Q1. Key Features of the Wine Quality Dataset

The Wine Quality dataset (commonly from UCI) includes **physicochemical properties** of wine and a **quality score** (0–10). Key features:

Feature	Description	Importance in Predicting Quality
`fixed acidity`	Tartaric, malic, citric acids	Affects sourness and balance
`volatile acidity`	Acetic acid content	High values → bad taste, spoilage
`citric acid`	Contributes to freshness	Moderate levels improve taste
`residual sugar`	Remaining sugar after fermentation	Too high → overly sweet or unbalanced
`chlorides`	Salt content	High → off-flavors
`free sulfur dioxide`	Preserves wine, prevents oxidation	Too low → spoilage; too high → taste changes
`total sulfur dioxide`	Combined with free SO_2	Similar effect on preservation
`density`	Wine's mass per volume	Related to sugar/alcohol content
`pH`	Acidity measure	Impacts stability and taste
`sulphates`	Adds stability	High levels → better preservation, taste
`alcohol`	Percentage by volume	High alcohol → stronger taste, quality often correlates with alcohol
`quality`	Score 0–10	Target variable

Importance:

- Features like **volatile acidity, alcohol, and sulphates** have strong correlation with wine quality.

- Understanding these features helps **build predictive models** and interpret results for winemaking.

Q2. Handling Missing Data

Common techniques:

Drop missing rows

```
df.dropna(inplace=True)
```

1.
    - **Advantage:** Simple, ensures complete data
    - **Disadvantage:** Loss of data, may bias dataset if missingness is not random

Mean/Median/Mode Imputation

```
df['alcohol'].fillna(df['alcohol'].mean(), inplace=True)
```

2.
    - **Advantage:** Preserves dataset size, simple
    - **Disadvantage:** Can reduce variance, may bias results

K-Nearest Neighbors (KNN) Imputation

```
from sklearn.impute import KNNImputer
imputer = KNNImputer(n_neighbors=3)
df_filled = imputer.fit_transform(df)
```

3.
    - **Advantage:** Uses similarity among samples
    - **Disadvantage:** Computationally intensive for large datasets

4. **Regression Imputation**

- Predict missing values using other features
- Advantage: More accurate if strong correlations exist
- Disadvantage: Assumes linear relationship

Best Practice:

- If <5% missing → mean/median imputation
- If patterns exist → KNN or regression

Q3. Key Factors Affecting Students’ Performance

Factors that influence exam performance:

Factor	Reason
Study time	More preparation → higher score
Attendance	Regular attendance → better understanding
Parental education	Higher education → support at home
Socioeconomic status	Affects resources, stress levels
Health & sleep	Physical/mental condition impacts concentration
Extracurricular activities	Time management, engagement levels

Statistical Analysis Techniques:

- **Correlation analysis:** Find relationships between factors and scores
- **Regression analysis:** Predict scores from multiple features
- **ANOVA:** Compare performance across different groups (e.g., gender, study habits)
- **Chi-square test:** Analyze categorical factors like participation in activities

---

Q4. Feature Engineering for Student Performance Dataset

Steps:

1. **Identify relevant features**
    - Study hours, attendance, parental education, health, etc.
2. **Transform variables**
    - Convert categorical variables into numerical (One-Hot or Label Encoding)
    - Normalize continuous variables (Min-Max or StandardScaler)
3. **Create new features**
    - `study_efficiency = study_time / absenteeism`
    - `stress_index = (hours_of_sleep < 6)`
4. **Select features**
    - Remove features with low correlation with target
    - Use PCA if dimensionality is high
    - Use feature importance from tree-based models

Q5. Exploratory Data Analysis (EDA) of Wine Quality Dataset

Load dataset and check distributions:

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
```

```
df = pd.read_csv('winequality-red.csv')
df.hist(bins=15, figsize=(15,10))
plt.show()

# Check skewness
print(df.skew())
```

Observations:

- Features like `residual sugar`, `chlorides`, `free sulfur dioxide` are often **right-skewed**
- `pH` and `alcohol` are closer to normal

Transformations to improve normality:

- Log transform: `np.log(df['residual sugar'] + 1)`
- Box-Cox transform: `scipy.stats.boxcox(df['chlorides'] + 0.01)`
- Square root: `np.sqrt(df['free sulfur dioxide'])`

Q6. Principal Component Analysis (PCA) on Wine Quality Dataset

Purpose: Reduce dimensionality while retaining most variance.

```
from sklearn.preprocessing import StandardScaler
from sklearn.decomposition import PCA

X = df.drop('quality', axis=1)
y = df['quality']

# Standardize features
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)
```

```
# Apply PCA
pca = PCA().fit(X_scaled)

# Explained variance ratio
import numpy as np
cumulative_variance = np.cumsum(pca.explained_variance_ratio_)
n_components = np.argmax(cumulative_variance >= 0.95) + 1
print("Minimum number of components to retain 95% variance:",
n_components)
```

Interpretation:

- Usually, **6–8 components** can retain ~95% of variance
- Reduces feature space → faster model training, less multicollinearity

---

✅ Summary Table for Wine Quality EDA & Feature Engineering

Step	Technique	Purpose
Missing data	Mean/Median/KNN imputation	Fill gaps without biasing
Feature scaling	StandardScaler	Required for PCA and some ML models
Feature transformation	Log/Box-Cox	Reduce skewness, improve normality
PCA	Dimensionality reduction	Retain variance, reduce features
EDA	Histograms, skewness, correlations	Identify feature importance and relationships

---