Customer Churn Prediction Project Report (SQL + Python)

Project Overview:

Customer churn refers to when customers stop using a company's services. The objective of this project is to develop a **machine learning model** that predicts whether a customer is likely to churn based on their service usage patterns and demographic data.

The project uses:

- **SQL** for data storage, cleaning, and exploratory analysis.
- **Python with Machine Learning** for model building and deployment.

Dataset:

- Source of dataset is Kaggle (Telco-Customer-Churn dataset).
- This dataset contains more than 7000 rows and 21 columns
- Target is predict the customer churn(Yes/No).

Key Features(columns) of dataset :

- CustomerID
- Gender
- SeniorCitizen
- Partner
- Dependents
- Tenure (in months)
- PhoneService
- MultipleLines
- InternetService
- OnlineSecurity
- OnlineBackup
- DeviceProtection
- TechSupport
- StreamingTV
- StreamingMovies
- Contract
- PaperlessBilling
- PaymentMethod
- MonthlyCharges
- TotalCharges
- Churn (Target)

SQL Phase (Data Cleaning & Analysis):

1. **Create the TelcoDB database and create TelcoCustomerChurn table:**

```
    -- create TelcoDB database
    CREATE DATABASE TelcoDB;
     USE TelcoDB;

    -- create TelcoCustomerChurn table
    CREATE TABLE TelcoCustomerChurn (
       customerID VARCHAR(50) PRIMARY KEY,
       gender VARCHAR(10),
       SeniorCitizen INT,
       Partner VARCHAR(10),
       Dependents VARCHAR(10),
       tenure INT,
       PhoneService VARCHAR(10),
       MultipleLines VARCHAR(30),
       InternetService VARCHAR(30),
       OnlineSecurity VARCHAR(30),
       OnlineBackup VARCHAR(30),
       DeviceProtection VARCHAR(30),
       TechSupport VARCHAR(30),
       StreamingTV VARCHAR(30),
       StreamingMovies VARCHAR(30),
       Contract VARCHAR(30),
       PaperlessBilling VARCHAR(10),
       PaymentMethod VARCHAR(50),
       MonthlyCharges FLOAT Null,
       TotalCharges FLOAT,
       Churn VARCHAR(10)
);
```

2. **Load the dataset to MySQL:**

```
-- load the data into table
LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/Telco-Customer-Churn.csv'
INTO TABLE TelcoCustomerChurn
FIELDS TERMINATED BY ','
ENCLOSED BY '"'
LINES TERMINATED BY '\n'
IGNORE 1 ROWS
(customerID, gender, SeniorCitizen, Partner, Dependents, tenure, PhoneService,
 MultipleLines, InternetService, OnlineSecurity, OnlineBackup, DeviceProtection,
 TechSupport, StreamingTV, StreamingMovies, Contract, PaperlessBilling,
 PaymentMethod, MonthlyCharges, @TotalCharges, Churn)
SET TotalCharges = NULLIF(TRIM(@TotalCharges), '');
```

3. Data Cleaning:

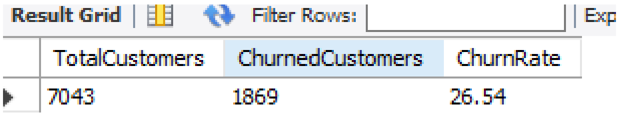
```
-- clean hidden characters

UPDATE TelcoCustomerChurn
SET Churn = TRIM(REPLACE(Churn, CHAR(13), ''));

UPDATE TelcoCustomerChurn
SET Contract = TRIM(REPLACE(Contract, CHAR(13), ''));

UPDATE TelcoCustomerChurn
SET PaymentMethod = TRIM(REPLACE(PaymentMethod, CHAR(13), ''));

UPDATE TelcoCustomerChurn
SET InternetService = TRIM(REPLACE(InternetService, CHAR(13), ''));
```

4. Data Analysis:

Ques-1) Calculate the Total customers, churned customers and churn rate.

```
SELECT
   COUNT(*) AS TotalCustomers,
   SUM(CASE WHEN Churn = 'Yes' THEN 1 ELSE 0 END) AS ChurnedCustomers,
   ROUND(SUM(CASE WHEN Churn = 'Yes' THEN 1 ELSE 0 END) * 100 / COUNT(*), 2) AS ChurnRate
FROM TelcoCustomerChurn;
```

