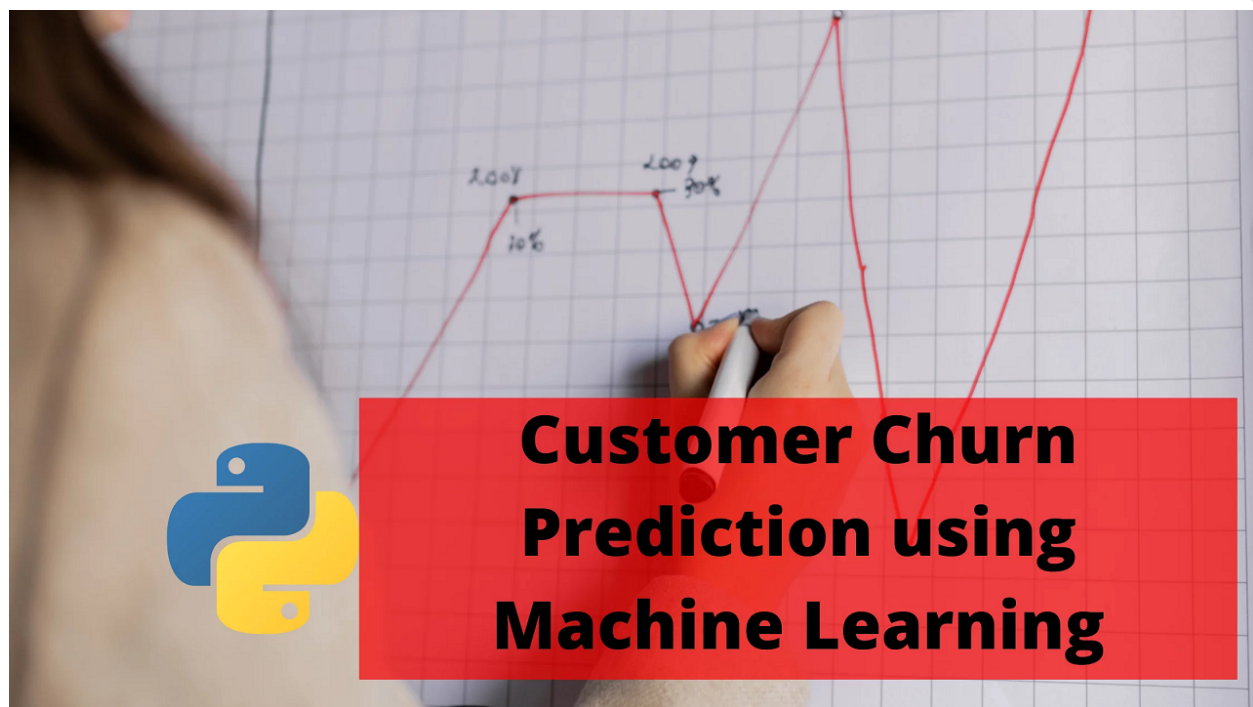


CUSTOMER CHURN PREDICTION



TEAM MEMBERS :

1. SENTHIL VEL S (LEADER)
2. VIGNESH M
3. GUNASEKARAN P
4. GUNASEKARAN S
5. KAMALESH K

Overview :

The objective is to create interactive dashboards and reports in IBM Cognos for analyzing customer churn in a telecommunications company.

STEP 1:

Clean the data, handling missing values and ensuring that it's ready for analysis.

By processing and analyzing the given dataset let's ensure the dataset is free of any

Null values ,duplicates data's ,etc ,.

