Main strategy overall

You should know the in and out of the data

Step 1 – Univariate Analysis

Step 2 - Bivariate Analysis

* 2 numerical columns
* 2 categorical columns
* 1 numerical + 1 categorical column

Step 3 – Missing values impute (Outliers)

Step 4 – Feature Engineering – adding new columns

Step 5 – Convert categorical to numerical by one hot encoding

EDA Plan

1. head -> tail -> sample

2. for numerical cols

- 8 number summary[count,min,max,mean,std,q1,q2,q3]

- missing values

- outliers

-> horizontal/vertical histograms

3. for categorical cols

- value counts -> pie chart

- missing value

4. numerical - numerical

- side by side 8 number analysis--

- scatterplot

- correlation

5. categorical-categorical

- contigency table -> stacked bar chart

- Ex Cpu\_brand and Company

6. numerical-categorical

→> compare distribution across categories

8. missing value treatment

9. feature engineering - ppi

-price\_bracket

10. one hot encoding

USE laptop\_data;

SELECT \* FROM laptop;

-- 1. head -> tail -> sample

SELECT \* FROM laptop

ORDER BY `index` LIMIT 5;

SELECT \* FROM laptop

ORDER BY `index` DESC LIMIT 5;

SELECT \* FROM laptop

ORDER BY rand() LIMIT 5;

-- Univariate analysis

-- 2. numerical\_col -> price

-- quartiles (Q1, Median, Q3) are not used because there is a syntax error

SELECT

COUNT(Price) AS total\_count,

MIN(Price) AS min\_price,

MAX(Price) AS max\_price,

AVG(Price) AS avg\_price,

STD(Price) AS std\_dev

FROM laptop;

-- just to show you what the ordered\_laptops look like

WITH ordered\_laptops AS (

SELECT Price,

ROW\_NUMBER() OVER (ORDER BY Price) AS row\_num,

COUNT(\*) OVER() AS total\_count

FROM laptop

)

SELECT \* FROM ordered\_laptops;

WITH ordered\_laptops AS (

SELECT Price,

ROW\_NUMBER() OVER (ORDER BY Price) AS row\_num,

COUNT(\*) OVER() AS total\_count

FROM laptop

)

SELECT

COUNT(Price) AS total\_count,

MIN(Price) AS min\_price,

MAX(Price) AS max\_price,

AVG(Price) AS avg\_price,

STD(Price) AS std\_dev,

-- Q1 (25th Percentile)

(SELECT Price FROM ordered\_laptops

WHERE row\_num = FLOOR(0.25 \* total\_count)) AS Q1,

-- Median (50th Percentile)

(SELECT Price FROM ordered\_laptops

WHERE row\_num = FLOOR(0.5 \* total\_count)) AS Median,

-- Q3 (75th Percentile)

(SELECT Price FROM ordered\_laptops

WHERE row\_num = FLOOR(0.75 \* total\_count)) AS Q3

FROM laptop;

-- missing value

SELECT COUNT(Price)

FROM laptop

WHERE Price IS NULL;

-- outliers

WITH ordered\_laptops AS (

SELECT Price,

ROW\_NUMBER() OVER (ORDER BY Price) as row\_num,

COUNT(\*) OVER() AS total\_count

FROM laptop

)

SELECT \*

FROM

(SELECT \*,

(SELECT Price FROM ordered\_laptops

WHERE row\_num = FLOOR(0.25 \* total\_count)) AS Q1,

(SELECT Price FROM ordered\_laptops

WHERE row\_num = FLOOR(0.75 \* total\_count)) AS Q3

FROM laptop) t

WHERE t.Price < t.Q1 - (1.5 \* (t.Q3 - t.Q1)) OR

t.Price > t.Q3 + (1.5 \* (t.Q3 - t.Q1));

-- horizontal/vertical Histogram

-- we have data in range 9k to 343k

-- Bucket size of 25k.

-- 0 - 25, 26 - 50, 51 - 75, 76 - 100, 101 - 350.......

