PassQueen
a.k.a dumpsqueen.com

# CertNexus Certified Artificial Intelligence Practitioner (CAIP)

## CertNexus AIP-210

**Version Demo**

**Total Demo Questions: 11**

**Total Premium Questions: 112**

**Buy Premium PDF**

https://passqueen.com

support@passqueen.com

# Topic Break Down

| Topic | No. of Questions |
|---|---|
| Topic 1, Solving Business Problems Using AI and ML | 13 |
| Topic 2, Collecting and Preparing Data | 25 |
| Topic 3, Developing Machine Learning Models | 38 |
| Topic 4, Developing Deep Learning Models | 11 |
| Topic 5, Implementing AI Solutions | 11 |
| Topic 6, Maintaining AI Solutions | 14 |
| Total | 112 |

QUESTION NO: 1

Which of the following is NOT a valid cross-validation method?

**A.** Bootstrapping

**B.** K-fold

**C.** Leave-one-out

**D.** Stratification

**ANSWER: D**

**Explanation:**

Stratification is correct because it is a data-splitting principle, not a complete cross-validation procedure by itself. Stratification preserves the proportion of classes or target groups across generated subsets, which is particularly important for imbalanced classification datasets. It can be incorporated into a cross-validation method—for example, stratified k-fold cross-validation—but it does not specify the essential procedure of repeatedly defining training and validation partitions on its own. A valid cross-validation approach must describe how observations are partitioned and rotated or resampled so that model performance can be estimated on data not used for fitting. In practice, stratification augments that process by helping each fold better represent the overall class distribution, improving the reliability of evaluation metrics when some classes are uncommon. Scikit-learn documents stratified k-fold as a variation of k-fold that preserves class percentages in each fold, demonstrating that stratification is a property applied to a fold-generation strategy rather than an independent validation method. See the scikit-learn cross-validation guide and the StratifiedKFold API reference.

**QUESTION NO: 2**

You have a dataset with many features that you are using to classify a dependent variable. Because the sample size is small, you are worried about overfitting. Which algorithm is ideal to prevent overfitting?

**A.** Decision tree

**B.** Logistic regression

**C.** Random forest

**D.** XGBoost

**ANSWER: C**

**Explanation:**

Random forest is the best choice because it is an ensemble method designed to reduce the overfitting tendency of an individual decision tree. It trains many trees using bootstrap samples of the available training data and, at each split, considers only a randomly selected subset of features. These two sources of randomness make the individual trees less correlated. For classification, the forest combines their predictions by majority vote, which reduces prediction variance and typically generalizes more reliably than relying on a single highly flexible tree. This is especially valuable when a dataset has many candidate features relative to a small sample size, a situation in which a single model can easily learn idiosyncrasies of the training observations rather than stable patterns. Random forests also provide tunable controls, such as limiting tree depth, requiring a minimum number of samples at a leaf, and limiting the number of features evaluated at splits, so the model's complexity can be further constrained when data is scarce. The scikit-learn documentation explicitly describes random forests as fitting multiple decision-tree classifiers on subsamples and using averaging to improve predictive accuracy and control over-fitting. See the scikit-learn ensemble-method documentation and its RandomForestClassifier reference.

**QUESTION NO: 3**

A team is preparing a supervised learning model using customer records that include age, income, purchase history, and a final account-status field. The account-status field is assigned only after a customer has missed three payments, and the team is trying to predict whether a customer will miss three payments in the next 90 days.

Which two actions would help prevent target leakage? (Select two.)

**A.** Remove the final account-status field from the predictors.

**B.** Ensure that every predictor was available before the 90-day prediction point.

**C.** Apply one-hot encoding to all categorical predictors.

**D.** Standardize the continuous predictors.

**E.** Increase the size of the training data set.

**ANSWER: A B**

**Explanation:**

Removing the final account-status field from the predictors prevents the model from directly using information that is determined after the outcome of interest. Ensuring that each feature was available before the 90-day prediction point prevents the pipeline from including future information, such as payments or account changes that occurred after the prediction should have been made.

Scaling and one-hot encoding can be useful preprocessing steps, but they do not address leakage. Increasing the training set can improve estimation, but it can also increase the amount of leaked information if the feature definition is not corrected.

## QUESTION NO: 4

An AI system recommends New Year ' s resolutions. It has an ML pipeline without monitoring components. What retraining strategy would be BEST for this pipeline?

**A.** Periodically before New Year ' s Day and after New Year ' s Day

**B.** Periodically every year

**C.** When concept drift is detected

**D.** When data drift is detected

**ANSWER: A**

**Explanation:**

Periodically before New Year ' s Day and after New Year ' s Day is the best strategy because this recommendation use case is strongly seasonal and has a predictable annual demand cycle. People’s goals, interests, and engagement patterns are likely to change substantially as the New Year approaches and then again once resolutions are put into practice. Retraining shortly before the event allows the model to use the most recent pre-New-Year behavior and content signals when recommendations are most relevant. Retraining after the event incorporates the new interaction data generated during the high-activity period, helping the system remain useful as user behavior transitions from planning resolutions to pursuing them.

