Exploratory Data Analysis

Questions to always ask before starting

Q1. How big is your data? : df.shape

Q2. How does the data look like: df.head(), df.sample(5) ⇒ will display random rows

Q3. What is the data type of cols: df.info() ⇒ fix any dtype issues and improve memory.

Q4. Are there any missing values: ↳ `df.isnull().sum()` → gives a list of total null values in every column

Q5. How does the data look like mathematically? ya phir descriptive statistics?
Ans: df.describe() ⇒ only useful for numerical values.

Q6 Are there duplicate values?
Ans: df.duplicated().sum()

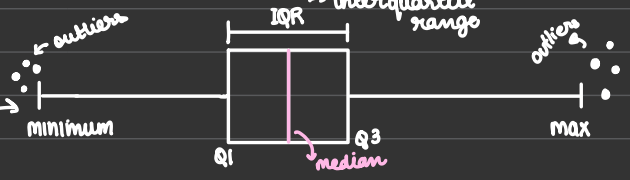
Q7. What is the correlation b/w cols? `df.corr()['column_name']` ⇒ gives a relation with every column

univariate analysis

technique that focuses on examining and summarising a single variable in isolation.

Categorical data
1. bar chart
2. pie chart

numerical data
histogram
box plot
line chart
distplot



bivariate analysis

statiscal analysis of 2 variables to determine the relationship between them It involves examining the distribution, association and interaction b/w them.

pandas profiling

python library for EDA.

profile report

```
from pandas_profiling import ProfileReport
prof = ProfileReport(df)
prof.to_file(output_file='output.html')
```

→ generates a web page for the dataset analysis.
has overviews, warnings and much more at a glance instantly.

Feature Engineering > Feature Scaling

defination

is the process of transforming raw data into a format that is suitable for ML algorithms to extract meaningful patterns and classifications.

overwiew

Transformation	Construction	Selection	Extraction
① missing values	↳ create a new feature from existing ones.	↳ for optimising model	
② categorical features			
③ outlier detection			
④ Feature scaling			

feature scaling ↳ data normalisation

the process of transforming the values of different features (variables) in a dataset to a common scale. This is done to ensure that all features have a similar magnitude and distribution.

when to use feature scaling ?

1. ML algorithms that rely on distance metrics like KNN or SVM.
2. algorithms that use gradient based optimisation such as regression as it helps in achieving faster convergence and prevents certain features from dominating the optimization process due to their larger magnitude.

tip

check the documentation or specific guidelines for the ML algorithm being used to determine if feature scaling is necessary.

types

Standardisation, normalisation

Standardization ↳ Z-score normalisation

$$X_i' = \frac{X_i - \bar{X}}{\sigma}$$

is a feature scaling technique that transforms the values of features to have a mean of 0 and a standard deviation of 1.

when to use

distance based algorithm, linear models, PCA and regularisation

When not to use

decision trees, random forest and sparse data

geometric intuition

data is centered around the mean

Code

1. Train test split
2. from sklearn.preprocessing import StandardScaler

```
scaler = StandardScaler()
scaler.fit(X_train)                      → fitting is on training data only
X_scaled_train = scaler.transform(X_train)   } transformation is
X_scaled_test = scaler.transform(X_test)     } on entire dataset.
```

X_train df → transformed → X train scaled is np array.

impact of outliers

outliers affect both mean and standard deviation causing the standardisation data to be skewed.

Alternative methods like robust scaling or normalisation are better if your data has alot of outliers.

Feature Scaling > Normalisation

minmax scaling	rescales the values of features to a specific range. $X_{norm} = X_i - X_{min} / X_{max} - X_{min}$ it does not change the distribution or shape the data. It only rescales. range will always be in 0-1.
mean normalisation	$X_i' = \frac{X_i - X_{mean}}{X_{max} - X_{min}}$ mean centering.
max absolute scaling ↳ use when data is sparse	also known as max-min scaling, feature scaling technique that rescales the values of a feature to a range between -1 and 1. $X\_scaled = X / max\_abs$. 'max_abs' is the maximum absolute value. It does not center the data around zero or adjust the variance.
median-mad scaling ↳ robust scaling	technique that rescales the values of features based on their robust statistics, making it less sensitive to outliers compared to other scaling methods. $X\_scaled = (x - M)/MAD$ where M is median and MAD = M absolute deviation used when outliers are many.
how is it different from standardisation	most problems will be solved by Standardisation. Normalisation tends to compress or stretches the original distribution It is also more sensitive to outliers. Use it in image processes.

Encoding Categorical data

defination

refers to the process of converting categorical variables into a numerical representation.

types

one hot encoding (dummy coding), Label Encoding, Ordinal Encoding and Binary Encoding.

label encoding

↳ used for target value

Each unique category is assigned to different integer value. For example, consider a categorical variable "Color" with categories 'Red', 'Green' can be assigned 0, 1 and so on for other colors.

ordinal encoding

similar to label encoding however ordinal encoding takes into account the inherent order or hierarchy present in the categories.
example: PhD > Masters then the label assigned would be 3, 2.

```
from sklearn.preprocessing import OrdinalEncoder
oe = OrdinalEncoder(categories=['Poor', 'Average', 'Good'])
oe.fit(X_train)
X_train = oe.transform(X_train)
X_test = oe.transform(X_test)
```

one hot encoding

a technique used to convert categorical variables into binary vectors. It creates a binary columns for each unique category within a categorical feature.

this may introduce high dimensionality to the dataset, especially if there are many unique categories within a variable.

multicollineasity or dummy variable trap.

2 or more predictor variables in a regression model are highly correlat

Y	B	R
1	0	0
0	1	0
0	0	1
1	0	0

→ there is a mathematical relationship formed i.e. $\sum YBR = 1$.
→ this is multicollineasity
AVOID

→ to resolve this → if n dimensions are created keep n-1. Delete the 1st column.

dimensionality