# Pre and post-assessment

From beginner to pro (No Coding):

Pre-Assessment:

### **Pandas**

**1. Question**: What is the most efficient way to filter rows in a pandas DataFrame based on multiple conditions?

A. Using the .iterrows() method

B. Using the .apply() function  
C. Using boolean indexing  
D. Using nested for loops

**Answer**: A  
**Explanation**: Boolean indexing is the most efficient method for filtering rows based on conditions without iterating over the DataFrame.

### **Data Analysis**

**2. Question**: How would you handle missing data in a DataFrame if the percentage of missing values in a column is less than 5%?

A. Drop the column  
B. Drop the rows with missing values  
C. Fill missing values using mean or median  
D. Replace missing values with zero

**Answer**: C

**Explanation**: If the percentage of missing values is low, filling them with the mean or median is a standard practice.

### 

### **Machine Learning**

**3. Question**: Explain the concept of bias-variance tradeoff in machine learning. How would you balance bias and variance when building a model for a high-dimensional dataset prone to overfitting?

A. Use a high-complexity model like deep learning to avoid bias  
B. Reduce the model complexity by removing features  
C. Use cross-validation and increase the number of features  
D. Increase the training data size and apply regularization

**Answer**: D  
**Explanation**: Increasing data size helps reduce variance, while regularization techniques (like L2) can control overfitting and reduce bias.

### **Python**

**4. Question**: Which of the following methods is used to combine two lists in Python?

A. list.append()  
B. list.extend()  
C. list.insert()  
D. list.remove()

**Answer**: B  
**Explanation**: The list.extend() method adds elements of another list to the end of the current list.

**5. Question**: What is the primary purpose of normalization in data preprocessing?

A. To remove duplicates from the dataset  
B. To scale features to a standard range, such as [0, 1]  
C. To detect outliers in the data  
D. To convert categorical variables into numeric form

**Answer**: B  
**Explanation**: Normalization scales features to a standard range, which is important for machine learning models sensitive to feature magnitudes.

**6. Question**: Which SQL function can be used to get the total number of rows in a table?

A. COUNT()  
B. SUM()  
C. AVG()  
D. MAX()

**Answer**: A  
**Explanation**: The COUNT() function returns the total number of rows in a table, either overall or based on specific conditions.

### **Matplotlib**

**7. Question**: Which Matplotlib function is used to display multiple plots in a single figure?

A. plt.grid()  
B. plt.figure()  
C. plt.subplots()

D. plt.multi()

**Answer**: C  
**Explanation**: plt.subplots() is used to create multiple subplots within a single figure, allowing multiple plots to be displayed.

### **SQL**

**8. Question**: How would you retrieve records where a column value is null in SQL?

A. SELECT \* FROM table WHERE column = NULL;  
B. SELECT \* FROM table WHERE column IS NULL;  
C. SELECT \* FROM table WHERE column IS NOT NULL;  
D. SELECT \* FROM table WHERE column LIKE NULL;

**Answer**: B  
**Explanation**: The correct syntax to check for null values in SQL is IS NULL, as = NULL does not work due to SQL's handling of nulls.

**9. Question**: Which of the following evaluation metrics is most suitable for an imbalanced classification problem?

A. Accuracy  
B. precision   
C. F1 Score  
D. Recall

**Answer**: C  
**Explanation**: The F1 Score is a better metric for imbalanced classification problems as it considers both precision and recall, providing a balanced measure of a model’s performance when the classes are unevenly distributed.

**10. Question**: What is the main advantage of using a Random Forest over a single Decision Tree?

A. It reduces the training time

B. It is less prone to overfitting

C. It improves interpretability

D. All of the above

**Answer**: B

**Explanation**: Random Forests reduce overfitting by averaging the predictions of multiple decision trees, which helps in creating a more generalized model.

**11. Question**: Which measure is generally faster to compute when determining the best split in a decision tree?

A. Entropy  
B. Gini Index  
C. Both are equally fast  
D. It depends on the dataset size

**Answer**: B  
**Explanation**: The Gini Index is generally faster to compute than entropy because it involves fewer mathematical operations, making it more efficient during tree construction.

**12. Question**: In the boosting process, which of the following best describes how the weight adjustments for training instances are made over iterations?

A. The weight of each training instance is reset to an equal value in every iteration  
B. Instances that are classified correctly in previous iterations have their weights increased to reinforce learning  
C. The weights of instances that are misclassified in previous iterations are adjusted to increase their influence on the model  
D. Instances that are misclassified in previous iterations have their weights decreased to reduce their impact

**Answer**: C  
**Explanation**: Boosting adjusts the weights of training instances by increasing the weight of those misclassified in previous iterations. This adjustment helps the model to focus more on these difficult cases in subsequent iterations.

### **13. Scenario: Fraud Detection Model**

**Question**: You have developed a fraud detection model for a financial institution. The dataset consists of a large number of legitimate transactions and a small number of fraudulent transactions. After training and evaluating the model, you find that it achieves a high accuracy score, as it correctly classifies most of the legitimate transactions.

**Scenario**: Despite the high accuracy, the model performs poorly in identifying fraudulent transactions, resulting in a very low F1 Score. This is because the model is highly biased towards predicting transactions as legitimate, given their abundance in the dataset.

**Question**: Why might the model achieve high accuracy but have a very poor F1 Score in this case, and which metric should you focus on to better evaluate the model's performance?

**A. The model's high accuracy is misleading, but the F1 Score is not relevant in this context. Focus on overall accuracy for performance evaluation.**

**B. The model achieves high accuracy because it has a high precision for the fraudulent class, which is why the F1 Score is poor. Focusing on precision alone would be sufficient for evaluating fraud detection performance.**

**C. The model's high accuracy is due to overfitting on the fraudulent class. The F1 Score is high because it is calculated only for the fraudulent class.**

**D. The model achieves high accuracy by predicting most transactions as legitimate, which means it has a low recall for the fraudulent class, leading to a poor F1 Score. Focusing on the F1 Score is important for understanding the balance between precision and recall in detecting fraud.**

**Answer**: D  
**Explanation**: In this scenario, the model achieves high accuracy by predominantly predicting the majority class (legitimate transactions) correctly. However, this results in poor performance in detecting the minority class (fraudulent transactions), leading to a low recall and, consequently, a low F1 Score. The F1 Score provides a better evaluation of the model's ability to correctly identify both classes, balancing precision and recall.

**14. Question**: You have a list of integers and want to create a new list that contains the squares of the even numbers from the original list. Which of the following Python list comprehensions will achieve this?

A. [x\*\*2 for x in my\_list if x % 2 != 0]  
B. [x for x in my\_list if x % 2 == 0\*\*2]   
C. [x\*\*2 if x % 2 == 0 for x in my\_list]  
D. [x\*\*2 for x in my\_list if x % 2 == 0]

**Answer**: D  
**Explanation**: The list comprehension [x\*\*2 for x in my\_list if x % 2 == 0] correctly filters the even numbers (x % 2 == 0) and then computes their squares (x\*\*2).

### 

**15. Question**: What does the map() function do in Python?

A. It creates a dictionary with the items of an iterable as keys and the results of applying a function as values.

B. It maps a list of values to their indices in the original list.  
C. It applies a given function to each item of an iterable (like a list) and returns a map object (which can be converted to a list).

D. It maps the items of an iterable to a new iterable of the same length.

**Answer**: C  
**Explanation**: The map() function applies a given function to each item of an iterable and returns an iterator (map object) that can be converted to a list or other iterable types.