standard deviation.

**Robust Scaling:** Rescaling the features to be robust to outliers by dividing them by the interquartile range.

Benefits of Feature Scaling:

**Improves Model Performance:** By transforming the features to have a similar scale, the model can learn from all features equally and avoid being dominated by a few large features.

**Increases Model Robustness:** By transforming the features to be robust to outliers, the model can become more robust to anomalies.

**Improves Computational Efficiency:** Many machine learning algorithms, such as k-nearest neighbors, are sensitive to the scale of the features and perform better with scaled features.

**Improves Model Interpretability:** By transforming the features to have a similar scale, it can be easier to understand the

model's predictions.

### NEW QUESTION 39

Which one is the incorrect option to share data in Snowflake?

- \* a Listing, in which you offer a share and additional metadata as a data product to one or more accounts.
- \* a Direct Marketplace, in which you directly share specific database objects (a share) to another account in your region using Snowflake Marketplace.
- \* a Direct Share, in which you directly share specific database objects (a share) to another account in your region.
- \* a Data Exchange, in which you set up and manage a group of accounts and offer a share to that group.

Explanation

Options for Sharing in Snowflake

You can share data in Snowflake using one of the following options:

- a Listing, in which you offer a share and additional metadata as a data product to one or more accounts,
- a Direct Share, in which you directly share specific database objects (a share) to another account in your region,
- a Data Exchange, in which you set up and manage a group of accounts and offer a share to that group.

### NEW QUESTION 40

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

- \* `remove_duplicates()`
- \* `duplicates()`
- \* `drop_duplicates()`
- \* `clean_duplicates()`

Explanation

The `drop_duplicates()` method removes duplicate rows.

```
dataframe.drop_duplicates(subset, keep, inplace, ignore_index)
```

Remove duplicate rows from the DataFrame:

1. `import pandas as pd`
2. `data = {`
3. `'name': ['Peter', 'Mary', 'John', 'Mary'],`
4. `'age': [50, 40, 30, 40],`
5. `'qualified': [True, False, False, False]`
6. `}`
- 7.

```
8.df = pd.DataFrame(data)
```

```
9.newdf = df.drop_duplicates()
```

## NEW QUESTION 41

Which one is not the types of Feature Engineering Transformation?

- \* Scaling
- \* Encoding
- \* Aggregation
- \* Normalization

Explanation

What is Feature Engineering?

Feature engineering is the process of transforming raw data into features that are suitable for machine learning models. In other words, it is the process of selecting, extracting, and transforming the most relevant features from the available data to build more accurate and efficient machine learning models.

The success of machine learning models heavily depends on the quality of the features used to train them.

Feature engineering involves a set of techniques that enable us to create new features by combining or transforming the existing ones. These techniques help to highlight the most important patterns and relationships in the data, which in turn helps the machine learning model to learn from the data more effectively.

What is a Feature?

In the context of machine learning, a feature (also known as a variable or attribute) is an individual measurable property or characteristic of a data point that is used as input for a machine learning algorithm. Features can be numerical, categorical, or text-based, and they represent different aspects of the data that are relevant to the problem at hand.

For example, in a dataset of housing prices, features could include the number of bedrooms, the square footage, the location, and the age of the property. In a dataset of customer demographics, features could include age, gender, income level, and occupation.

The choice and quality of features are critical in machine learning, as they can greatly impact the accuracy and performance of the model.

Why do we Engineer Features?

We engineer features to improve the performance of machine learning models by providing them with relevant and informative input data. Raw data may contain noise, irrelevant information, or missing values, which can lead to inaccurate or biased model predictions. By engineering features, we can extract meaningful information from the raw data, create new variables that capture important patterns and relationships, and transform the data into a more suitable format for machine learning algorithms.

Feature engineering can also help in addressing issues such as overfitting, underfitting, and high dimensionality. For example, by reducing the number of features, we can prevent the model from becoming too complex or overfitting to the training data. By selecting the most relevant features, we can improve the model's accuracy and interpretability.

In addition, feature engineering is a crucial step in preparing data for analysis and decision-making in various fields, such as finance,



healthcare, marketing, and social sciences. It can help uncover hidden insights, identify trends and patterns, and support data-driven decision-making.