PROGRAM :

```
import pandas as pd
df=pd.read_csv(r'C:\Users\admin\Downloads\WA_Fn-UseC_-Telco-Customer-Churn.csv')

df.head()

missing_values = df.isnull().sum()

print(missing_values)

df_cleaned = df.dropna()

print(f"Original dataset shape: {df.shape}")

print(f"Cleane dataset shape: {df_cleaned.shape}")

duplicates = df.duplicated().sum()

df_cleaned = df_cleaned.drop_duplicates()

print(f"Number of duplicates: {duplicates}")

print(f"Cleane datasetshape:{df_cleaned.shape}")
```

```
df_cleaned.to_csv('telco_customer_churn_cleaned.csv', index=False)
```

OUTPUT:

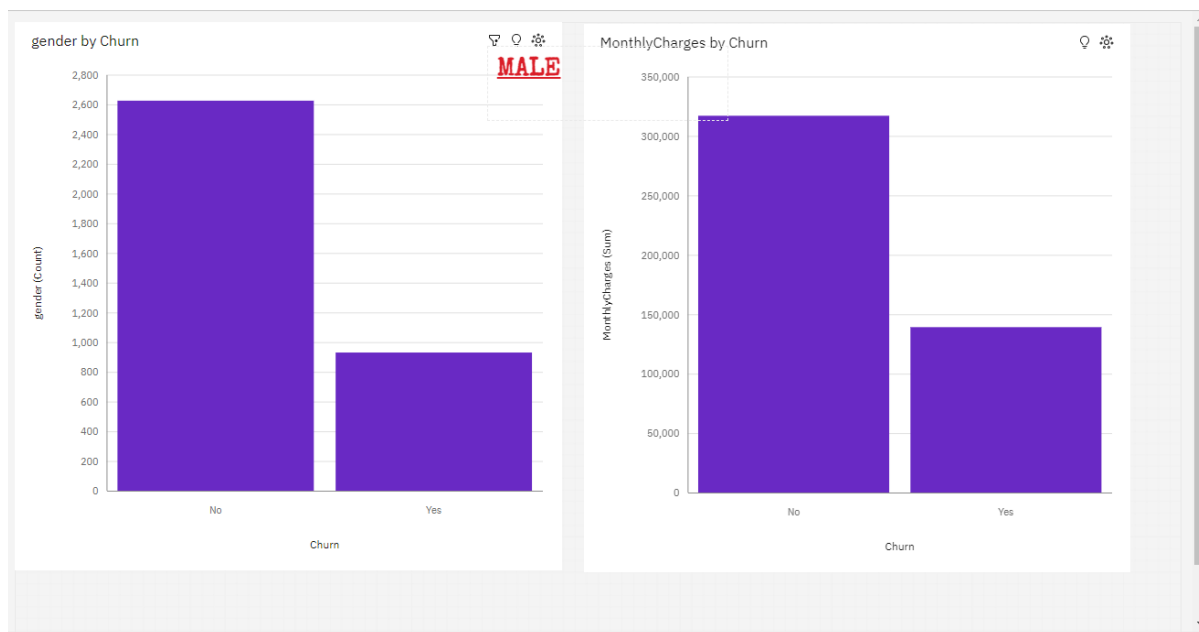
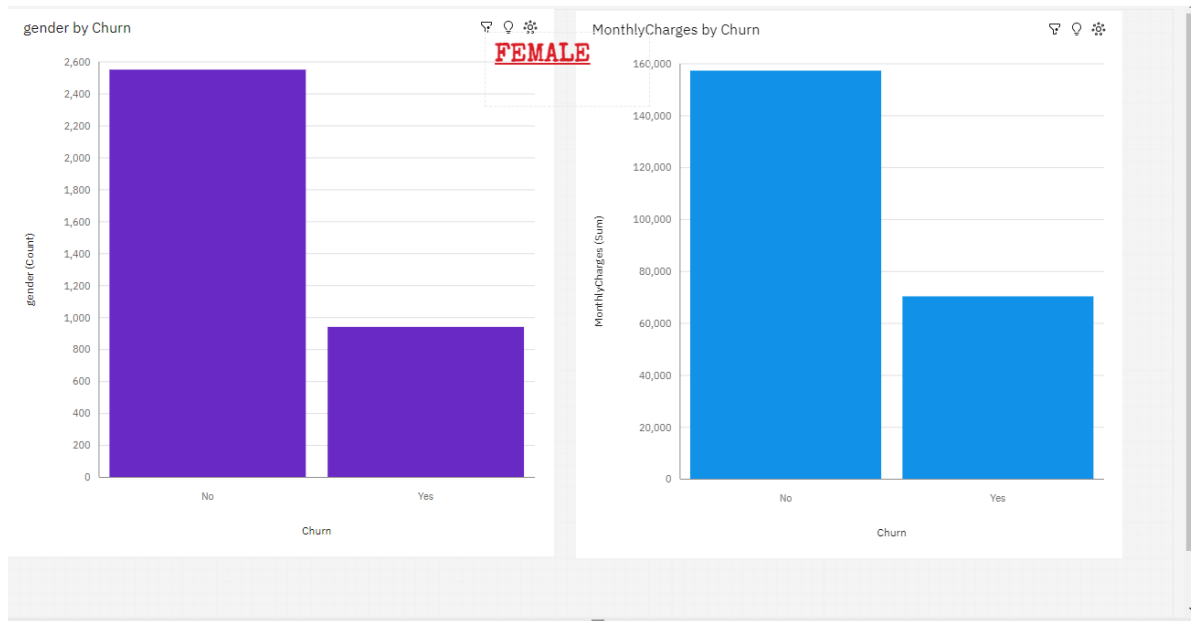
```
customerID      0
gender          0
SeniorCitizen   0
Partner         0
Dependents      0
tenure          0
PhoneService    0
MultipleLines   0
InternetService 0
OnlineSecurity  0
OnlineBackup    0
DeviceProtection 0
TechSupport     0
StreamingTV     0
StreamingMovies 0
Contract        0
PaperlessBilling 0
PaymentMethod   0
MonthlyCharges  0
TotalCharges    0
Churn           0
dtype: int64
Original dataset shape: (7043, 21)
Cleaned dataset shape: (7043, 21)
Number of duplicates: 0
Cleaned dataset shape: (7043, 21)
```

