PassQueen
a.k.a dumpsqueen.com

Databricks Certified Professional Data Scientist Exam

Databricks Databricks-Certified-Professional-Data-Scientist

Version Demo

Total Demo Questions: 18

Total Premium Questions: 188

Buy Premium PDF

https://passqueen.com

support@passqueen.com

Topic Break Down

Topic	No. of Questions
Topic 1, Data Processing	3
Topic 2, Exploratory Data Analysis	35
Topic 3, Model Development	139
Topic 4, Machine Learning Operations	9
Topic 5, Mix Questions	2
Total	188

QUESTION NO: 1

You are training a model to predict next week's product demand. The dataset contains daily records ordered by date. Which validation approach should be used to obtain a realistic estimate of future performance?

A. Randomly assign daily records to training and validation datasets

B. Train on earlier dates and validate on later dates

C. Use the same records for both training and validation

D. Remove the date column and use a random split

ANSWER: B

Explanation:

A **time-based split** is correct because next week's demand must be predicted from information available before next week. Training on earlier dates and validating on later dates simulates the actual forecasting use case and avoids allowing future observations to influence the training process.

A random split can mix later records into the training data while earlier records appear in validation. This can lead to overly optimistic performance estimates, especially when demand is affected by trends, seasonality, promotions, or other time-dependent patterns. See the scikit-learn TimeSeriesSplit documentation.

QUESTION NO: 2

What are the advantages of the Hashing Features?

A. Requires the less memory

B. Less pass through the training data

C. Easily reverse engineer vectors to determine which original feature mapped to a vector location

ANSWER: A B

Explanation:

"Requires the less memory" and "Less pass through the training data" are advantages of feature hashing. Feature hashing converts features, such as categorical values or text tokens, directly into positions in a fixed-size numeric feature vector by applying a hash function. Because the vector dimensionality can be selected in advance, a pipeline does not need to build and retain a complete dictionary of every distinct feature value. Avoiding that vocabulary or mapping reduces memory requirements, especially for high-cardinality categorical data and large text corpora.

Feature hashing also allows transformation without an initial pass over the full training dataset to discover all distinct values and assign indices. This is useful in distributed and streaming-oriented workloads, as well as in situations where the possible vocabulary is unknown or unbounded. In Databricks machine-learning workflows, these properties can make hashing a practical preprocessing choice for sparse data at scale. The trade-off is that hashing is a one-way mapping and different inputs can collide in the same vector position, so it is less interpretable than dictionary-based indexing. Spark ML documents the hashing-based `FeatureHasher` transformer and its fixed-dimensional output behavior in its FeatureHasher documentation. For related scalable text-feature processing, see the Spark ML TF-IDF documentation.

QUESTION NO: 3

Suppose you have been given two Random Variables X and Y, whose joint distribution is already known, the marginal distribution of X is simply the probability distribution of X averaging over information about Y. It is the probability distribution of X when the value of Y is not known. So how do you calculate the marginal distribution of X

A. This is typically calculated by summing the joint probability distribution over Y.

B. This is typically calculated by integrating the joint probability distribution over Y

C. This is typically calculated by summing (In case of discrete variable) the joint probability distribution over Y

D. This is typically calculated by integrating(In case of continuous variable) the joint probability distribution over Y.

E. The marginal distribution of X can also be obtained by averaging the conditional distribution of X given Y over the distribution of Y.

ANSWER: A B C D E

Explanation:

The marginal distribution of X is obtained by eliminating, or marginalizing, Y from their joint distribution. For discrete variables, this means summing the joint probability mass function across every possible value of Y: $P(X = x) = \Sigma y\ P(X = x, Y = y)$. Thus, summing the joint distribution over Y and the more explicit discrete-variable formulation both correctly calculate the marginal probability mass function of X.

For continuous variables, the corresponding operation is integration: $fX(x) = \int fX,Y(x,y)\ dy$, with the integral taken across the support of Y. Therefore, integrating the joint probability density over Y and the explicitly continuous-variable formulation are both correct. The same principle can also be expressed through the law of total probability: the marginal probability or density of X is an average of the conditional distribution of X given Y, weighted by the probability distribution or density of Y. This equivalence explains why marginalization captures the distribution of X when Y is unobserved. See the marginal distribution definition and Penn State's treatment of marginal distributions.