Output:

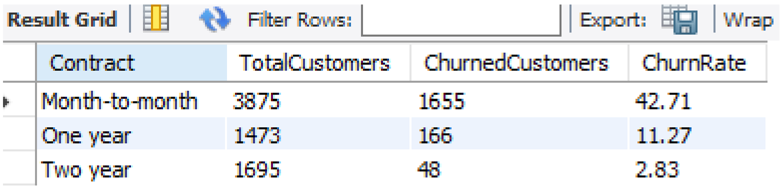


TotalCustomers	ChurnedCustomers	ChurnRate
7043	1869	26.54

Ques-2) Calculate churn by contract type.

```
SELECT
   Contract,
   COUNT(*) AS TotalCustomers,
   SUM(CASE WHEN Churn = 'Yes' THEN 1 ELSE 0 END) AS ChurnedCustomers,
   ROUND(SUM(CASE WHEN Churn = 'Yes' THEN 1 ELSE 0 END) * 100 / COUNT(*), 2) AS ChurnRate
FROM TelcoCustomerChurn
GROUP BY Contract
ORDER BY ChurnRate DESC;
```

Output:

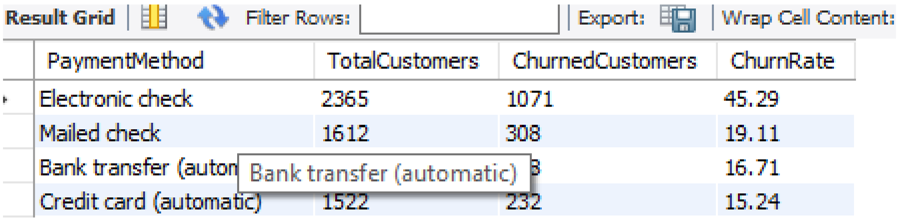


Contract	TotalCustomers	ChurnedCustomers	ChurnRate
Month-to-month	3875	1655	42.71
One year	1473	166	11.27
Two year	1695	48	2.83

Ques-3) Calculate churn by payment method.

```
SELECT

    PaymentMethod,

    COUNT(*) AS TotalCustomers,

    SUM(CASE WHEN Churn = 'Yes' THEN 1 ELSE 0 END) AS ChurnedCustomers,

    ROUND(SUM(CASE WHEN Churn = 'Yes' THEN 1 ELSE 0 END) * 100 / COUNT(*), 2) AS ChurnRate

FROM TelcoCustomerChurn

GROUP BY PaymentMethod

ORDER BY ChurnRate DESC;
```

Output:

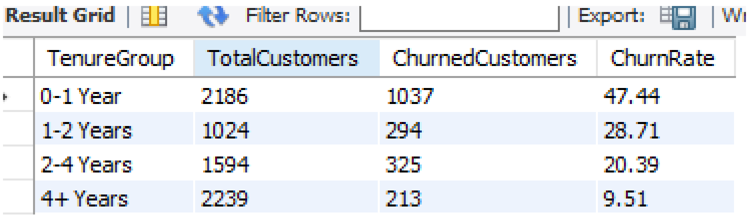


PaymentMethod	TotalCustomers	ChurnedCustomers	ChurnRate
Electronic check	2365	1071	45.29
Mailed check	1612	308	19.11
Bank transfer (auton	[illegible]	3	16.71
Credit card (automatic)	1522	232	15.24

Ques-4) Perform tenure group analysis.

```
SELECT

    CASE

        WHEN tenure <= 12 THEN '0-1 Year'

        WHEN tenure <= 24 THEN '1-2 Years'

        WHEN tenure <= 48 THEN '2-4 Years'

        ELSE '4+ Years'

    END AS TenureGroup,

    COUNT(*) AS TotalCustomers,

    SUM(CASE WHEN Churn = 'Yes' THEN 1 ELSE 0 END) AS ChurnedCustomers,
    ROUND(SUM(CASE WHEN Churn = 'Yes' THEN 1 ELSE 0 END) * 100 / COUNT(*), 2) AS ChurnRate

FROM TelcoCustomerChurn

GROUP BY TenureGroup

ORDER BY ChurnRate DESC;
```