Step 2: Data Visualization and Reporting :

- ❖ Create a data connection to your pre-processed Telco Customer Churn dataset.
- ❖ Design and build visualizations to show churn patterns and retention rates.
- ❖ Some potential visualizations include:
 - Churn rate over time (e.g., monthly or quarterly).
 - Customer demographics (e.g., age, gender, and location) and their impact on churn.
 - Customer tenure and its relationship to churn.
 - Subscription details (e.g., contract type, internet service, and phone service) and their relation to churn.

Customer demographics :

VISUALIZATION:



Analyzing customer demographics, including age, gender, and location, and their impact on churn is crucial for understanding the customer base and tailoring retention strategies.

1. Age and Churn:

- Create a visualization, such as a bar chart or histogram, showing the distribution of customer ages.
- You can also create a stacked bar chart to display the churn rate within different age groups.

2. Gender and Churn:

- Create a pie chart or a bar chart to show the gender distribution among your customers.
- Additionally, create a separate visualization to illustrate churn rates by gender.

Less tenure and high monthly charges

More likely to be Female

High tenure and High monthly charges

More likely to be male and senior citizen

Less tenure and low monthly charges

More likely to be male

PROGRAM:

```
group_gp=df_cluster_gp['Gender'].value_counts(normalize=True).to_frame()  
(
```

```
#pd.concat([group_gp.index.name , group_gp.values])
```

```
group_gp.columns = ['Count']
group_gp = group_gp.reset_index()
group_gp_new = group_gp.copy()
Group_gp_new
```

OUTPUT:

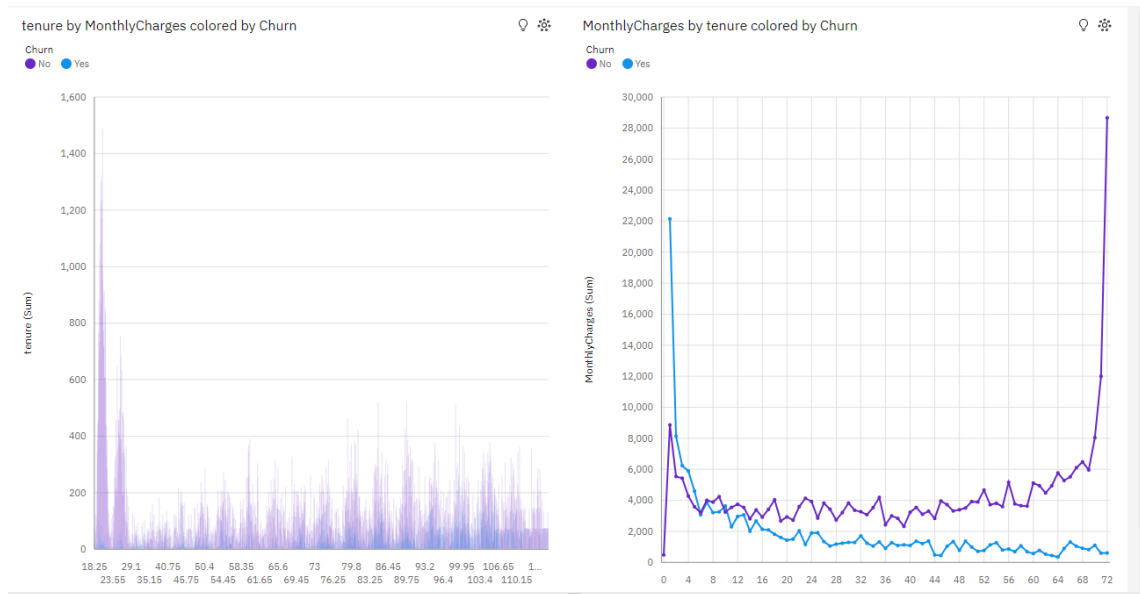
	Cluster	Gender	Count
0	0	Female	0.531085
1	0	Male	0.468915
2	1	Male	0.519397
3	1	Female	0.480603
4	2	Male	0.535088
5	2	Female	0.464912

Analytics Insights:

- Churn No has the highest values of both MonthlyCharges and TotalCharges.
- No and 1 and Yes.MonthlyCharges is unusually high when the combinations of tenure and Churn are 72 and
- MonthlyCharges is unusually high when tenure is 1 and 72.
- Yes Churn accounted for 71% of 1 MonthlyCharges compared to 2% for 72.
- Churn No has the highest MonthlyCharges at nearly 317 thousand, out of which tenure 72 contributed the most at almost 29 thousand.
- tenure is unusually high when MonthlyCharges is 20.05.

- Churn No has the highest tenure at over 194 thousand, out of which MonthlyCharges 20.05 contributed the most at almost 1500.

