Ml 1 assignment

**Question 1. Defining Artificial Intelligence (AI) ?**

**Artificial Intelligence (AI)** is a branch of computer science that aims to

create intelligent agents, which are systems that can reason, learn, and

act autonomously. In simpler terms, AI involves developing computer

systems that can perform tasks that typically require human intelligence,

such as understanding natural language, recognizing patterns, and

making decisions.

**Question 2 Differences Between AI, Machine Learning (ML), Deep Learning (DL), and Data Science (DS)**

* **AI:** The broader field encompassing systems that can perform tasks that require human intelligence.
* **ML:** A subset of AI that focuses on algorithms that allow computers to learn from data without being explicitly programmed.
* **DL:** A subset of ML that uses artificial neural networks with multiple layers to learn complex patterns from large datasets.
* **DS:** A field that uses statistical and computational techniques to extract insights from data.

**Ques 3 How AI Differs from Traditional Software Development ?**

While traditional software development follows a structured approach

with predefined rules, AI involves creating systems that can adapt and

learn. AI systems can handle uncertainty, ambiguity, and new data,

making them more flexible and responsive to changing environments.

**Ques 4 Examples of AI, ML, DL, and DS Applications**

* **AI:** Virtual assistants (like Siri, Alexa), recommendation systems, autonomous vehicles
* **ML:** Spam filters, fraud detection, customer segmentation
* **DL:** Image recognition, natural language processing, speech recognition
* **DS:** Market analysis, customer behavior analysis, healthcare analytics

**Importance of AI, ML, DL, and DS in Today's World**

These technologies are driving innovation in various industries, including healthcare, finance, manufacturing, and transportation. They can improve efficiency, enhance decision-making, and create new products and services.

**Ques 5 What is Supervised Learning?**

**Supervised learning** is a type of machine learning where the algorithm is

trained on a dataset with labeled inputs and outputs. The goal is to learn

a mapping function that can predict outputs for new, unseen inputs.

**Ques 6 Examples of Supervised Learning Algorithms**

* **Regression:** Linear regression, logistic regression
* **Classification:** Decision trees, random forests, support vector machines
* **Neural networks:** Multilayer perceptrons

**Ques 7The Process of Supervised Learning**

1. **Data preparation:** Collect and preprocess data.
2. **Model selection:** Choose an appropriate algorithm.
3. **Training:** Train the model on the labeled data.
4. **Evaluation:** Assess the model's performance on a validation set.
5. **Prediction:** Use the trained model to make predictions on new data.

**Ques 8 Characteristics of Unsupervised Learning**

**Unsupervised learning** involves training models on data without labeled

outputs. The goal is to discover patterns, structures, or relationships

within the data itself.

**Ques 9 Examples of Unsupervised Learning Algorithms**

* **Clustering:** K-means, hierarchical clustering
* **Dimensionality reduction:** Principal component analysis (PCA)
* **Association rules:** Apriori algorithm

**Ques 10 Unsupervised Learning Algorithms**

* **Clustering:**
  + **K-means:** Groups data points into k clusters based on their similarity.
  + **Hierarchical clustering:** Creates a hierarchy of clusters, starting from individual data points and merging them into larger clusters.
* **Dimensionality reduction:**
  + **Principal Component Analysis (PCA):** Reduces the dimensionality of data while preserving the most important information.
  + **t-SNE:** Visualizes high-dimensional data in a lower-dimensional space.
* **Association rules:**
  + **Apriori algorithm:** Finds frequent itemsets and association rules between items in a dataset.

**Ques 11 Semi-Supervised Learning**

**Semi-supervised learning** is a type of machine learning where the algorithm is trained on a dataset with both labeled and unlabeled data. This approach is useful when labeling data is expensive or time-consuming. The algorithm can learn from the labeled data and then use the unlabeled data to improve its performance.

**Significance:**

* **Leverages limited labeled data:** Can be effective when labeled data is scarce.
* **Improves model performance:** Can enhance accuracy compared to supervised learning with limited labeled data.
* **Real-world applications:** Widely used in tasks like image classification, natural language processing, and recommendation systems.

**Ques 12 Reinforcement Learning**

**Reinforcement learning** is a type of machine learning where an agent learns to interact with an environment to maximize a reward signal. The agent takes actions and receives feedback in the form of rewards or penalties. Over time, the agent learns the optimal policy to achieve its goals.