We engineer features for various reasons, and some of the main reasons include:

**Improve User Experience:** The primary reason we engineer features is to enhance the user experience of a product or service. By adding new features, we can make the product more intuitive, efficient, and user-friendly, which can increase user satisfaction and engagement.

**Competitive Advantage:** Another reason we engineer features is to gain a competitive advantage in the marketplace. By offering unique and innovative features, we can differentiate our product from competitors and attract more customers.

**Meet Customer Needs:** We engineer features to meet the evolving needs of customers. By analyzing user feedback, market trends, and customer behavior, we can identify areas where new features could enhance the product's value and meet customer needs.

**Increase Revenue:** Features can also be engineered to generate more revenue. For example, a new feature that streamlines the checkout process can increase sales, or a feature that provides additional functionality could lead to more upsells or cross-sells.

**Future-Proofing:** Engineering features can also be done to future-proof a product or service. By anticipating future trends and potential customer needs, we can develop features that ensure the product remains relevant and useful in the long term.

#### Processes Involved in Feature Engineering

Feature engineering in Machine learning consists of mainly 5 processes: Feature Creation, Feature Transformation, Feature Extraction, Feature Selection, and Feature Scaling. It is an iterative process that requires experimentation and testing to find the best combination of features for a given problem. The success of a machine learning model largely depends on the quality of the features used in the model.

#### Feature Transformation

Feature Transformation is the process of transforming the features into a more suitable representation for the machine learning model. This is done to ensure that the model can effectively learn from the data.

#### Types of Feature Transformation:

**Normalization:** Rescaling the features to have a similar range, such as between 0 and 1, to prevent some features from dominating others.

**Scaling:** Rescaling the features to have a similar scale, such as having a standard deviation of 1, to make sure the model considers all features equally.

**Encoding:** Transforming categorical features into a numerical representation. Examples are one-hot encoding and label encoding.

**Transformation:** Transforming the features using mathematical operations to change the distribution or scale of the features. Examples are logarithmic, square root, and reciprocal transformations.

#### **NEW QUESTION 42**

Select the Correct Statements regarding Normalization?

- \* Normalization technique uses minimum and max values for scaling of model.
- \* Normalization technique uses mean and standard deviation for scaling of model.
- \* Scikit-Learn provides a transformer `MinMaxScaler` for Normalization.
- \* Normalization got affected by outliers.

Explanation

Normalization is a scaling technique in Machine Learning applied during data preparation to change the values of numeric columns in the dataset to use a common scale. It is not necessary for all datasets in a model. It is required only when features of machine learning models have different ranges.

Scikit-Learn provides a transformer called `MinMaxScaler` for Normalization.

This technique uses minimum and max values for scaling of model. It is useful when feature distribution is unknown. It got affected by outliers.

### NEW QUESTION 43

Data Scientist used streams in ELT (extract, load, transform) processes where new data inserted in-to a staging table is tracked by a stream. A set of SQL statements transform and insert the stream contents into a set of production tables. Raw data is coming in the JSON format, but for analysis he needs to transform it into relational columns in the production tables. which of the following Data transformation SQL function he can used to achieve the same?

- \* He could not apply Transformation on Stream table data.
- \* `lateral flatten()`
- \* `METADATA$ACTION ()`
- \* `Transpose()`

Explanation

To know about lateral flatten SQL Function, please refer:

<https://docs.snowflake.com/en/sql-reference/constructs/join-lateral#example-of-using-lateral-with-flatten>

### NEW QUESTION 44

Mark the correct steps for saving the contents of a DataFrame to a Snowflake table as part of Moving Data from Spark to Snowflake?

- \* Step 1. Use the `PUT()` method of the DataFrame to construct a `DataFrameWriter`.

Step 2. Specify `SNOWFLAKE_SOURCE_NAME` using the `NAME()` method.

Step 3. Use the `dbtable` option to specify the table to which data is written.

Step 4. Specify the connector options using either the `option()` or `options()` method.

Step 5. Use the `save()` method to specify the save mode for the content.

