Comprehensive Guide to Data Science Concepts using Titanic Dataset - Set 1 (Questions 1–12)

Q1. How to identify null values?

Concept Explanation:
Null values are missing entries in your dataset. Identifying them is essential to decide how to handle them.

Detailed Answer:
They can be detected using pandas methods like `.isnull()` and `.sum()` to get column-wise null counts.

Code Example:

```python
import pandas as pd
df = pd.read_csv("titanic.csv")
print(df.isnull().sum())
```

Dry Run:
Prints count of missing values per column.

Summary (Hinglish):
Jo data missing hai usko null kehte hain. Upar wale code se har column ka null count milta hai.

Q2. How to impute null values?

Concept Explanation:
Imputation means replacing missing values with meaningful data.

Detailed Answer:
We can use statistical measures like mean, median, or mode to fill in missing values.

Code Example:

```python
# Fill Age with median
df['Age'].fillna(df['Age'].median(), inplace=True)

# Fill Embarked with mode
df['Embarked'].fillna(df['Embarked'].mode()[0], inplace=True)
```

Summary (Hinglish):
Jo values missing hain unko average (mean/median) ya mostly used value (mode) se replace karte hain.

Q3. Methods of null value imputation and removal

Concept Explanation:
You can either fill (impute) or remove rows/columns with nulls.

Detailed Answer:
Techniques include: dropping rows, filling with mean/median/mode, forward fill, backward fill.

Code Example:

```python
# Drop rows with any nulls
df.dropna(inplace=True)

# Forward fill
df.fillna(method='ffill', inplace=True)

# Backward fill
df.fillna(method='bfill', inplace=True)
```

Summary (Hinglish):
Null values ko ya to hata do ya phir kuch value se bhar do jaise mean ya mode.

Q4. What are outliers?

Concept Explanation:
Outliers are extreme values that deviate from the rest of the data.

Detailed Answer:
They can distort statistical analyses and ML models. Often identified in numeric columns.

Summary (Hinglish):
Outliers matlab aise numbers jo baaki data se bahut zyada alag hote hain.

Q5. Ways to identify outliers

Concept Explanation:
Common methods: IQR, Z-score, boxplots (for visual inspection).

Code Example:

```python
# Using IQR to find outliers in 'Age'
Q1 = df['Age'].quantile(0.25)
Q3 = df['Age'].quantile(0.75)
IQR = Q3 - Q1

outliers = df[(df['Age'] < Q1 - 1.5  IQR) | (df['Age'] > Q3 + 1.5  IQR)]
print(outliers[['Age']])
```

Summary (Hinglish):
IQR method se hum find kar sakte hain kaunse values bahar ke hain (outliers).

Q6. How to remove outliers?

Concept Explanation:
Once detected, outliers can be dropped to improve model performance.

Code Example:

```python
# Remove outliers based on IQR in 'Age'
df = df[(df['Age'] >= Q1 - 1.5  IQR) & (df['Age'] <= Q3 + 1.5  IQR)]
```

Summary (Hinglish):
Outlier values ko hata dene se model zyada accurate ho sakta hai.

Q7. What are independent and dependent variables?

Concept Explanation:
Dependent variable is what we predict. Independent variables are used for prediction.

Example:
In Titanic dataset:

 Dependent: Survived
 Independent: Age, Sex, Pclass, etc.

Summary (Hinglish):

Jiska prediction karna hai woh dependent, aur jo input dete hain woh independent variables hain.

Q8. What is Standard Deviation and Covariance?

Concept Explanation:

 Standard Deviation: spread of data
 Covariance: relationship between two variables

Code Example:

```python
print(df['Age'].std())           # Standard Deviation
print(df[['Age', 'Fare']].cov())  # Covariance
```

Summary (Hinglish):
Std deviation se data ka spread pata chalta hai. Covariance batata hai do columns ka relation.

Q9. Linear, Logistic, Naive Bayes - Explanation

Linear Regression: Predict continuous values
Logistic Regression: Predict binary outcomes
Naive Bayes: Classification using probability

Advantages:

 Simple, fast, interpretable

Disadvantages:

 Can underperform with complex data

Summary (Hinglish):
Linear regression numbers predict karta hai, logistic 0/1. Naive Bayes probability ke base pe kaam karta hai.

Q10. Supervised vs Unsupervised Learning

Supervised: Data has labels (e.g., Survived)
Unsupervised: No labels (e.g., clustering passengers)

Summary (Hinglish):
Supervised mein answers diye hote hain, unsupervised mein nahi hote.

Q11. Why perform Scaling and Normalization?