Because the pipeline has no monitoring components, it cannot reliably use automated data-drift or concept-drift detection to trigger retraining. A scheduled approach aligned to the known business and behavioral cycle is therefore appropriate. Google’s production ML guidance emphasizes that model behavior can change as data and real-world conditions change and that operational processes should account for such change: Google Rules of ML. AWS similarly describes scheduled retraining as a practical method for refreshing models when an anticipated cadence is known: Amazon SageMaker Pipeline triggering.

## QUESTION NO: 5

You create a prediction model with 96% accuracy. While the model ' s true positive rate (TPR) is performing well at 99%, the true negative rate (TNR) is only 50%. Your supervisor tells you that the TNR needs to be higher, even if it decreases the TPR. Upon further inspection, you notice that the vast majority of your data is truly positive.

What method could help address your issue?

**A.** Normalization

**B.** Oversampling

**C.** Principal components analysis

**D.** Quality filtering

**ANSWER: B**

**Explanation:**

Oversampling is appropriate because the data set is class-imbalanced: truly positive examples greatly outnumber truly negative examples. In this situation, a model can obtain an apparently strong overall accuracy and an excellent true positive rate primarily by favoring the majority positive class, while still failing to recognize a meaningful portion of negative cases. Oversampling increases the representation of the minority, truly negative class during training. This may be done by duplicating minority examples or, where appropriate, generating synthetic minority examples with techniques such as SMOTE. Giving the learner more negative-class observations helps it identify patterns associated with negative outcomes and can improve true negative rate (specificity). The expected tradeoff is consistent with the supervisor's requirement: changing the training distribution can make the decision boundary less biased toward positive predictions, potentially reducing true positive rate while improving detection of negatives. Oversampling should be applied only to the training data after splitting data into training and evaluation sets, so duplicated or synthetic observations do not leak into validation or test data. Performance should then be assessed with class-sensitive measures, including specificity, sensitivity, precision, recall, and a confusion matrix, rather than accuracy alone. See scikit-learn's imbalanced-classification guidance and Google's guidance on imbalanced datasets.

**QUESTION NO: 6**

Which two encodes can be used to transform categories data into numerical features? (Select two.)

**A.** Count Encoder

**B.** Log Encoder

**C.** Mean Encoder

**D.** Median Encoder

**E.** One-Hot Encoder

**ANSWER: C E**

**Explanation:**

Mean Encoder and One-Hot Encoder are both established approaches for converting categorical feature values into numeric values that machine-learning algorithms can consume. A Mean Encoder, often called target encoding, maps each category to a statistic derived from the target associated with that category, commonly its mean. This can represent useful category-to-outcome relationships compactly, particularly when a feature has many distinct values. It must be fit carefully—such as using cross-validation or out-of-fold calculations—to prevent target leakage into training features. Scikit-learn documents this form of supervised categorical transformation through its TargetEncoder reference.

One-Hot Encoder creates a separate binary indicator feature for each possible category. For every input record, the indicator corresponding to its observed category is set to one and the other category indicators are set to zero. This preserves category membership without incorrectly imposing a numeric ranking between nominal categories, making it a standard preprocessing method for many model types. The behavior and handling of categorical values are described in scikit-learn's OneHotEncoder documentation. Together, these methods represent two distinct, valid encoding strategies: target-statistic representation and binary indicator representation.

## QUESTION NO: 7

Which of the following statements are true regarding highly interpretable models? (Select two.)

**A.** They are usually binary classifiers.

**B.** They are usually easier to explain to business stakeholders.

**C.** They are usually referred to as " black box " models.

**D.** They are usually very good at solving non-linear problems.

**E.** They usually compromise on model accuracy for the sake of interpretability.

## ANSWER: B E

**Explanation:**

Highly interpretable models are designed so that people can directly understand how input features contribute to a prediction or decision. Therefore, *They are usually easier to explain to business stakeholders.* is correct. Models such as small decision trees and linear or logistic regression can communicate their logic in accessible forms: a tree expresses a sequence of decision rules, while a linear model exposes the direction and relative magnitude of each feature's relationship to the outcome. This transparency supports stakeholder review, trust, governance, and the ability to validate whether the model uses sensible factors.

*They usually compromise on model accuracy for the sake of interpretability.* is also correct as a general modeling trade-off. Interpretable model families often impose simpler functional forms or constrained structures, which may not represent complex interactions and nonlinear relationships as effectively as more flexible models. In practice, practitioners balance predictive performance against the need for transparency, auditability, and human oversight; the appropriate balance depends on the business context and consequences of model decisions. The NIST AI Risk Management Framework identifies explainability and interpretability as important characteristics for trustworthy AI, while Google's machine-learning guidance discusses the practical value of interpretable models. See NIST AI Risk Management Framework and Google Machine Learning Crash Course: Interpretability.