**Applications:**

* **Game playing:** AlphaGo, AlphaZero
* **Robotics:** Autonomous navigation, manipulation
* **Finance:** Algorithmic trading
* **Healthcare:** Personalized treatment plans

**Ques 13 Differences Between Supervised, Unsupervised, and Reinforcement Learning**

**Ques 14 Purpose of Train-Test-Validation Split**

The train-test-validation split is a common technique in machine learning to evaluate the performance of a model. It involves dividing the dataset into three parts:

* **Training set:** Used to train the model.
* **Validation set:** Used to tune hyperparameters and select the best model.
* **Testing set:** Used to evaluate the final performance of the selected model on unseen data.

**Significance of the Training Set**

The training set is crucial for teaching the model the underlying patterns and relationships in the data. A larger training set can typically lead to better model performance, but there are diminishing returns beyond a certain point.

**Ques 15 Determining the Size of Training, Testing, and Validation Sets**

The optimal split ratios can vary depending on the specific problem and dataset. However, common guidelines include:

* **Training set:** 60-80%
* **Validation set:** 10-20%
* **Testing set:** 10-20%

**Ques 16 Consequences of Improper Train-Test-Validation Splits**

* **Overfitting:** If the training set is too small or the validation set is too large, the model may become overly specialized to the training data and perform poorly on new data.
* **Underfitting:** If the training set is too large or the validation set is too small, the model may not learn the underlying patterns well enough and perform poorly on both training and testing data.

**Ques 17 Trade-offs in Selecting Appropriate Split Ratios**

* **Larger training set:** Can improve model performance but may lead to overfitting if the validation set is too small.
* **Larger validation set:** Can help prevent overfitting but may reduce the amount of data available for training.
* **Larger testing set:** Provides a more reliable evaluation of the model's performance but may limit the amount of data available for training and validation.

**Ques 18 Model Performance in Machine Learning**

**Model performance** refers to how well a machine learning model performs on its intended task. It is typically measured using metrics that quantify the model's accuracy, precision, recall, F1-score, or other relevant metrics.

**Ques 19 Measuring Model Performance**

* **Regression:** Mean squared error (MSE), root mean squared error (RMSE), mean absolute error (MAE)
* **Classification:** Accuracy, precision, recall, F1-score, confusion matrix
* **Clustering:** Silhouette coefficient, Calinski-Harabasz index
* **Ranking:** Normalized discounted cumulative gain (NDCG), mean reciprocal rank (MRR)

**Ques 21 Overfitting and Underfitting**

**Overfitting**

Overfitting occurs when a model becomes overly complex and learns the training data too well, to the point where it performs poorly on new, unseen data. This is often due to the model memorizing the training data instead of learning the underlying patterns.

**Why it's problematic:**

* **Poor generalization:** The model fails to generalize to new data.
* **Reduced performance:** Lower accuracy on unseen data.
* **Overly complex model:** Difficult to interpret and deploy.

**Ques 22 Techniques to Address Overfitting**

* **Regularization:** Penalizes complex models to prevent overfitting (e.g., L1, L2 regularization).
* **Early stopping:** Stop training when performance on the validation set starts to deteriorate.
* **Cross-validation:** Evaluate the model's performance on multiple subsets of the data to avoid overfitting to a particular split.
* **Feature selection:** Remove irrelevant or redundant features to simplify the model.
* **Ensemble methods:** Combine multiple models to reduce overfitting and improve generalization (e.g., bagging, boosting).

**Ques 23 Underfitting**

Underfitting occurs when a model is too simple to capture the underlying patterns in the data. This leads to poor performance on both the training and testing sets.

**Implications:**

* **High bias:** The model is biased towards simple models and cannot capture complex relationships.
* **Poor performance:** Low accuracy on both training and testing data.
* **Inaccurate predictions:** The model cannot make accurate predictions.

**Ques 24 Preventing Underfitting**

* **Increase model complexity:** Add more layers, neurons, or features to the model.
* **Gather more data:** Increase the size of the training set to provide more information.
* **Feature engineering:** Create new features that may be more informative.
* **Try different algorithms:** Experiment with different algorithms that are better suited for the problem.

**Ques 25 Bias-Variance Trade-off**

The bias-variance trade-off is a fundamental concept in machine learning. Bias refers to the model's systematic error due to underfitting, while variance refers to the model's sensitivity to small changes in the training data.