Concept Explanation:
To bring all features to the same scale, especially important for distance-based models.

Summary (Hinglish):
Features ka scale same karne ke liye scaling/normalization ki zarurat padti hai.

Q12. How to perform Scaling and Normalization?

Code Example:

```python
from sklearn.preprocessing import MinMaxScaler, StandardScaler

# Normalization (0 to 1)
scaler = MinMaxScaler()
df[['Age', 'Fare']] = scaler.fit_transform(df[['Age', 'Fare']])

# Standardization (mean=0, std=1)
scaler = StandardScaler()
df[['Age', 'Fare']] = scaler.fit_transform(df[['Age', 'Fare']])
```

Summary (Hinglish):
Normalization se values 0-1 ke beech aati hain, standardization mein mean 0 aur std 1 ho jaata hai.

Comprehensive Guide to Data Science Concepts using Titanic Dataset - Set 1 (Questions 1–12)

\[... existing content from Q1 to Q12 remains unchanged ...]

Q13. How and why to convert categorical values into numerical (all methods)

Concept Explanation:
Most machine learning algorithms require numerical input. Categorical variables

like "Sex" or "Embarked" must be converted into numbers so that models can understand them.

Detailed Answer:
There are several ways to convert categorical columns:

1. Label Encoding: Assigns each category a unique integer. Useful when categories are ordinal.
2. One-Hot Encoding: Creates binary columns for each category (used when there’s no order in the categories).
3. Mapping: Convert using dictionaries manually (used when specific mapping is required).
4. get_dummies(): A pandas shortcut for one-hot encoding.

Code Example:

```python
# Label Encoding (if order matters)
df['Sex'] = df['Sex'].map({'male': 0, 'female': 1})

# One-hot encoding using pandas
df = pd.get_dummies(df, columns=['Embarked'], drop_first=True)
```

Summary (Hinglish):
Model ko sirf numbers samajh aate hain, isliye "male/female" jaise words ko 0,1 ya alag columns mein convert karna padta hai.

Q14. What is Hyperparameter Tuning?

Concept Explanation:
Hyperparameters are settings that you configure before training a machine learning model (like number of trees in Random Forest).

Detailed Answer:
Tuning is the process of finding the best hyperparameters for better performance.
Common methods:

1. Grid Search
2. Random Search
3. Bayesian Optimization

Code Example:

```python
from sklearn.model_selection import GridSearchCV
from sklearn.ensemble import RandomForestClassifier
```

```
param_grid = {'n_estimators': [50, 100], 'max_depth': [4, 6]}
grid = GridSearchCV(RandomForestClassifier(), param_grid, cv=5)
grid.fit(X_train, y_train)
print(grid.best_params_)
```

Summary (Hinglish):
Model ke parameters jaise "kitne trees banane hain" ko tune karne se model ka result improve hota hai.

Q15. What are evaluation parameters for classification and regression algorithms?

Concept Explanation:
Evaluation parameters help us measure how well our model is performing.

Classification Metrics:

- Accuracy
- Precision, Recall, F1-score
- Confusion Matrix

Regression Metrics:

- Mean Absolute Error (MAE)
- Mean Squared Error (MSE)
- Root Mean Squared Error (RMSE)
- R-squared (R^2)

Code Example:

```python
from sklearn.metrics import accuracy_score, mean_squared_error, r2_score

# Classification example
print("Accuracy:", accuracy_score(y_test, y_pred))

# Regression example
print("MSE:", mean_squared_error(y_test, y_pred))
print("R2:", r2_score(y_test, y_pred))
```

Summary (Hinglish):
Model sahi kaam kar raha hai ya nahi, yeh check karne ke liye accuracy, error score jaise tools use karte hain.

Q16. What is Precision, Recall, F1-Score, MSE, MAE, R^2 Score?

Concept Explanation:
These are evaluation metrics:

 Precision: True positives out of predicted positives
 Recall: True positives out of actual positives
 F1-score: Harmonic mean of precision and recall
 MSE/MAE: Average squared or absolute errors in predictions
 R² Score: Explains how much variance is captured by the model

Code Example:

```python
from sklearn.metrics import precision_score, recall_score, f1_score,
mean_absolute_error

print("Precision:", precision_score(y_test, y_pred))
print("Recall:", recall_score(y_test, y_pred))
print("F1:", f1_score(y_test, y_pred))
print("MAE:", mean_absolute_error(y_test, y_pred))
```

Summary (Hinglish):
Precision aur recall batate hain model ne kitne sahi results diye. MSE/MAE error kitna hai yeh batate hain. R² score batata hai model kitna acha fit hua.