## QUESTION NO: 8

Which of the following methods can be used to rebalance a dataset using the rebalance design pattern?

**A.** Bagging

**B.** Boosting

**C.** Stacking

**D.** Weighted class

## ANSWER: D

**Explanation:**

Weighted class is a dataset-rebalancing method that changes how strongly a learning algorithm penalizes errors from each target class. Rather than altering the number of records in the training set, class weights assign greater importance to underrepresented classes and lower importance to classes with many examples. A common approach is to make each class's weight inversely related to its frequency. Consequently, a misclassification involving a rare class contributes more to the training loss, encouraging the model to learn decision boundaries that better recognize minority-class examples.

This approach is especially useful when duplicating minority examples or removing majority examples would be undesirable, such as when the available training data is limited or retaining the original data distribution is important for governance and traceability. Many machine-learning libraries directly support class weighting during model training. For example, scikit-learn documents balanced class weights as weights inversely proportional to class frequencies, and TensorFlow supports supplying class weights to emphasize classes during fitting. These capabilities implement the core objective of the rebalance

design pattern: reducing majority-class dominance so that the model can learn appropriately from all relevant classes. See the scikit-learn class-weight documentation and TensorFlow Model.fit documentation.

## QUESTION NO: 9

A team is building a model to predict equipment failures. Only 2% of historical records represent actual failures. The team wants evaluation results that show both whether predicted failures are credible and whether actual failures are being found.

Which two measures should the team use? (Select two.)

**A.** Accuracy

**B.** Mean absolute error

**C.** Precision

**D.** R-squared

**E.** Recall

## ANSWER: C E

**Explanation:**

**Precision** measures how often predicted failures are true failures, which shows whether maintenance alerts are credible. **Recall** measures how many actual failures are identified, which shows whether the model is missing important failure events.

Overall accuracy is not sufficient for this imbalanced data set because predicting no failure for every record could still yield approximately 98% accuracy. Mean absolute error and R-squared are regression measures and are not appropriate primary measures for this binary classification task.

## QUESTION NO: 10

Which two encoders can be used to transform categorical data into numerical features? (Select two.)

**A.** Count Encoder

**B.** Log Encoder

**C.** Mean Encoder

**D.** Median Encoder

**E.** One-Hot Encoder

## ANSWER: A E

**Explanation:**

Count Encoder and One-Hot Encoder are both established methods for converting categorical variables into numeric features suitable for machine-learning algorithms. A Count Encoder replaces each category with the number of times that category occurs in the relevant dataset. This produces one numeric feature per encoded categorical column and can be useful when category frequency itself may carry meaningful signal. Feature-engine documents this approach as count/frequency encoding for categorical variables: Feature-engine CountFrequencyEncoder documentation.

One-Hot Encoder transforms a categorical feature into a set of binary indicator columns, with each resulting column representing one possible category. Each observation receives numeric values indicating whether it belongs to each category, allowing models to use nominal categories without imposing an artificial rank or magnitude among them. It is a standard preprocessing transformation and is directly supported by scikit-learn's OneHotEncoder documentation. Both

methods therefore meet the requirement of transforming categorical data into numerical features, while representing the category information in different numeric forms.

## QUESTION NO: 11

A product manager is designing an Artificial Intelligence (AI) solution and wants to do so responsibly, evaluating both positive and negative outcomes.

The team creates a shared taxonomy of potential negative impacts and conducts an assessment along vectors such as severity, impact, frequency, and likelihood.

Which modeling technique does this team use?

**A.** Business

**B.** Harms modeling

**C.** Process

**D.** Threat

## ANSWER: B

**Explanation:**

**Harms modeling** is the responsible-AI technique described because it provides a structured, collaborative way to anticipate how an AI-enabled product could create negative outcomes for people, organizations, or society. The team first establishes a common taxonomy and vocabulary for harms, ensuring that product, engineering, design, policy, and other stakeholders can identify risks consistently. It then assesses the identified harms using dimensions such as severity, scale or impact, frequency, and likelihood. This supports prioritization: harms that are both serious and plausible can be addressed early through product requirements, safeguards, testing, monitoring, or changes to the intended use of the system.

Importantly, harms modeling is performed during design rather than only after deployment. Considering positive and negative outcomes together helps teams make informed trade-offs and incorporate risk mitigation throughout the AI solution lifecycle. Google describes harms modeling as a framework for identifying potential harms and incorporating responsible product decisions, while Microsoft documents a similar process of identifying, measuring, and mitigating potential AI harms. See Google Developers: Harms Modeling and Microsoft Learn: Harms modeling.