- \* Step 1. Use the `PUT()` method of the DataFrame to construct a `DataFrameWriter`.

Step 2. Specify `SNOWFLAKE_SOURCE_NAME` using the `format()` method.

Step 3. Specify the connector options using either the `option()` or `options()` method.

Step 4. Use the `dbtable` option to specify the table to which data is written.

Step 5. Use the `save()` method to specify the save mode for the content.

\* Step 1. Use the `write()` method of the `DataFrame` to construct a `DataFrameWriter`.

Step 2. Specify `SNOWFLAKE_SOURCE_NAME` using the `format()` method.

Step 3. Specify the connector options using either the `option()` or `options()` method.

Step 4. Use the `dbtable` option to specify the table to which data is written.

Step 5. Use the `mode()` method to specify the save mode for the content.

(Correct)

\* Step 1. Use the `writer()` method of the `DataFrame` to construct a `DataFrameWriter`.

Step 2. Specify `SNOWFLAKE_SOURCE_NAME` using the `format()` method.

Step 3. Use the `dbtable` option to specify the table to which data is written.

Step 4. Specify the connector options using either the `option()` or `options()` method.

Step 5. Use the `save()` method to specify the save mode for the content.

Explanation

Moving Data from Spark to Snowflake

The steps for saving the contents of a `DataFrame` to a Snowflake table are similar to writing from Snowflake to Spark:

1. Use the `write()` method of the `DataFrame` to construct a `DataFrameWriter`.
2. Specify `SNOWFLAKE_SOURCE_NAME` using the `format()` method.
3. Specify the connector options using either the `option()` or `options()` method.
4. Use the `dbtable` option to specify the table to which data is written.
5. Use the `mode()` method to specify the save mode for the content.

Examples

1. `df.write`

2. `format(SNOWFLAKE_SOURCE_NAME)`

3. `options(sfOptions)`

4. `option("dbtable", "t2")`

5. `mode(SaveMode.Overwrite)`

6..save()

### NEW QUESTION 45

Which type of Python UDFs let you define Python functions that receive batches of input rows as Pandas DataFrames and return batches of results as Pandas arrays or Series?

- \* MPP Python UDFs
- \* Scaler Python UDFs
- \* Vectorized Python UDFs
- \* Hybrid Python UDFs

Explanation

Vectorized Python UDFs let you define Python functions that receive batches of input rows as Pandas DataFrames and return batches of results as Pandas arrays or Series. You call vectorized Python UDFs the same way you call other Python UDFs.

Advantages of using vectorized Python UDFs compared to the default row-by-row processing pattern include:

The potential for better performance if your Python code operates efficiently on batches of rows.

Less transformation logic required if you are calling into libraries that operate on Pandas DataFrames or Pandas arrays.

When you use vectorized Python UDFs:

You do not need to change how you write queries using Python UDFs. All batching is handled by the UDF framework rather than your own code.

As with non-vectorized UDFs, there is no guarantee of which instances of your handler code will see which batches of input.

### NEW QUESTION 46

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

- \* Publish listings for free-to-use datasets to generate interest and new opportunities among the Snowflake customer base.
- \* Publish listings for datasets that can be customized for the consumer.
- \* Share live datasets securely and in real-time without creating copies of the data or imposing data integration tasks on the consumer.
- \* Eliminate the costs of building and maintaining APIs and data pipelines to deliver data to customers.

Explanation

All are correct!

About the Snowflake Marketplace

You can use the Snowflake Marketplace to discover and access third-party data and services, as well as market your own data products across the Snowflake Data Cloud.

As a data provider, you can use listings on the Snowflake Marketplace to share curated data offerings with many consumers simultaneously, rather than maintain sharing relationships with each individual consumer.

With Paid Listings, you can also charge for your data products.

As a consumer, you might use the data provided on the Snowflake Marketplace to explore and access the following:

Historical data for research, forecasting, and machine learning.

Up-to-date streaming data, such as current weather and traffic conditions.

Specialized identity data for understanding subscribers and audience targets.

New insights from unexpected sources of data.

The Snowflake Marketplace is available globally to all non-VPS Snowflake accounts hosted on Amazon Web Services, Google Cloud Platform, and Microsoft Azure, with the exception of Microsoft Azure Government.