QUESTION NO: 4

Select the correct statement regarding the naive Bayes classification

A. it only requires a small amount of training data to estimate the parameters

B. Independent variables can be assumed

C. only the variances of the variables for each class need to be determined

D. for each class entire covariance matrix need to be determined

ANSWER: A B C

Explanation:

Naive Bayes is a probabilistic classifier that applies Bayes' theorem while making a simplifying conditional-independence assumption for features given the class label. This simplification substantially reduces the number of parameters that must be learned. Rather than estimating the relationships among every pair of features through a full class-specific covariance matrix, the model estimates class priors and feature-level class-conditional parameters. For Gaussian naive Bayes, these feature-level parameters include an individual mean and variance for each feature within each class. Consequently, the model can often be trained effectively with relatively limited labeled data compared with classifiers that must estimate full covariance structures. This efficiency is especially useful for high-dimensional feature spaces, where estimating all covariance terms reliably would require much more data. Apache Spark's MLlib documentation describes Naive Bayes as a classifier based on conditional independence and supports several feature-distribution models, while its broader machine-learning documentation explains the scalable classification workflow used in Databricks environments. See the Apache Spark MLlib Naive Bayes documentation and the Spark ML classification documentation.

QUESTION NO: 5

Which of the following technique can be used to the design of recommender systems?

A. Naive Bayes classifier

B. Power iteration

C. Collaborative filtering

D. 1 and 3

E. 2 and 3

ANSWER: C

Explanation:

Collaborative filtering is a core technique for designing recommender systems because it produces recommendations from patterns in user-item interactions, such as ratings, clicks, purchases, or viewing history. Rather than requiring a detailed description of every product, film, song, or article, it identifies relationships in the interaction data: users with similar historical preferences may like similar items, and items liked by similar users may be relevant to the same audience. This makes the approach especially useful when item content is sparse, difficult to interpret automatically, or unavailable.

In practical implementations, collaborative filtering commonly uses a user-item matrix and learns latent factors that represent underlying preference patterns. Matrix-factorization approaches can then estimate missing user-item preferences and rank candidate items for each user. Apache Spark MLlib provides the Alternating Least Squares algorithm specifically for this collaborative-filtering use case, including support for explicit ratings and implicit-feedback data. Databricks workloads can use Spark-based feature engineering, model training, and scalable batch or streaming pipelines to operationalize these recommendation workflows. See the Apache Spark collaborative filtering documentation and the Databricks machine learning documentation for implementation context.

QUESTION NO: 6

Which of the following true with regards to the K-Means clustering algorithm?

A. Labels are not pre-assigned to each objects in the cluster.

B. Labels are pre-assigned to each objects in the cluster.

C. It classify the data based on the labels.

D. It discovers the center of each cluster.

E. It find each objects fall in which particular cluster

ANSWER: A D E

Explanation:

K-Means is an unsupervised clustering algorithm, so training examples do not require pre-assigned class labels. Instead, it uses the feature values of observations to identify groups whose members are similar according to a distance measure, commonly squared Euclidean distance. This makes “Labels are not pre-assigned to each objects in the cluster.” a correct description of its learning setup.

The algorithm represents every cluster with a centroid, which is the mean of the observations currently assigned to that cluster. It iteratively assigns each observation to its nearest centroid and recalculates the centroids from the resulting assignments. Therefore, “It discovers the center of each cluster.” correctly describes the role of K-Means centroids, and “It find each objects fall in which particular cluster” correctly reflects the assignment produced by the algorithm. The resulting cluster identifiers are inferred group memberships, rather than known target labels supplied as part of supervised training.

Databricks documents K-Means in MLlib as a clustering method that partitions a dataset into a specified number of clusters: Apache Spark MLlib K-Means documentation. For the underlying estimator behavior and parameters, see the PySpark KMeans API reference.

QUESTION NO: 7

Which of the following skills a data scientists required?