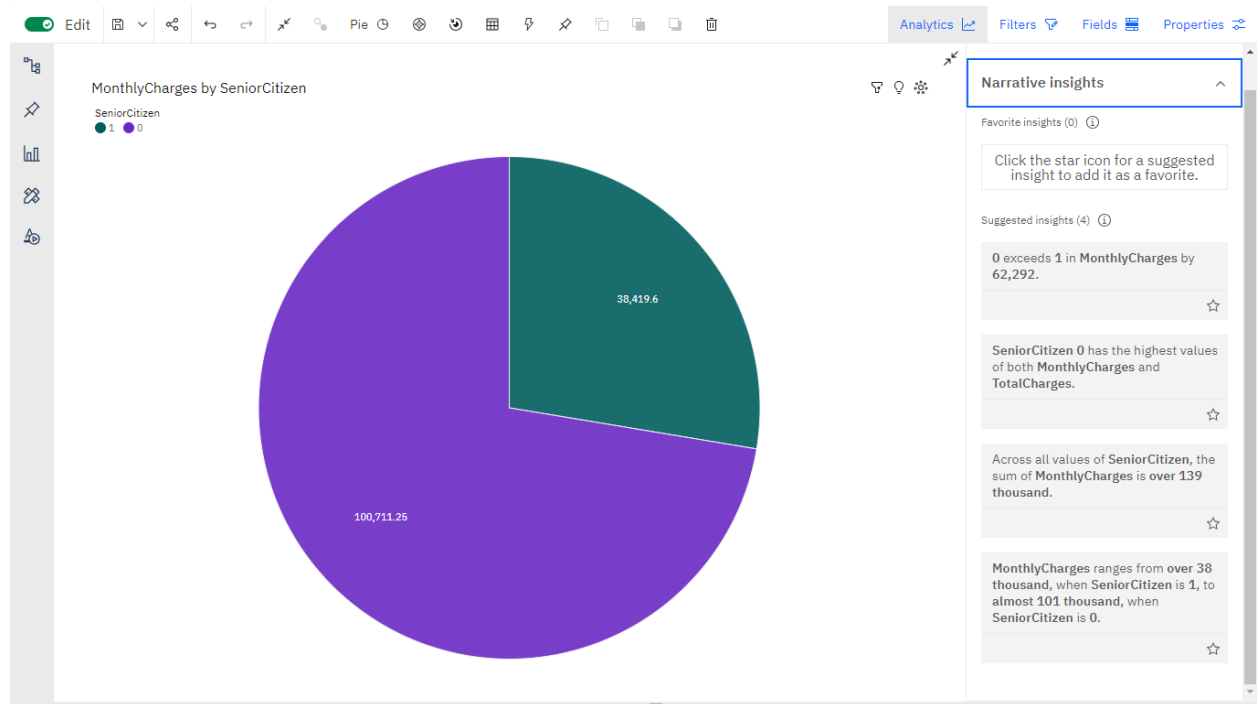
VISUALIZATION :



ANALYTICS INSIGHTS:

- 0 exceeds 1 in MonthlyCharges by 62,292.
- SeniorCitizen 0 has the highest values of both MonthlyCharges and TotalCharges.
- Across all values of SeniorCitizen, the sum of MonthlyCharges is over 139 thousand.
- MonthlyCharges ranges from over 38 thousand, when SeniorCitizen is 1, to almost 101 thousand, when SeniorCitizen is 0