Output:

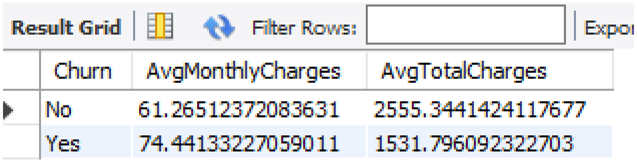


TenureGroup	TotalCustomers	ChurnedCustomers	ChurnRate
0-1 Year	2186	1037	47.44
1-2 Years	1024	294	28.71
2-4 Years	1594	325	20.39
4+ Years	2239	213	9.51

Ques-5) Perform revenue analysis.

```
SELECT

    Churn,

    AVG(MonthlyCharges) AS AvgMonthlyCharges,

    AVG(TotalCharges) AS AvgTotalCharges

FROM TelcoCustomerChurn

GROUP BY Churn;
```

Output:

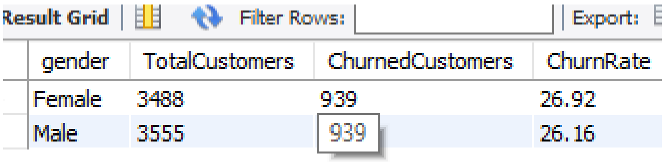


Churn	AvgMonthlyCharges	AvgTotalCharges
No	61.26512372083631	2555.3441424117677
Yes	74.44133227059011	1531.796092322703

Ques-6) Calculate Churn by Gender.

```
SELECT

    gender,

    COUNT(*) AS TotalCustomers,

    SUM(CASE WHEN Churn = 'Yes' THEN 1 ELSE 0 END) AS ChurnedCustomers,

    ROUND(SUM(CASE WHEN Churn = 'Yes' THEN 1 ELSE 0 END) * 100 / COUNT(*), 2) AS ChurnRate

FROM TelcoCustomerChurn

GROUP BY gender

ORDER BY ChurnRate DESC;
```

Output:

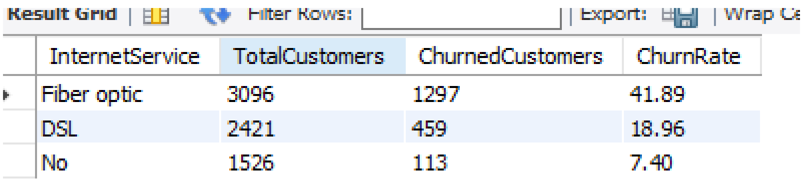


gender	TotalCustomers	ChurnedCustomers	ChurnRate
Female	3488	939	26.92
Male	3555	939	26.16

Ques-7) Calculate churn by internet service type.

```
SELECT

    InternetService,

    COUNT(*) AS TotalCustomers,

    SUM(CASE WHEN Churn = 'Yes' THEN 1 ELSE 0 END) AS ChurnedCustomers,

    ROUND(SUM(CASE WHEN Churn = 'Yes' THEN 1 ELSE 0 END) * 100 / COUNT(*), 2) AS ChurnRate

FROM TelcoCustomerChurn

GROUP BY InternetService

ORDER BY ChurnRate DESC;
```

Output:

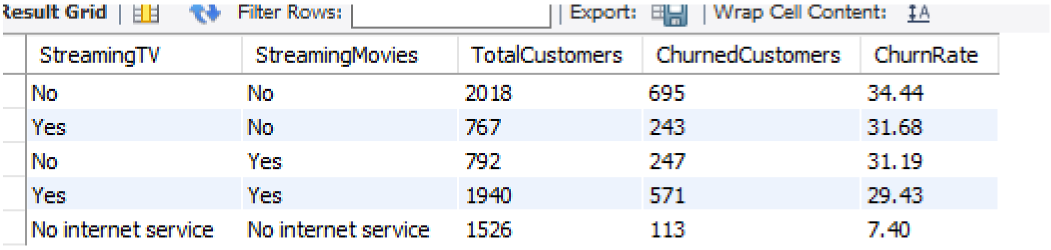


InternetService	TotalCustomers	ChurnedCustomers	ChurnRate
Fiber optic	3096	1297	41.89
DSL	2421	459	18.96
No	1526	113	7.40