A. Web designing to represent best visuals of its results from algorithm.

B. He should be creative

C. Should possess good programming skills

D. Should be very good at mathematics and statistic

E. He should possess database administrative skills.

ANSWER: B C D

Explanation:

“He should be creative,” “Should possess good programming skills,” and “Should be very good at mathematics and statistic” are foundational skills for a data scientist. Creativity is important because data-science work is not limited to applying a predefined formula: practitioners frame ambiguous business problems, identify useful signals in available data, formulate hypotheses, and select practical ways to evaluate and communicate a solution. Programming skills are essential for preparing and exploring data, building reproducible feature-engineering and machine-learning workflows, training models, and operationalizing work in a collaborative environment. Databricks workflows commonly use Python, SQL, Scala, and libraries such as pandas and scikit-learn, while distributed processing can require Spark-based skills.

Mathematics and statistics provide the basis for choosing, training, and evaluating models responsibly. A data scientist needs statistical reasoning to understand sampling, distributions, uncertainty, hypothesis testing, bias, and appropriate evaluation measures; mathematical knowledge supports optimization and the behavior of many machine-learning algorithms. These capabilities work together: creative problem formulation guides what to build, programming implements the solution at scale, and mathematical and statistical knowledge validates whether the result is reliable and useful. Databricks describes the machine-learning lifecycle as including data preparation, model training, evaluation, and deployment, all of which draw on these competencies. See Databricks MLOps workflow documentation and Databricks model training documentation.

QUESTION NO: 8

Suppose A, B , and C are events. The probability of A given B , relative to P(|C), is the same as the probability of A given B and C (relative to P). That is,

A. P(A,B|C) / P(B|C) = P(A|B,C)

B. P(A,B|C) P(B|C) =P(B|A,C)

C. P(A,B|C) P(B|C) =P(C|B,C)

D. P(A,B|C) P(B|C) =P(A|C,B)

ANSWER: A

Explanation:

`P(A,B|C) / P(B|C) = P(A|B,C)` is the conditional-probability chain-rule identity, provided that the conditioning probabilities are nonzero. Conditioning on `C` creates a conditional probability space in which the probability of both `A` and `B` is `P(A,B|C)`. Within that space, the definition of conditional probability gives `P(A|B,C) = P(A,B|C) / P(B|C)`. Equivalently, multiplying both sides by `P(B|C)` yields the commonly used product form `P(A,B|C) = P(A|B,C)P(B|C)`. Thus, dividing the conditional joint probability by the conditional probability of `B` isolates the probability that `A` occurs once both `B` and `C` are known. This is a fundamental probability identity used in statistical modeling, including probabilistic graphical models and Bayesian calculations. The expression must use division, rather than multiplication, to match the stated conditional probability. See the conditional probability definition in the NIST/SEMATECH e-Handbook and the related conditional-probability reference from ProbabilityCourse.com.

QUESTION NO: 9

If E1 and E2 are two events, how do you represent the conditional probability given that E2 occurs given that E 1 has occurred?

A. P(E1)/P(E2)

B. P(E1+E2)/P(E1)

C. P(E2)/P(E1)

D. P(E2)/(P(E1+E2)

E. P(E1 ∩ E2)/P(E1)

ANSWER: E

Explanation:

The correct representation is **P(E2 | E1) = P(E1 ∩ E2) / P(E1)**, provided that P(E1) is greater than zero. Conditional probability restricts the sample space to outcomes in E1 because E1 is known to have occurred. Within that restricted sample space, the desired outcomes are those that also belong to E2. The numerator, P(E1 ∩ E2), measures the probability that both events occur together, while the denominator, P(E1), normalizes that joint probability relative to the condition that E1 has occurred. This is the standard definition used in probability theory and in data science when calculating probabilities after observing evidence or a feature value. The vertical bar in P(E2 | E1) is read as “the probability of E2 given E1.” The formula is also the basis for Bayes’ theorem and for many probabilistic modeling workflows. See the conditional-probability definition in the NIST/SEMATECH e-Handbook of Statistical Methods and the related treatment in OpenStax Introductory Statistics.

QUESTION NO: 10

Which of the following statement true with regards to Linear Regression Model?

A. Ordinary Least Squares can be used to estimate the parameters in a linear model. In a linear model, it finds a single linear function that approximates the relationship between the outcome and input variables.

B. Ordinary Least Square is a sum of the individual distance between each point and the fitted line of regression model.

C. Ordinary Least Square is a sum of the squared individual distance between each point and the fitted line of regression model.