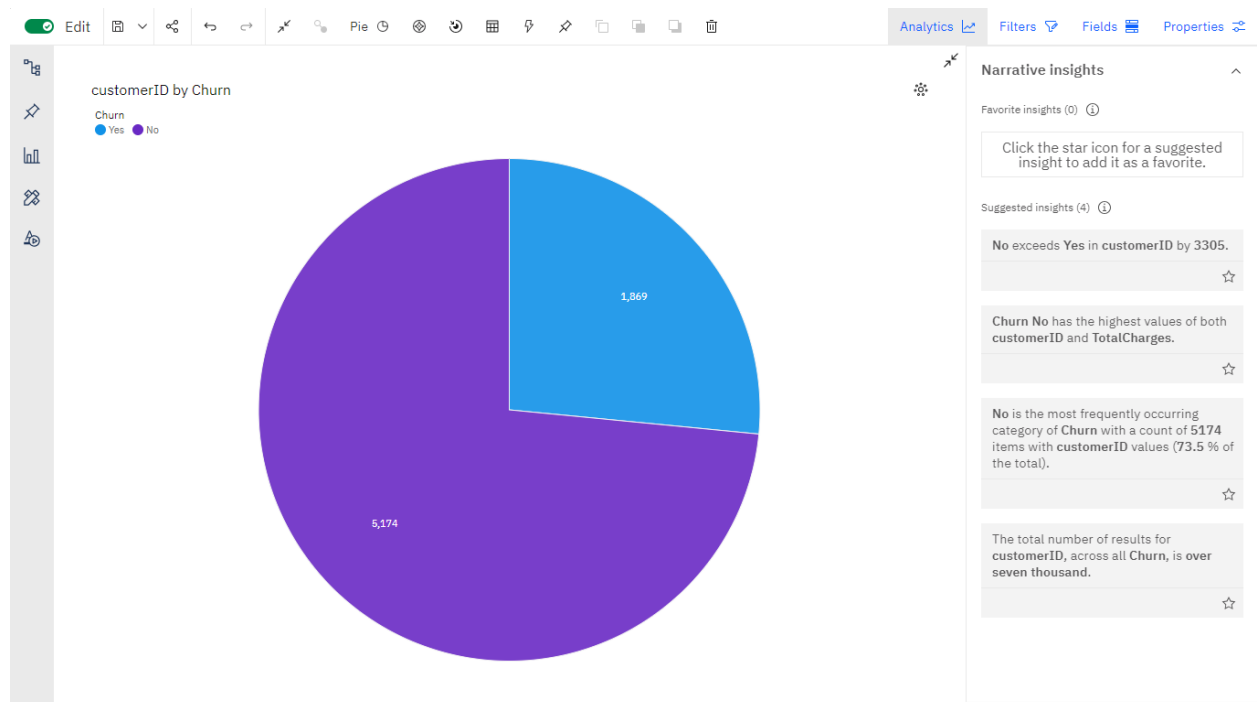
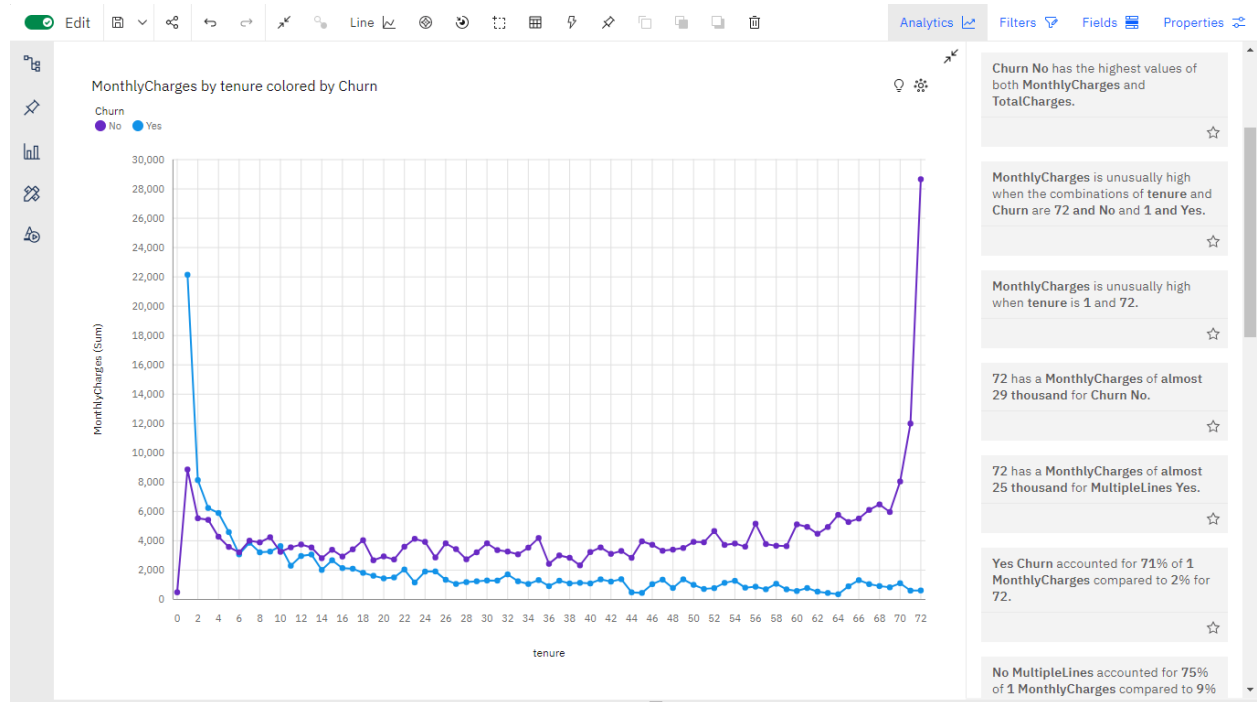
VISUALIZATION:



ANALYTICS:

- ❖ Contract Month-to-month has the highest InternetService due to Churn No.
- ❖ Contract Month-to-month has the highest InternetService due to MultipleLines No.
- ❖ Churn No has the highest values of both InternetService and TotalCharges.
- ❖ The overall number of results for InternetService is over seven thousand.
- ❖ Month-to-month is the most frequently occurring category of Contract with a count of 3875 items with InternetService values (55 % of the total).
- ❖ MultipleLines No has the highest InternetService at 9, out of which Contract Month-to-month contributed the most at 3.