* **High bias, low variance:** The model is underfitted and performs consistently poorly.
* **Low bias, high variance:** The model is overfitted and performs well on the training data but poorly on new data.

The goal is to find a balance between bias and variance to achieve optimal performance.

**Ques 26 Handling Missing Data**

**Common Techniques**

* **Deletion:** Remove rows or columns with missing values.
* **Imputation:** Replace missing values with estimated values (e.g., mean, median, mode, imputation algorithms).
* **Ignoring missing values:** If missing values are few and random, they may be ignored.

**Ques 27 Implications of Ignoring Missing Data**

* **Biased results:** If missing values are not randomly distributed, ignoring them can lead to biased results.
* **Reduced model performance:** Missing data can reduce the model's ability to learn patterns and make accurate predictions.

**Ques 28 Pros and Cons of Imputation Methods**

* **Mean/median/mode imputation:** Simple but can introduce bias if the distribution is skewed.
* **Hot-deck imputation:** Replaces missing values with values from similar observations.
* **Regression imputation:** Predicts missing values using regression models.
* **Multiple imputation:** Creates multiple imputed datasets and combines the results to reduce bias.

**Ques 29 Impact of Missing Data on Model Performance**

Missing data can significantly affect model performance. It can introduce bias, reduce accuracy, and make it difficult to interpret the results. The choice of imputation method and the extent of missing data will influence the impact on model performance.

**Imbalanced Data**

**Imbalanced data** occurs when the classes in a dataset are not equally represented. This can lead to biased models that favor the majority class.

**Examples:**

* **Fraud detection:** Few fraudulent transactions compared to legitimate ones.
* **Medical diagnosis:** Rare diseases compared to common ones.

**Ques 30 Addressing imbalanced data:**

* **Oversampling:** Increase the number of samples from the minority class.
* **Undersampling:** Reduce the number of samples from the majority class.
* **SMOTE (Synthetic Minority Over-sampling Technique):** Generates new synthetic samples for the minority class.
* **Class weighting:** Assign higher weights to samples from the minority class during training.

**Ques 31 Challenges Posed by Imbalanced Data**

* **Biased models:** Models trained on imbalanced data may be biased towards the majority class, leading to poor performance on the minority class.
* **Underfitting:** Models may fail to learn the patterns in the minority class due to limited data.
* **Overfitting:** Models may overfit to the majority class, leading to poor generalization.

**Ques 32 Techniques to Address Imbalanced Data**

* **Oversampling:**
  + Increase the number of samples from the minority class.
  + Random oversampling: Randomly duplicates samples from the minority class.
  + SMOTE (Synthetic Minority Over-sampling Technique): Generates new synthetic samples for the minority class.
* **Undersampling:**
  + Reduce the number of samples from the majority class.
  + Random undersampling: Randomly removes samples from the majority class.
  + Cluster-based undersampling: Clusters the majority class and removes samples from dense clusters.
* **Class weighting:**
  + Assign higher weights to samples from the minority class during training.

**Ques 33 Upsampling and Downsampling**

* **Upsampling:** Increases the number of samples in the minority class by creating duplicates or generating new samples.
* **Downsampling:** Reduces the number of samples in the majority class by randomly removing samples.

**Ques 34 When to Use Upsampling vs. Downsampling**

* **Upsampling:** When the minority class has very few samples or when the data is high-dimensional.
* **Downsampling:** When the majority class has a large number of samples and the data is low-dimensional.

**Ques 35 SMOTE (Synthetic Minority Over-sampling Technique)**

SMOTE generates new synthetic samples for the minority class by interpolating between existing minority class samples and their nearest neighbors. This helps to address the problem of class imbalance by creating more diverse samples.

**Ques 36 Role of SMOTE in Handling Imbalanced Data**

* **Increases minority class representation:** SMOTE helps to balance the class distribution.
* **Creates synthetic samples:** Generates new, diverse samples for the minority class.
* **Improves model performance:** Can improve the model's ability to classify the minority class.

**Ques 37 Advantages and Limitations of SMOTE**

**Advantages:**

* Effective for handling imbalanced data.
* Can improve model performance on the minority class.
* Can create more diverse samples.

**Limitations:**

* May introduce noise or bias if the minority class is highly clustered.
* Can be computationally expensive for large datasets.