ANSWER: A C

Explanation:

Ordinary Least Squares can be used to estimate the parameters in a linear model, and Ordinary Least Squares is the sum of the squared individual distances between each point and the fitted regression line. In linear regression, the model expresses the expected value of a target as a linear combination of input features and estimated coefficients, including an intercept where applicable. Ordinary Least Squares chooses the coefficient values that minimize the residual sum of squares: the total of each observed target value’s residual (observed value minus predicted value) squared. Squaring makes negative and positive residuals contribute positively, gives relatively greater weight to larger errors, and produces an optimization objective with well-established analytical and numerical solutions under the usual assumptions. For a simple linear regression model, this corresponds visually to choosing the single fitted line whose vertical residuals have the smallest possible total squared magnitude. With several predictors, the same principle estimates one linear prediction function, geometrically represented as a hyperplane rather than several independently fitted lines. Databricks AutoML supports regression experiments and evaluates regression models using standard regression metrics, while the underlying linear-regression concept remains parameter estimation by minimizing prediction error. See the Apache Spark LinearRegression documentation and the Databricks AutoML documentation.

QUESTION NO: 11

Which of the following are advantages of the Support Vector machines?

A. Effective in high dimensional spaces.

B. it is memory efficient

C. possible to specify custom kernels

D. Effective in cases where number of dimensions is greater than the number of samples

E. Number of features is much greater than the number of samples, the method still give good performances

F. SVMs directly provide probability estimates

ANSWER: A B C D

Explanation:

Support vector machines are well suited to several situations reflected in the correct choices. They can be effective in high-dimensional feature spaces because the learning objective is based on maximizing the separating margin, while the kernel approach can represent nonlinear decision boundaries without explicitly constructing every transformed feature. They can also remain effective when the number of dimensions exceeds the number of training examples, a setting that occurs frequently in sparse text and other feature-rich data sets. The fitted decision function depends on support vectors, which are the training observations lying on or within the learned margin. Retaining this subset for prediction can make the model comparatively memory efficient when only a limited portion of the training set becomes support vectors. Finally, custom kernels may be specified, allowing domain-specific similarity functions to be used when suitable, provided they meet the requirements of the SVM optimization method. Scikit-learn documents these properties as key strengths of support vector machines, including high-dimensional effectiveness, support-vector-based memory efficiency, and kernel versatility. See the scikit-learn Support Vector Machines guide and the scikit-learn kernel documentation.

QUESTION NO: 12

A bio-scientist is working on the analysis of the cancer cells. To identify whether the cell is cancerous or not, there has been hundreds of tests are done with small variations to say yes to the problem. Given the test result for a sample of healthy and cancerous cells, which of the following technique you will use to determine whether a cell is healthy?

A. Linear regression

B. Collaborative filtering

C. Naive Bayes

D. Identification Test

ANSWER: C

Explanation:

Naive Bayes is appropriate because this is a supervised binary-classification task: historical cell samples have known labels such as healthy or cancerous, and the model must predict one of those labels for a new cell from many test-result features. Naive Bayes estimates the probability of each class given the observed features, then selects the class with the highest posterior probability. It is especially well suited to a dataset whose inputs are categorical, binary, count-like, or otherwise representable as nonnegative independent feature values. For example, a large collection of yes/no test outcomes can be encoded as feature indicators and used to estimate how strongly each outcome is associated with healthy or cancerous samples. The method is computationally efficient and can provide a strong baseline for high-dimensional classification problems, particularly when there are many features relative to the number of labeled examples. In Databricks, Naive Bayes can be built with Apache Spark ML's `NaiveBayes` estimator, which supports multinomial and Bernoulli variants; Bernoulli Naive Bayes is relevant when features represent binary test outcomes. See the Spark ML NaiveBayes API documentation and the Spark ML classification guide.

QUESTION NO: 13

Select the correct statement which applies to Principal component analysis (PCA)

A. Is a mathematical procedure that transforms a number of (possibly) correlated variables into a (smaller) number of uncorrelated variables.