VISUALIZATION:



PREDICTIVE MODEL DEVELOPMENT:

FEATURE SELECTION :

Usage pattern:

```
print("Phone service  
distribution\n",df_cal['PhoneService'].value_counts(normalize=True))  
  
print("Phone & churn distribution  
\n",df_cal.groupby('Churn')['PhoneService'].value_counts(normalize=True))
```

OUTPUT:

Phone service distribution

1 0.903166

0 0.096834

Name: PhoneService, dtype: float64

Phone & churn distribution

Churn PhoneService

0 1 0.901044

0 0.098956

1 1 0.909042

0 0.090958

Name: PhoneService, dtype: float64

MACHINE LEARNING MODEL:

K Means algorithm :

It is an iterative algorithm that divides the unlabeled dataset into k different clusters in such a way that each dataset belongs only one group that has similar properties

PROGRAM:

```
from sklearn.cluster import KMeans

df_kmeans_data =
df_cal[df_cal.Churn==1][['Tenure_norm','MonthlyCharges_norm']]

k = range(1,10)

k means = [KMeans(n_clusters=i) for i in k]

score = [k means[i].fit(df_kmeans_data).score(df_kmeans_data) for i in
range(len(k means))]

plt.figure(figsize=(10,6))

plt.plot(k,score)

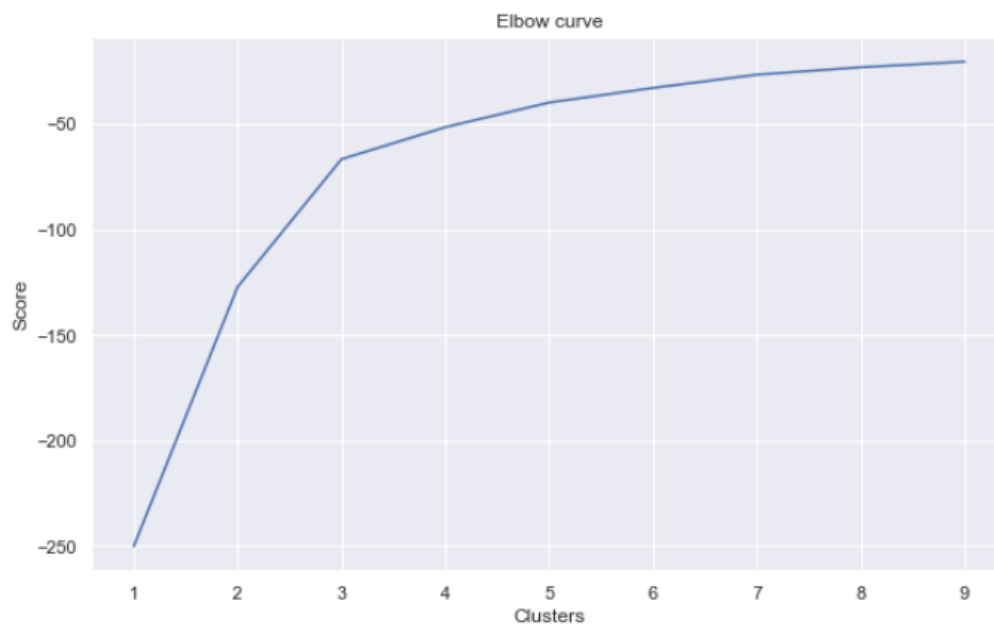
plt.xlabel("Clusters")

plt.ylabel("Score")

plt.title("Elbow curve")
```

OUTPUT:

```
: Text(0.5,1,'Elbow curve')
```



THANKING YOU