**Ques 38 Scenarios Where SMOTE is Beneficial**

* Medical diagnosis with rare diseases.
* Fraud detection with few fraudulent transactions.
* Customer churn prediction with low churn rates.
* virtual layer that provides a unified view of data from multiple sources.

**Data Interpolation**

**Ques 39 Data interpolation** is the process of estimating missing or unknown values in a dataset based on the known values. It's often used to create a complete dataset for analysis or modeling.

**Purpose:**

* **Fill missing values:** Complete datasets for analysis.
* **Create new data points:** Generate data between existing points.
* **Smooth data:** Reduce noise and irregularities.

**Ques 40 Common Methods of Data Interpolation**

* **Linear interpolation:** Assumes a linear relationship between data points and estimates missing values by drawing a straight line between the nearest known points.
* **Polynomial interpolation:** Uses a polynomial function to fit the data points and estimate missing values.
* **Spline interpolation:** Uses piecewise polynomial functions to fit the data points, ensuring smoothness between segments.
* **Nearest neighbor interpolation:** Assigns the value of the nearest known data point to the missing value.
* **Kriging interpolation:** A geostatistical method that considers the spatial correlation between data points.

**Choice of method depends on:**

* **Nature of data:** Continuous or categorical.
* **Expected relationships between data points:** Linear, polynomial, or other.
* **Desired level of smoothness:** How smooth the interpolated data should be.

**Ques 41 Implications of Using Data Integration in Machine Learning**

* **Improved data quality:** Integrated data can be cleaned and standardized, leading to better model performance.
* **Increased dataset size:** Combining data from multiple sources can increase the amount of data available for training, potentially improving model accuracy.
* **Enhanced insights:** Integrated data can provide a more comprehensive view of the problem, leading to deeper insights and better decision-making.
* **Increased complexity:** Integrating data from multiple sources can increase the complexity of the data and require more sophisticated data processing techniques.

**Ques 42 outliers in a Dataset**

**Outliers** are data points that significantly deviate from the majority of the data. They can be caused by errors, anomalies, or rare events.

**Ques 43 Impact of Outliers on Machine Learning Models**

* **Biased models:** Outliers can skew the model's training, leading to biased predictions.
* **Reduced accuracy:** Outliers can reduce the model's accuracy, especially for sensitive algorithms.
* **Overfitting:** Outliers can cause the model to overfit to the training data, leading to poor generalization.

**Ques 44 Techniques for Identifying Outliers**

* **Statistical methods:** Z-score, IQR (Interquartile Range), Grubbs' test, Dixon's Q-test
* **Visualization:** Box plots, scatter plots, histograms
* **Machine learning methods:** Isolation Forest, One-Class SVM

**Ques 45 Handling Outliers in a Dataset**

* **Removal:** Remove outliers if they are clearly erroneous or have a significant impact on the model.
* **Capping:** Replace outliers with extreme values (e.g., maximum or minimum).
* **Transformation:** Apply transformations (e.g., log transformation) to reduce the impact of outliers.
* **Robust algorithms:** Use algorithms that are less sensitive to outliers (e.g., robust regression).

**Ques 46 Feature Selection Methods**

* **Filter methods:** Rank features based on their individual relevance to the target variable.
* **Wrapper methods:** Evaluate different combinations of features using a machine learning model.
* **Embedded methods:** Feature selection is integrated into the model training process.

**Ques 47 Examples of Algorithms**

* **Filter methods:** Chi-square test, ANOVA, correlation
* **Wrapper methods:** Forward selection, backward elimination, recursive feature elimination
* **Embedded methods:** L1 regularization (Lasso), L2 regularization (Ridge), decision trees

**Ques 48 Advantages and Disadvantages of Feature Selection Methods**

**Ques 49 Feature Scaling**

**Feature scaling** is the process of standardizing the range of features in a dataset to improve the performance of machine learning algorithms.

**Ques 50 Standardization**

**Standardization** transforms features to have a mean of 0 and a standard deviation of 1. It's commonly used when features have different scales or distributions.

**Formula:**

z = (x - mean) / standard deviation

**Advantages:**

* Improves convergence of gradient descent algorithms.
* Makes features comparable on a common scale.
* Helps prevent domination by features with large magnitudes.

**Disadvantages:**

* May not be suitable for certain algorithms (e.g., decision trees) that are invariant to scaling.