SELECT Price,

CASE WHEN Price BETWEEN 0 and 25000 THEN '\*' ELSE 0 END AS '0-25k',

CASE WHEN Price BETWEEN 25001 and 50000 THEN '\*' ELSE 0 END AS '25k-50k',

CASE WHEN Price BETWEEN 50001 and 75000 THEN '\*' ELSE 0 END AS '50k-75k',

CASE WHEN Price BETWEEN 75001 and 100000 THEN '\*' ELSE 0 END AS '75k-100k',

CASE WHEN Price > 100001 THEN '\*' ELSE 0 END AS '>100k'

FROM laptop;

SELECT t.buckets, REPEAT('\*', COUNT(\*)/5)

FROM

(SELECT Price,

CASE

WHEN Price BETWEEN 0 AND 25000 THEN '0-25k'

WHEN Price BETWEEN 25001 AND 50000 THEN '25k-50k'

WHEN Price BETWEEN 50001 AND 75000 THEN '50k-75k'

WHEN Price BETWEEN 75001 AND 100000 THEN '75k-100k'

WHEN Price > 100001 THEN '>100k'

ELSE '<100k'

END AS 'buckets'

FROM laptop) t

GROUP BY t.buckets;

-- 3. Categorical cols

-- value count = Frequency count

-- Company

SELECT Company, COUNT(Company)

FROM laptop

GROUP BY Company;

-- Bivariate Analysis

-- 4. numerical - numerical

WITH ordered\_laptops AS (

SELECT Price,

ROW\_NUMBER() OVER (ORDER BY Price) AS row\_num,

COUNT(\*) OVER() AS total\_count

FROM laptop

),

laptop\_stats AS (

SELECT

COUNT(Price) AS total\_count,

MIN(Price) AS min\_value,

MAX(Price) AS max\_value,

AVG(Price) AS avg\_value,

STDDEV(Price) AS std\_dev,

(SELECT Price FROM ordered\_laptops

WHERE row\_num = FLOOR(0.25 \* total\_count)) AS Q1,

(SELECT Price FROM ordered\_laptops

WHERE row\_num = FLOOR(0.5 \* total\_count)) AS Median,

(SELECT Price FROM ordered\_laptops

WHERE row\_num = FLOOR(0.75 \* total\_count)) AS Q3

FROM laptop

),

ordered\_inches AS (

SELECT Inches,

ROW\_NUMBER() OVER (ORDER BY Inches) AS row\_num,

COUNT(\*) OVER() AS total\_count

FROM laptop

),

inches\_stats AS (

SELECT

COUNT(Inches) AS total\_count,

MIN(Inches) AS min\_value,

MAX(Inches) AS max\_value,

AVG(Inches) AS avg\_value,

STDDEV(Inches) AS std\_dev,

(SELECT Inches FROM ordered\_inches

WHERE row\_num = FLOOR(0.25 \* total\_count)) AS Q1,

(SELECT Inches FROM ordered\_inches

WHERE row\_num = FLOOR(0.5 \* total\_count)) AS Median,

(SELECT Inches FROM ordered\_inches

WHERE row\_num = FLOOR(0.75 \* total\_count)) AS Q3

FROM laptop

)

SELECT total\_count, min\_value, max\_value, avg\_value, std\_dev, Q1, Median, Q3

FROM laptop\_stats

UNION ALL

SELECT total\_count, min\_value, max\_value, avg\_value, std\_dev, Q1, Median, Q3

FROM inches\_stats;

-- scatter plot

-- Price and cpu\_speed

SELECT cpu\_speed, Price

FROM laptop;

-- correlation - Sorry I don't think it exists

SELECT CORR(cpu\_speed, Price)

FROM laptop;

-- 5. Categorical - Categorical

-- contingency table

SELECT \* FROM laptop;

-- let's assume these numbers

-- Brand 0 1

-- Apple 37 43

-- Dell 110 28

-- HP 67 90

SELECT Company,

SUM(CASE WHEN Touchscreen = 1 THEN 1 ELSE 0 END) AS 'Touchscreen\_yes',

SUM(CASE WHEN Touchscreen = 0 THEN 1 ELSE 0 END) AS 'Touchscreen\_no'

FROM laptop

GROUP BY Company;

-- between cpu\_brand and Company

SELECT DISTINCT cpu\_brand FROM laptop;

SELECT Company,

SUM(CASE WHEN cpu\_brand = 'Intel' THEN 1 ELSE 0 END) AS 'Intel',

SUM(CASE WHEN cpu\_brand = 'Samsung' THEN 1 ELSE 0 END) AS 'Samsung',

SUM(CASE WHEN cpu\_brand = 'AMD' THEN 1 ELSE 0 END) AS 'AMD'

FROM laptop

GROUP BY Company;