Q17. What does the describe() method specify?

Concept Explanation:
The `describe()` method provides basic statistics for numeric columns.

Detailed Answer:
It includes count, mean, std deviation, min, 25%, 50%, 75%, and max values. Helps in understanding data distribution.

Code Example:

```python
print(df.describe())
```

Summary (Hinglish):
Describe se data ke numbers ka summary milta hai jaise average, max, min, etc. Analysis ke start mein useful hota hai.

Q18. What is Data Visualization?

Concept Explanation:
It’s the graphical representation of information and data to identify patterns and trends.

Detailed Answer:
Charts like bar graphs, line charts, histograms help present data in a clear and intuitive way. It simplifies complex relationships.

Summary (Hinglish):
Data ko samajhne ke liye uska graph banana easy hota hai. Isse patterns aur problems dono jaldi dikh jaate hain.

Comprehensive Guide to Data Science Concepts using Titanic Dataset - Set 1 (Questions 1–12)

\[... existing content from Q1 to Q12 remains unchanged ...]

\[... existing content from Q13 to Q18 remains unchanged ...]

Q19. What is the difference between Univariate, Bivariate, and Multivariate Analysis?

Concept Explanation:
These terms refer to how many variables (columns) you analyze at a time.

Detailed Answer:

 Univariate Analysis: Examining a single variable. E.g., analyzing only the "Age" column.
 Bivariate Analysis: Examining the relationship between two variables. E.g., "Age vs Survived".
 Multivariate Analysis: Examining relationships among more than two variables. E.g., "Age, Fare, and Pclass vs Survived".

Code Example:

```python
# Univariate analysis
print(df['Age'].describe())

# Bivariate analysis
print(df.groupby('Sex')['Survived'].mean())
```

```
# Multivariate example
print(df[['Age', 'Fare', 'Pclass', 'Survived']].corr())
```

Summary (Hinglish):
Ek column analyze karna univariate hai, do columns ka relation bivariate, aur 2 se zyada ka multivariate.

Q20. What is RMSE (Root Mean Squared Error)?

Concept Explanation:
RMSE measures how far the predicted values are from actual values. It's the square root of average squared errors.

Detailed Answer:
Lower RMSE means better model. It is more sensitive to large errors due to squaring. It gives a clear idea of how much prediction is off on average.

Code Example:

```python
from sklearn.metrics import mean_squared_error
import numpy as np

mse = mean_squared_error(y_test, y_pred)
rmse = np.sqrt(mse)
print("RMSE:", rmse)
```

Summary (Hinglish):
RMSE batata hai prediction kitna galat ho raha hai. Jitna chhota ho, model utna better.

Q21. How can you select K in K-Means?

Concept Explanation:
“K” is the number of clusters you want to form in K-means algorithm.

Detailed Answer:
Best K is found using:

 Elbow Method: Plot Within-Cluster-Sum-of-Squares (WCSS) vs K. Point where graph bends (elbow) is best K.
 Silhouette Score: Measures how close each point is to its own cluster vs other clusters.

Code Example:

```python
from sklearn.cluster import KMeans
wcss = []
for i in range(1, 11):
    kmeans = KMeans(n_clusters=i, random_state=42)
    kmeans.fit(df[['Fare', 'Age']].dropna())
    wcss.append(kmeans.inertia_)

print(wcss)
```

Summary (Hinglish):
K-means mein kitne groups banane hain (K) yeh elbow graph ya silhouette score se decide karte hain.

Q22. Explain Normal Distribution

Concept Explanation:
A normal distribution is a bell-shaped curve where most values are centered around the mean.

Detailed Answer:
It is symmetrical and follows properties:

Mean = Median = Mode
68% data within 1 std dev, 95% within 2, and 99.7% within 3 std dev
Many statistical methods assume normal distribution.

Code Example:

```python
print(df['Age'].mean())
print(df['Age'].std())

# Check if Age follows normal distribution using skew/kurtosis
print(df['Age'].skew())
```

Summary (Hinglish):
Normal distribution ek bell curve hoti hai jisme zyada values average ke paas hoti hain.

Q23. What is a Confusion Matrix?

Concept Explanation:
It is a summary table showing how well a classification model performs.

Detailed Answer:
It includes:

- True Positives (TP)
- False Positives (FP)
- True Negatives (TN)
- False Negatives (FN)
- Helps in calculating precision, recall, etc.

Code Example:

```python
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test, y_pred)
print(cm)
```

Summary (Hinglish):
Confusion matrix model ke sahi aur galat predictions ka count dikhata hai. Use se precision/recall nikalte hain.