B. Is a mathematical procedure that transforms a number of (possibly) correlated variables into a (higher) number of uncorrelated variables

C. Increase the dimensionality of the data set.

D. 1 and 3 are correct

E. 1 and 2 are correct

ANSWER: A

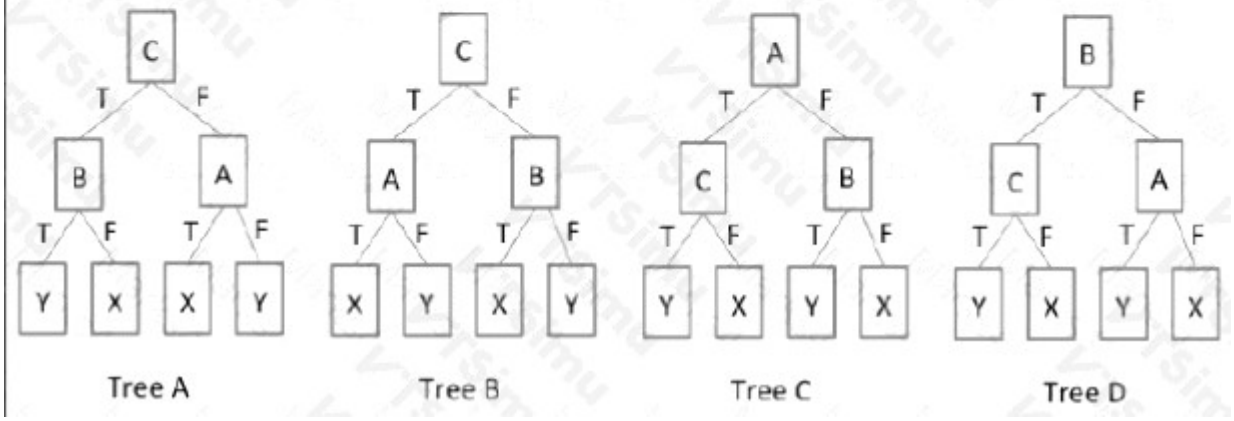
Explanation:

“Is a mathematical procedure that transforms a number of (possibly) correlated variables into a (smaller) number of uncorrelated variables.” is correct. PCA constructs new features, called principal components, as linear combinations of the original features. These components are mutually orthogonal; consequently, their scores are uncorrelated in the standard PCA setting. The components are ordered by how much variance they explain: the first captures the greatest possible variance, with each subsequent component capturing the greatest remaining variance subject to being orthogonal to those already selected. This ordering makes PCA especially useful for dimensionality reduction: a practitioner can retain only the leading components that preserve an appropriate amount of the dataset’s variation, rather than retaining every original feature. In Databricks, PCA is available through Spark ML and is commonly used in a pipeline after vectorizing and, when feature scales differ materially, scaling numerical inputs. The retained-component count is a modeling decision that balances information preservation, computational efficiency, and simpler downstream models. See the Spark ML PCA documentation at Apache Spark ML: PCA and the Databricks guidance on Spark ML pipelines.

QUESTION NO: 14

Refer to the Exhibit.

C	B	A	CLASS
T	T	T	X
T	T	F	Y
T	F	T	X
F	F	F	Y
F	T	T	X
F	F	T	Y



In the Exhibit, the table shows the values for the input Boolean attributes " A " , " B " , and " C " . It also shows the values for the output attribute " class " . Which decision tree is valid for the data?

A. Tree A

B. Tree B

C. Tree C

D. Tree D

ANSWER: B

Explanation:

Tree B is valid because a decision tree is valid for a labeled Boolean data set when every input row follows a sequence of tests to a leaf whose predicted class matches that row's class value. The tree's internal tests partition the examples according to the Boolean input attributes, and its leaf assignments are consistent with the class labels shown in the exhibit. In particular, an input can be evaluated from the root through exactly one branch at each Boolean split, so the resulting leaf provides a deterministic prediction for that combination of feature values. This is the essential behavior of a classification decision tree: recursively split the feature space using feature-based rules and assign a class prediction at a terminal node. A valid tree does not need to test every input attribute on every path; it needs only enough tests on each path to distinguish examples that require different class predictions. This representation also permits an attribute to be tested in different parts of the tree when appropriate for the subsets reaching those nodes. For background on decision-tree classifiers and their splitting-based classification behavior, see the scikit-learn decision tree documentation and the Apache Spark ML decision tree classifier documentation.