Ques-8) Calculate churn by streaming service usage.

```
SELECT

    StreamingTV,

    StreamingMovies,

    COUNT(*) AS TotalCustomers,

    SUM(CASE WHEN Churn = 'Yes' THEN 1 ELSE 0 END) AS ChurnedCustomers,

    ROUND(SUM(CASE WHEN Churn = 'Yes' THEN 1 ELSE 0 END) * 100 / COUNT(*), 2) AS ChurnRate

FROM TelcoCustomerChurn

GROUP BY StreamingTV, StreamingMovies

ORDER BY ChurnRate DESC;
```

Output:

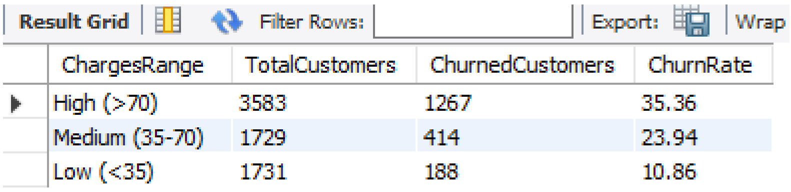


StreamingTV	StreamingMovies	TotalCustomers	ChurnedCustomers	ChurnRate
No	No	2018	695	34.44
Yes	No	767	243	31.68
No	Yes	792	247	31.19
Yes	Yes	1940	571	29.43
No internet service	No internet service	1526	113	7.40

Ques-9) Calculate churn by monthly charges ranges.

```
SELECT
  CASE
    WHEN MonthlyCharges < 35 THEN 'Low (<35)'
    WHEN MonthlyCharges BETWEEN 35 AND 70 THEN 'Medium (35-70)'
    ELSE 'High (>70)'
  END AS ChargesRange,
  COUNT(*) AS TotalCustomers,
  SUM(CASE WHEN Churn = 'Yes' THEN 1 ELSE 0 END) AS ChurnedCustomers,
  ROUND(SUM(CASE WHEN Churn = 'Yes' THEN 1 ELSE 0 END) * 100 / COUNT(*), 2) AS ChurnRate
FROM TelcoCustomerChurn
GROUP BY ChargesRange
ORDER BY ChurnRate DESC;
```

Output:

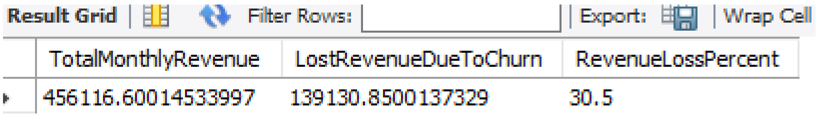


ChargesRange	TotalCustomers	ChurnedCustomers	ChurnRate
High (>70)	3583	1267	35.36
Medium (35-70)	1729	414	23.94
Low (<35)	1731	188	10.86

Ques-10) Calculate revenue loss due to churn.

```
SELECT
  SUM(MonthlyCharges) AS TotalMonthlyRevenue,
  SUM(CASE WHEN Churn = 'Yes' THEN MonthlyCharges ELSE 0 END) AS LostRevenueDueToChurn,
  ROUND(SUM(CASE WHEN Churn = 'Yes' THEN MonthlyCharges ELSE 0 END) * 100 / SUM(MonthlyCharges),2)
AS RevenueLossPercent
FROM TelcoCustomerChurn;
```

Output:

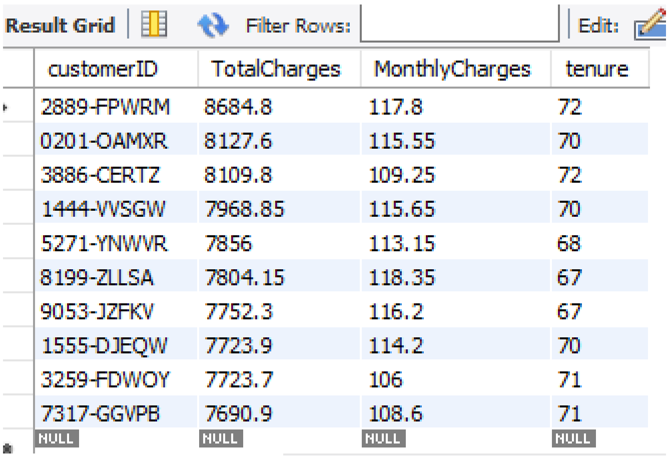


TotalMonthlyRevenue	LostRevenueDueToChurn	RevenueLossPercent
456116.60014533997	139130.8500137329	30.5