**Ques 51 Difference Between Standardization and Min-Max Scaling**

* **Standardization:** Scales data to have a mean of 0 and a standard deviation of 1, assuming a normal distribution.
* **Min-Max scaling:** Scales data to a specific range (e.g., 0 to 1) by subtracting the minimum value and dividing by the range.

**Ques 52 Advantages and Disadvantages of Min-Max Scaling**

**Advantages:**

* Preserves the original distribution of the data.
* Can be useful for algorithms that require features to be in a specific range.

**Disadvantages:**

* Sensitive to outliers, which can distort the scale.
* May not be suitable for algorithms that assume a normal distribution.

**Ques 53 Purpose of Unit Vector Scaling**

Unit vector scaling scales features to have a magnitude of 1. This is often used when features represent vectors or directions, such as in natural language processing or image analysis.

**Ques 54 Principal Component Analysis (PCA)**

PCA is a dimensionality reduction technique that transforms a high-dimensional dataset into a lower-dimensional dataset while preserving the most important information. It finds a new set of features (principal components) that are uncorrelated and capture the maximum variance in the data.

**Ques 55 Steps Involved in PCA**

1. **Center the data:** Subtract the mean from each feature.
2. **Calculate the covariance matrix:** Compute the covariance between pairs of features.
3. **Compute the eigenvectors and eigenvalues of the covariance matrix.**
4. **Select the principal components:** Choose the eigenvectors corresponding to the largest eigenvalues.
5. **Transform the data:** Project the original data onto the selected principal components.

**Ques 56 Significance of Eigenvalues and Eigenvectors**

* **Eigenvalues:** Represent the variance explained by each principal component. Larger eigenvalues correspond to more important components.
* **Eigenvectors:** Define the direction of the principal components.

**Ques 57 PCA for Dimensionality Reduction**

PCA can significantly reduce the dimensionality of a dataset while preserving the most important information. This can improve model performance, reduce computational cost, and make the data easier to visualize.

**Ques 58 Data Encoding**

**Data encoding** is the process of converting categorical data into a numerical format that can be used by machine learning algorithms. This is necessary because most machine learning algorithms require numerical input.

**Ques 59 Nominal Encoding**

**Nominal encoding** assigns a unique integer to each category in a nominal variable. This is suitable for variables with no inherent order.

**Example:**

* Colour: Red (1), Blue (2), Green (3)

**Ques 60 One-Hot Encoding**

**One-hot encoding** creates a new binary feature for each category in a nominal variable. Each category is represented by a binary vector with a 1 in the corresponding position and 0s elsewhere.

**Example:**

* Colour: Red (1, 0, 0), Blue (0, 1, 0), Green (0, 0, 1)

**Ques 61 Handling Unique Categories in One-Hot Encoding**

When dealing with unique categories in One-Hot Encoding, you can:

* **Create a separate feature:** If the unique category appears frequently, create a separate binary feature for it.
* **Combine with similar categories:** If the unique category is similar to other categories, combine them into a single category.
* **Ignore unique categories:** If the unique category is rare and doesn't significantly affect the target variable, you can ignore it.

**Ques 62 Mean Encoding**

**Mean Encoding** is a technique where categorical variables are replaced with the mean of the target variable for that category. This can be useful when dealing with ordinal or categorical variables that have a meaningful order.

**Advantages:**

* Captures the relationship between the categorical variable and the target variable.
* Can improve model performance, especially for ordinal variables.

**Ques 63 Examples of Ordinal and Label Encoding**

**Ordinal Encoding:**

* Education level: High School (1), Bachelor's (2), Master's (3), PhD (4)

**Label Encoding:**

* Customer status: Active (1), Inactive (0)

**Ques 64 Target Guided Ordinal Encoding**

**Target Guided Ordinal Encoding** is a technique where the order of categories in an ordinal variable is determined based on the target variable. Categories with similar average target values are assigned similar ordinal values. This can improve model performance by capturing the relationship between the ordinal variable and the target variable.

**Ques 65 Covariance and its Significance in Statistics**

**Covariance** measures the joint variability of two variables. It indicates how much the two variables change together. A positive covariance means the variables increase or decrease together, while a negative covariance means one variable increases as the other decreases.

**Significance:**

* Understanding the relationship between variables.
* Identifying dependencies.
* Building predictive models.

