Question 1: What is the difference between AI, ML, DL, and Data Science? Provide a brief explanation of each.

Answer:
Artificial Intelligence (AI) is a broad field focused on creating systems that mimic human intelligence in decision-making, reasoning, and perception. Machine Learning (ML) is a subset of AI that enables systems to learn patterns from data and improve automatically. Deep Learning (DL) is a specialized subset of ML that uses multi-layer neural networks to learn from large and complex datasets like images, speech, and text. Data Science is an interdisciplinary field that combines statistics, programming, and domain knowledge to extract insights from data for decision-making and business understanding.

Question 2: Explain overfitting and underfitting in ML. How can you detect and prevent them?

Answer:
Overfitting occurs when a model learns the training data too closely, resulting in low training error but high test error because it fails to generalize. Underfitting occurs when the model is too simple and performs poorly on both training and testing data. Overfitting can be detected when there's a large gap between training and validation accuracy. It can be prevented using regularization (L1/L2), reducing model complexity, cross-validation, dropout, or collecting more data. Underfitting can be prevented by increasing model complexity, adding more features, or reducing regularization. The bias-variance trade-off helps select the right model complexity.

Question 3: How would you handle missing values in a dataset? Explain at least three methods with examples.

Answer:
One method is **deletion**, where rows containing missing values are removed when the percentage of missing values is very small. A second method is **imputation using statistics**, where missing numeric values are replaced with mean/median and categorical values with mode. A third method is **predictive modeling**, where techniques like regression or KNN imputation estimate the missing values using other available features. Advanced multivariate imputation methods can also be used to preserve relationships between features.

Question 4: What is an imbalanced dataset? Describe two techniques to handle it.

Answer:
An imbalanced dataset is one in which one class has significantly more samples than another, making the model biased toward the majority class. One solution is **resampling**, such as oversampling the minority class (e.g., SMOTE) or undersampling the majority class. Another approach is **algorithmic handling**, where class weights are added to give more importance to minority classes during training, and performance is evaluated using metrics like F1-score, precision-recall, and ROC-AUC instead of accuracy.

Question 5: Why is feature scaling important in ML? Compare Min-Max scaling and Standardization.

Answer:
Feature scaling is important because many ML models rely on distance measurements or gradient-based optimization and perform poorly when features are on different scales. **Min-Max scaling** transforms data to a [0,1] range, preserving the shape but being sensitive to outliers. **Standardization (Z-score scaling)** transforms features to mean 0 and standard deviation 1, making it suitable for models that assume Gaussian-distributed data. Standardization is preferred for algorithms like SVM, Logistic Regression, and Neural Networks.

Question 6: Compare Label Encoding and One-Hot Encoding. When would you prefer one over the other?

Answer:
Label Encoding converts each category into a unique integer and is suitable for **ordinal features** where order matters (e.g., low, medium, high). One-Hot Encoding creates separate binary columns for each category and is suitable for **nominal features** where no order exists (e.g., colors, cities). Label Encoding is memory-efficient and works well with tree-based models, while One-Hot Encoding is preferred for linear and distance-based models to avoid implying false order in data.

Question 7: Google Play Store Dataset — Relationship between Categories and Ratings

Python Code:

```
import pandas as pd

url = "<raw_url_to_google_play_store_dataset>"
df = pd.read_csv(url)

df = df[['Category', 'Rating']].dropna()
df['Rating'] = pd.to_numeric(df['Rating'], errors='coerce')
df = df.dropna()

stats =
df.groupby('Category')['Rating'].mean().sort_values(ascending=False)
print(stats.head(10))
print(stats.tail(10))
```

Short Explanation:
This code calculates the average rating for each category and ranks them from highest to lowest. Categories with niche applications often show higher ratings due to fewer apps with dedicated users, while categories like communication or tools may show lower ratings due to large user bases and varying app quality.

Question 8: Titanic Dataset — Effect of Class and Age on Survival

Python Code:

```
import pandas as pd

url = "<raw_url_to_titanic_dataset>"
df = pd.read_csv(url)

# survival rate by class
print(df.groupby('Pclass')['Survived'].mean())
```

```
# survival by age groups
df = df.dropna(subset=['Age'])
df['AgeGroup'] = df['Age'].apply(lambda x: 'Child' if x < 18 else
'Adult')
print(df.groupby('AgeGroup')['Survived'].mean())
```

Short Explanation:
The results generally show that 1st-class passengers had the highest survival rate due to location near lifeboats and priority treatment. Children usually exhibit a higher survival rate than adults due to the "women and children first" evacuation policy.

Question 9: Flight Price Prediction Dataset — Variation in Prices

Python Code:

```
import pandas as pd

url = "<raw_url_to_flight_price_dataset>"
df = pd.read_csv(url)

# price vs days left
print(df.groupby('Days_left')['Price'].mean())

# airline comparison for a particular route
route = "Delhi-Mumbai"
print(df[df['Route'] == route].groupby('Airline')['Price'].mean())
```

Short Explanation:
Flight prices typically increase as the departure date approaches, with noticeable surges in the last few days. For airline comparison, low-cost carriers generally have lower mean prices, while premium airlines show higher average prices due to service value and pricing strategy.

Question 10: HR Analytics Dataset — Factors Affecting Employee Attrition

Python Code:

```
import pandas as pd

url = "<raw_url_to_hr_analytics_dataset>"
df = pd.read_csv(url)

df['Attrition'] = df['Attrition'].map({'Yes':1, 'No':0})
print(df.corr()['Attrition'].sort_values(ascending=False).head(10))

print(df.groupby('NumProjects')['Attrition'].mean())
```

Short Explanation:
Attributes such as low job satisfaction, high overtime, low salary, and long working hours commonly show strong correlation with attrition. When grouped by project count, attrition patterns reveal whether high workload increases the probability of employees leaving the organization