QUESTION NO: 15

You are creating a regression model with the input income, education and current debt of a customer, what could be the possible output from this model.

A. Customer fit as a good

B. Customer fit as acceptable or average category

C. expressed as a percent, that the customer will default on a loan

D. 1 and 3 are correct

E. 2 and 3 are correct

ANSWER: C

Explanation:

"expressed as a percent, that the customer will default on a loan" is a possible regression output because it represents a numeric value on a continuous scale, such as a value from 0 to 100 percent. A regression model learns the relationship between input features—including income, education, and current debt—and a numerical target. For a new customer, the model can produce an estimated likelihood of default, for example 18.4%, rather than assigning the customer only to a named category.

In practical machine-learning implementations, a loan-default probability is also frequently produced by a binary classification model that outputs calibrated probabilities. However, the key distinction intended here is that the stated result is numerical and can be represented as a continuous predicted value, which is consistent with regression-style output. Databricks describes regression as a supervised learning task for predicting continuous numerical values, and its machine-learning guidance includes evaluation approaches appropriate for numerical predictions. See the Databricks regression documentation and the Databricks classification documentation for the distinction between continuous predictions and class predictions.

QUESTION NO: 16

Which of the following are appropriate considerations when deploying a real-time classification model with Databricks Model Serving?

A. Register a model with an input schema for inference

B. Use traffic splitting to perform a controlled rollout of model versions

C. Restrict endpoint invocation to authorized users or service principals

D. Send the complete training dataset in every inference request

ANSWER: A B C

Explanation:

A registered model with a defined input schema helps establish a reliable inference contract for a serving endpoint. Endpoint traffic can be split between served model versions to support controlled rollout and evaluation. Authentication and access control should be applied so that only authorized clients invoke the endpoint.

Training data does not need to be sent with each prediction request. A serving endpoint receives the feature values needed for inference, while the trained model artifacts are loaded by the serving infrastructure. See the Databricks Model Serving documentation.

QUESTION NO: 17

Which of the following practices help prevent target leakage when training a machine learning model?

A. Fit feature transformations only on the training data

B. Use only features available when a prediction will be made

C. Split chronologically for a future prediction problem

D. Include the final outcome column as an input feature

E. Calculate normalization statistics using all train, validation, and test records

ANSWER: A B C

Explanation:

Creating features using only information available at the prediction time is required to prevent target leakage. For example, a loan-default model must not use a payment status that becomes available only after the loan decision has been made. Leakage can also occur when preprocessing is fitted using validation or test records. Therefore, transformations such as imputers, scalers, and encoders should be fitted on the training data and then applied to validation and test data.

For time-based prediction problems, splitting data chronologically helps ensure that the model is evaluated on future observations rather than information that would not have been available at training time. Randomly splitting data can produce optimistic estimates when records from the same future period influence training. See the scikit-learn data leakage guidance and Databricks model training documentation.

QUESTION NO: 18

You have collected the 100's of parameters about the 1000's of websites e.g. daily hits, average time on the websites, number of unique visitors, number of returning visitors etc. Now you have find the most important parameters which can best describe a website, so which of the following technique you will use

A. PCA (Principal component analysis)

B. Linear Regression

C. Logistic Regression

D. Clustering

ANSWER: A

Explanation:

PCA (Principal component analysis) is the appropriate technique because the task is to reduce a dataset containing many potentially correlated website metrics into a smaller set of informative dimensions. Rather than selecting one original metric at a time, PCA derives new features, called principal components, that are linear combinations of the original measurements. The first component captures the greatest possible variation in the data, and subsequent components capture the largest remaining variation while being orthogonal to earlier components. This makes it possible to represent each website using relatively few components while retaining much of the information present across metrics such as traffic, session duration, and visitor behavior.

In practice, the input features should generally be numerically encoded and scaled before applying PCA, particularly when their units or ranges differ substantially. The number of components can then be chosen based on explained variance or a modeling and interpretability requirement. In Spark ML, PCA is implemented as a transformer that projects feature vectors into a lower-dimensional representation; see the PySpark PCA API documentation. For the underlying estimator behavior and explained-variance output, see the scikit-learn PCA documentation.