**Ques 66 Correlation Check Process**

1. **Calculate the covariance matrix:** Compute the covariance between all pairs of variables.
2. **Calculate the correlation coefficients:** Divide the covariance by the product of the standard deviations.
3. **Interpret the results:** A correlation coefficient of 1 indicates a perfect positive correlation, -1 indicates a perfect negative correlation, and 0 indicates no correlation.

**Ques 67 Pearson Correlation Coefficient**

The Pearson Correlation Coefficient measures the linear relationship between two variables. It ranges from -1 to 1.

**Ques 68 Difference Between Spearman's Rank Correlation and Pearson's Correlation**

* **Pearson Correlation:** Measures the linear relationship between two variables.
* **Spearman's Rank Correlation:** Measures the monotonic relationship between two variables, regardless of linearity. It is more robust to outliers and non-linear relationships.

**Ques 69 Importance of Variance Inflation Factor (VIF) in Feature Selection**

**VIF** measures the multicollinearity among features. A high VIF indicates that a feature is highly correlated with other features, which can lead to unstable models and difficulty in interpreting the results.

**Ques 70 Feature Selection and Its Purpose**

**Feature selection** is the process of selecting the most relevant features for a machine learning model. Its purpose is to:

* **Improve model performance:** Reduce noise and improve generalization.
* **Reduce computational cost:** Simplify the model and make it faster to train and deploy.
* **Increase interpretability:** Make the model easier to understand.

**Ques 71 Recursive Feature Elimination (RFE)**

**RFE** is a wrapper method for feature selection. It involves:

1. **Training a model** with all features.
2. **Ranking features** based on their importance.
3. **Eliminating the least important feature**.
4. **Retraining the model** with the remaining features.
5. **Repeating steps 2-4** until a desired number of features is reached.

**Backward Elimination**

**Backward elimination** is a similar approach to RFE, but it starts with all features and gradually eliminates the least important ones until the model's performance starts to deteriorate.

**Advantages and Disadvantages of Forward Elimination**

**Advantages:**

* Can be efficient for large datasets.
* Can identify important features that might be missed by other methods.

**Disadvantages:**

* May miss important combinations of features.
* Can be computationally expensive for large datasets.

**Ques 72 Feature Engineering and Its Importance**

**Feature engineering** is the process of creating new features from existing data to improve model performance. It is crucial because:

* **Improves model accuracy:** Can capture hidden patterns and relationships in the data.
* **Reduces dimensionality:** Can reduce the number of features, making the model faster and easier to interpret.
* **Enhances interpretability:** Can create features that are more meaningful and easier to understand.

**Ques 73Steps Involved in Feature Engineering**

1. **Data exploration:** Understand the data and identify potential features.
2. **Feature creation:** Create new features using mathematical operations, transformations, or domain knowledge.
3. **Feature selection:** Choose the most relevant features using techniques like RFE or correlation analysis.
4. **Feature scaling:** Standardize or normalize features to improve model performance.

**Ques 74 Examples of Feature Engineering Techniques**

* **Aggregation:** Create new features by aggregating existing features (e.g., mean, sum, maximum).
* **Transformation:** Apply transformations to existing features (e.g., log, square root, normalization).
* **Interaction:** Create new features by combining existing features (e.g., multiplying, dividing).
* **Time-series features:** Create features based on time-series data (e.g., lags, differences).
* **Domain-specific features:** Create features based on domain knowledge (e.g., customer lifetime value, churn probability).

**Ques 75 Difference Between Feature Selection and Feature Engineering**

* **Feature selection:** Chooses a subset of existing features.
* **Feature engineering:** Creates new features from existing data.

**Ques 76 Importance of Feature Selection in Machine Learning Pipelines**

* **Improves model performance:** Reduces noise and improves generalization.
* **Reduces computational cost:** Simplifies the model and makes it faster to train and deploy.
* **Increases interpretability:** Makes the model easier to understand.

**Ques 79 Impact of Feature Selection on Model Performance**

* **Improved accuracy:** Can improve model accuracy by focusing on the most relevant features.
* **Reduced overfitting:** Can prevent overfitting by reducing the number of features.
* **Increased interpretability:** Makes the model easier to understand and explain.

**Ques 80 Determining Which Features to Include**

* **Domain knowledge:** Use your understanding of the problem to identify relevant features.
* **Correlation analysis:** Identify features that are highly correlated with the target variable.
* **Feature importance:** Use techniques like RFE or permutation importance to assess feature importance.
* **Trial and error:** Experiment with different combinations of features to find the best set.