Q24. What are True Positive Rate and False Positive Rate?

Concept Explanation:
These are metrics used in evaluating classification performance.

Detailed Answer:

- True Positive Rate (TPR) = TP / (TP + FN) → Also called Recall
- False Positive Rate (FPR) = FP / (FP + TN)
- They help build ROC curves to compare models.

Code Example:

```python
from sklearn.metrics import roc_curve
y_scores = model.predict_proba(X_test)[:,1]
fpr, tpr, thresholds = roc_curve(y_test, y_scores)
print(f"FPR: {fpr[:3]}, TPR: {tpr[:3]}")
```

Summary (Hinglish):
TPR batata hai kitne actual positive sahi predict huye. FPR batata hai kitne negative ko galti se positive bola.

Q25. What is the difference between Type I and Type II Errors?

Concept Explanation:
These are two kinds of errors in classification.

Detailed Answer:

Type I Error (False Positive): Predict positive when it’s actually negative.
Type II Error (False Negative): Predict negative when it’s actually positive.
We want to minimize both but there's often a trade-off.

Summary (Hinglish):
Type I: Galti se "yes" bola jab "no" tha. Type II: Galti se "no" bola jab "yes" tha.

Q26. How can you calculate accuracy using a confusion matrix?

Concept Explanation:
Accuracy is the proportion of total correct predictions.

Detailed Answer:
Formula: (TP + TN) / Total

Code Example:

```python
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test, y_pred)
accuracy = (cm[0][0] + cm[1][1]) / cm.sum()
print("Accuracy:", accuracy)
```

Summary (Hinglish):
Accuracy ka matlab hai total sahi predictions ka percentage.

Comprehensive Guide to Data Science Concepts using Titanic Dataset - Set 1 (Questions 1–12)

\[... existing content from Q1 to Q12 remains unchanged ...]

\[... existing content from Q13 to Q26 remains unchanged ...]

Q27. Explain what tokenization is in NLP and why it is important?

Concept Explanation:
Tokenization is the process of splitting a sentence or text into individual words or tokens.

Detailed Answer:
In Natural Language Processing (NLP), machines need to break text into understandable units. Tokenization helps in dividing sentences into:

- Words (word-level tokenization)
- Sentences (sentence-level tokenization)

These tokens are used for further tasks like stemming, lemmatization, sentiment analysis, etc.

Code Example:

```python
from nltk.tokenize import word_tokenize
sentence = "The Titanic sank in 1912."
tokens = word_tokenize(sentence)
print(tokens)
```

Summary (Hinglish):
Tokenization matlab sentence ko chhote chhote parts (words) mein todna. NLP mein yeh first step hota hai processing ke liye.

Q28. Explain the difference between stemming and lemmatization?

Concept Explanation:
Both are used to reduce words to their root form, but they work differently.

Detailed Answer:

- Stemming: Rough cut of the word. It may not be a valid word.
- Lemmatization: More accurate, gives dictionary root word based on context.

Code Example:

```python
from nltk.stem import PorterStemmer, WordNetLemmatizer
stemmer = PorterStemmer()
lemmatizer = WordNetLemmatizer()
```

```
print(stemmer.stem("running"))       # run
print(lemmatizer.lemmatize("running", pos='v'))  # run
```

Summary (Hinglish):
Stemming bas word ka last part kaat deta hai. Lemmatization uska proper root word nikalta hai.

Q29. Explain Tokenization, POS Tagging, Stop Words

Concept Explanation:
These are common preprocessing steps in NLP.

Detailed Answer:

 Tokenization: Splitting text into words.
 POS (Part of Speech) Tagging: Labeling words as noun, verb, adjective, etc.
 Stop Words: Common words like “the”, “is”, “in” which don’t add much meaning.

Code Example:

```python
from nltk import pos_tag
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize

text = "Titanic was a huge ship."
tokens = word_tokenize(text)
print("POS Tags:", pos_tag(tokens))

stop_words = set(stopwords.words('english'))
filtered = [word for word in tokens if word.lower() not in stop_words]
print("After removing stop words:", filtered)
```

Summary (Hinglish):
Tokenization words mein todta hai, POS tagging unka role batata hai, aur stop words hata ke important words bachaata hai.

Q30. Explain the difference between a bar chart and a histogram?

Concept Explanation:
Both are used to display data but for different types.

Detailed Answer:

Bar Chart: Used for categorical data (e.g., count of males vs females).
Histogram: Used for numerical data range (e.g., age distribution).