Support for Microsoft Azure Government is planned.

#### NEW QUESTION 47

In a simple linear regression model (One independent variable), If we change the input variable by 1 unit. How much output variable will change?

- \* by 1
- \* no change
- \* by intercept
- \* by its slope

Explanation

What is linear regression?

Linear regression analysis is used to predict the value of a variable based on the value of another variable. The variable you want to predict is called the dependent variable. The variable you are using to predict the other variable's value is called the independent variable.

Linear regression attempts to model the relationship between two variables by fitting a linear equation to observed data. One variable is considered to be an explanatory variable, and the other is considered to be a dependent variable. For example, a modeler might want to relate the weights of individuals to their heights using a linear regression model.

A linear regression line has an equation of the form  $Y = a + bX$ , where  $X$  is the explanatory variable and  $Y$  is the dependent variable. The slope of the line is  $b$ , and  $a$  is the intercept (the value of  $y$  when  $x = 0$ ).

For linear regression  $Y = a + bx + \text{error}$ .

If neglect error then  $Y = a + bx$ . If  $x$  increases by 1, then  $Y = a + b(x+1)$  which implies  $Y = a + bx + b$ . So  $Y$  increases by its slope.

For linear regression  $Y = a + bx + \text{error}$ . If neglect error then  $Y = a + bx$ . If  $x$  increases by 1, then  $Y = a + b(x+1)$  which implies  $Y = a + bx + b$ . So  $Y$  increases by its slope.

#### NEW QUESTION 48

Mark the incorrect statement regarding usage of Snowflake Stream & Tasks?

- \* Snowflake automatically resizes and scales the compute resources for serverless tasks.
- \* Snowflake ensures only one instance of a task with a schedule (i.e. a standalone task or the root task in a DAG) is executed at a

given time. If a task is still running when the next scheduled execution time occurs, then that scheduled time is skipped.

- \* Streams support repeatable read isolation.
- \* An standard-only stream tracks row inserts only.

Explanation

All are correct except a standard-only stream tracks row inserts only.

A standard (i.e. delta) stream tracks all DML changes to the source object, including inserts, up-dates, and deletes (including table truncates).

#### **NEW QUESTION 49**

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

- \* Discover and test third-party data sources.
- \* Receive frictionless access to raw data products from vendors.
- \* Combine new datasets with your existing data in Snowflake to derive new business in-sights.
- \* Use the business intelligence (BI)/ML/Deep learning tools of her choice.

Explanation

As a consumer, you can do the following:

Discover and test third-party data sources.

Receive frictionless access to raw data products from vendors.

Combine new datasets with your existing data in Snowflake to derive new business insights.

Have datasets available instantly and updated continually for users.

Eliminate the costs of building and maintaining various APIs and data pipelines to load and up-date data.

Use the business intelligence (BI) tools of your choice.

#### **NEW QUESTION 50**

Data providers add Snowflake objects (databases, schemas, tables, secure views, etc.) to a share using Which of the following options?

- \* Grant privileges on objects to a share via Account role.
- \* Grant privileges on objects directly to a share.
- \* Grant privileges on objects to a share via a database role.
- \* Grant privileges on objects to a share via a third-party role.

Explanation What is a Share?

Shares are named Snowflake objects that encapsulate all of the information required to share a database.

Data providers add Snowflake objects (databases, schemas, tables, secure views, etc.) to a share using either or both of the following options:

Option 1: Grant privileges on objects to a share via a database role.

Option 2: Grant privileges on objects directly to a share.

You choose which accounts can consume data from the share by adding the accounts to the share.

After a database is created (in a consumer account) from a share, all the shared objects are accessible to users in the consumer account.

Shares are secure, configurable, and controlled completely by the provider account:

New objects added to a share become immediately available to all consumers, providing real-time access to shared data.

Access to a share (or any of the objects in a share) can be revoked at any time.

**100% Free DSA-C02 Files For passing the exam Quickly:** [https://www.dumpleader.com/DSA-C02\\_exam.html](https://www.dumpleader.com/DSA-C02_exam.html)