Code Example:

```python
import matplotlib.pyplot as plt

# Bar Chart: Categorical (Sex)
df['Sex'].value_counts().plot(kind='bar')
plt.show()

# Histogram: Numerical (Age)
df['Age'].plot(kind='hist', bins=10)
plt.show()
```

Summary (Hinglish):
Bar chart category dikhata hai (jaise male/female), histogram numbers ke range dikhata hai (jaise age distribution).

Q31. How do you choose the right type of chart for your data?

Concept Explanation:
Choosing the correct chart depends on the type of data and the story you want to tell.

Detailed Answer:

Categorical Data: Use bar charts or pie charts.
Numerical Distribution: Use histograms or boxplots.
Trends Over Time: Use line plots.
Comparison Between Groups: Use grouped bar charts.

Summary (Hinglish):
Data kis type ka hai us par chart choose karte hain – category ke liye bar, number ke liye histogram, time ke liye line plot.

Q32. What libraries or tools do you prefer for data visualization in Python? Why?

Concept Explanation:
Several libraries are available for plotting in Python. Selection depends on complexity and customization needed.

Detailed Answer:

Matplotlib: Base library, gives full control over plot.
Seaborn: Built on Matplotlib, more beautiful and statistical plots.
Plotly: Interactive visualizations.
Pandas: Simple plots directly from DataFrames.

Code Example:

```python
import matplotlib.pyplot as plt
import seaborn as sns

sns.countplot(x='Survived', data=df)
plt.show()
```

Summary (Hinglish):
Simple graphs ke liye Pandas/Matplotlib, sundar aur easy charts ke liye Seaborn, aur interactive ke liye Plotly use karte hain.

Comprehensive Guide to Data Science Concepts using Titanic Dataset - Set 1 (Questions 1–12)

\[... existing content from Q1 to Q12 remains unchanged ...]

\[... existing content from Q13 to Q32 remains unchanged ...]

Q33. How do you handle missing data when creating visualizations?

Concept Explanation:
Missing data can lead to misleading graphs. It's important to deal with missing values before plotting.

Detailed Answer:
You can either remove rows with missing values or fill them using techniques like mean, median, mode, etc. For visualizations, ignoring missing data or clearly marking it helps maintain accuracy.

Code Example:

```python
# Drop rows with missing Age
cleaned_df = df.dropna(subset=['Age'])
```

```
cleaned_df['Age'].plot(kind='hist', bins=10)

# Fill missing Age with median
df['Age'].fillna(df['Age'].median(), inplace=True)
df['Age'].plot(kind='hist', bins=10)
```

Summary (Hinglish):
Graphs se pehle missing data ya to hata do ya fill karo (jaise median se), warna galat results aayenge.

Q34. Explain the difference between qualitative and quantitative data visualization.

Concept Explanation:
Different types of data need different visual tools.

Detailed Answer:

Qualitative (Categorical): Data like gender, class. Visualized using bar charts, pie charts.
Quantitative (Numerical): Data like age, fare. Visualized using histograms, boxplots, line graphs.

Code Example:

```python
# Qualitative
df['Sex'].value_counts().plot(kind='bar')

# Quantitative
df['Fare'].plot(kind='box')
```

Summary (Hinglish):
Category wale data ke liye bar/pie charts use karo. Number wale data ke liye histogram, boxplot etc. use karo.

Q35. How do you evaluate the effectiveness of a data visualization?

Concept Explanation:
A good chart tells a clear story and avoids confusion.

Detailed Answer:
Check for:

 Clarity: Is it easy to understand?
 Relevance: Does it match the purpose?
 Accuracy: Any misrepresented scales?
 Labels and Legends: Are titles, axes and legends clear?

Summary (Hinglish):
Chart tabhi acha hota hai jab samajhne mein easy ho, galat data na dikhaye aur clearly labelled ho.

Q36. What is Scala and Impala? How to install and execute programs in Scala and Impala?

Concept Explanation:
These are big data technologies used for handling large datasets.

Detailed Answer:

 Scala: General-purpose programming language, used with Apache Spark.
 Impala: SQL engine for Hadoop, used for fast queries on big data.

Installation & Execution:

 Scala:

   Install from:
https://www.scala-lang.org/download/
   Use `scala` command to run code.

```scala
// Example Scala code
object Hello {
  def main(args: Array[String]) = println("Hello Titanic!")
}
```

 Impala:

   Part of Cloudera distribution.
   Connect using impala-shell:

```sql
SELECT  FROM titanic LIMIT 10;
```

Summary (Hinglish):
Scala Spark ke saath coding ke liye hota hai. Impala Hadoop mein SQL queries ke liye use hota hai. Dono big data tools hain.