Ques-11) Top 10 customers with highest total charges who churned.

```
select
    customerID,
    TotalCharges,
    MonthlyCharges,
    tenure
from TelcoCustomerChurn
where Churn = 'Yes'
order by TotalCharges desc
limit 10;
```

Output:



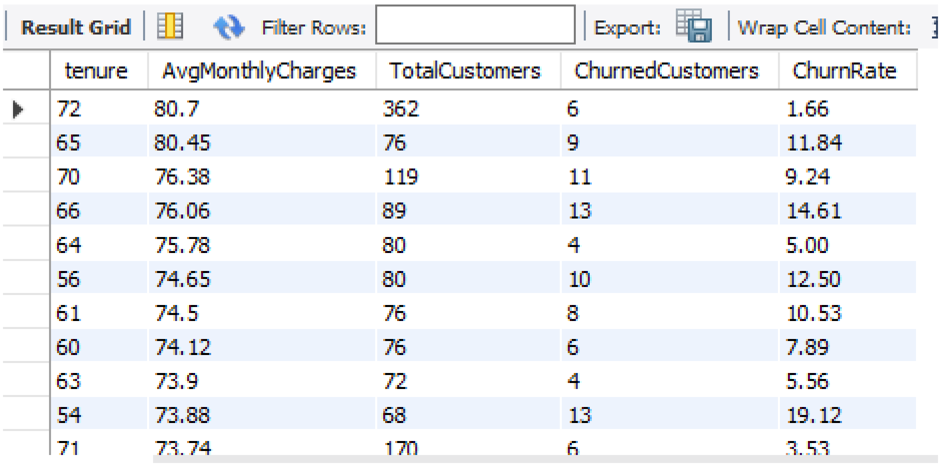
customerID	TotalCharges	MonthlyCharges	tenure
2889-FPWRM	8684.8	117.8	72
0201-OAMXR	8127.6	115.55	70
3886-CERTZ	8109.8	109.25	72
1444-VVSGW	7968.85	115.65	70
5271-YNWVR	7856	113.15	68
8199-ZLLSA	7804.15	118.35	67
9053-JZFKV	7752.3	116.2	67
1555-DJEQW	7723.9	114.2	70
3259-FDWOY	7723.7	106	71
7317-GGVPB	7690.9	108.6	71
NULL	NULL	NULL	NULL

Ques-12) Correlation between tenure and monthly charges.

```
select
    tenure,
    round(avg(MonthlyCharges), 2) as AvgMonthlyCharges,
    count(*) as TotalCustomers,
    sum(case when Churn = 'Yes' then 1 else 0 end) as ChurnedCustomers,
    round(sum(case when Churn = 'Yes' then 1 else 0 end) * 100 / count(*), 2) as ChurnRate
```

```
from TelcoCustomerChurn

group by tenure

order by AvgMonthlyCharges desc;
```

Output:



tenure	AvgMonthlyCharges	TotalCustomers	ChurnedCustomers	ChurnRate
72	80.7	362	6	1.66
65	80.45	76	9	11.84
70	76.38	119	11	9.24
66	76.06	89	13	14.61
64	75.78	80	4	5.00
56	74.65	80	10	12.50
61	74.5	76	8	10.53
60	74.12	76	6	7.89
63	73.9	72	4	5.56
54	73.88	68	13	19.12
71	73.74	170	6	3.53

Insights from SQL:

- Customers with **month-to-month contracts** are more likely to churn.
- Customers paying via **electronic checks** have higher churn rates.
- Customers with **longer tenure** have lower churn probabilities.

Python Phase (Machine Learning):

Libraries Used:

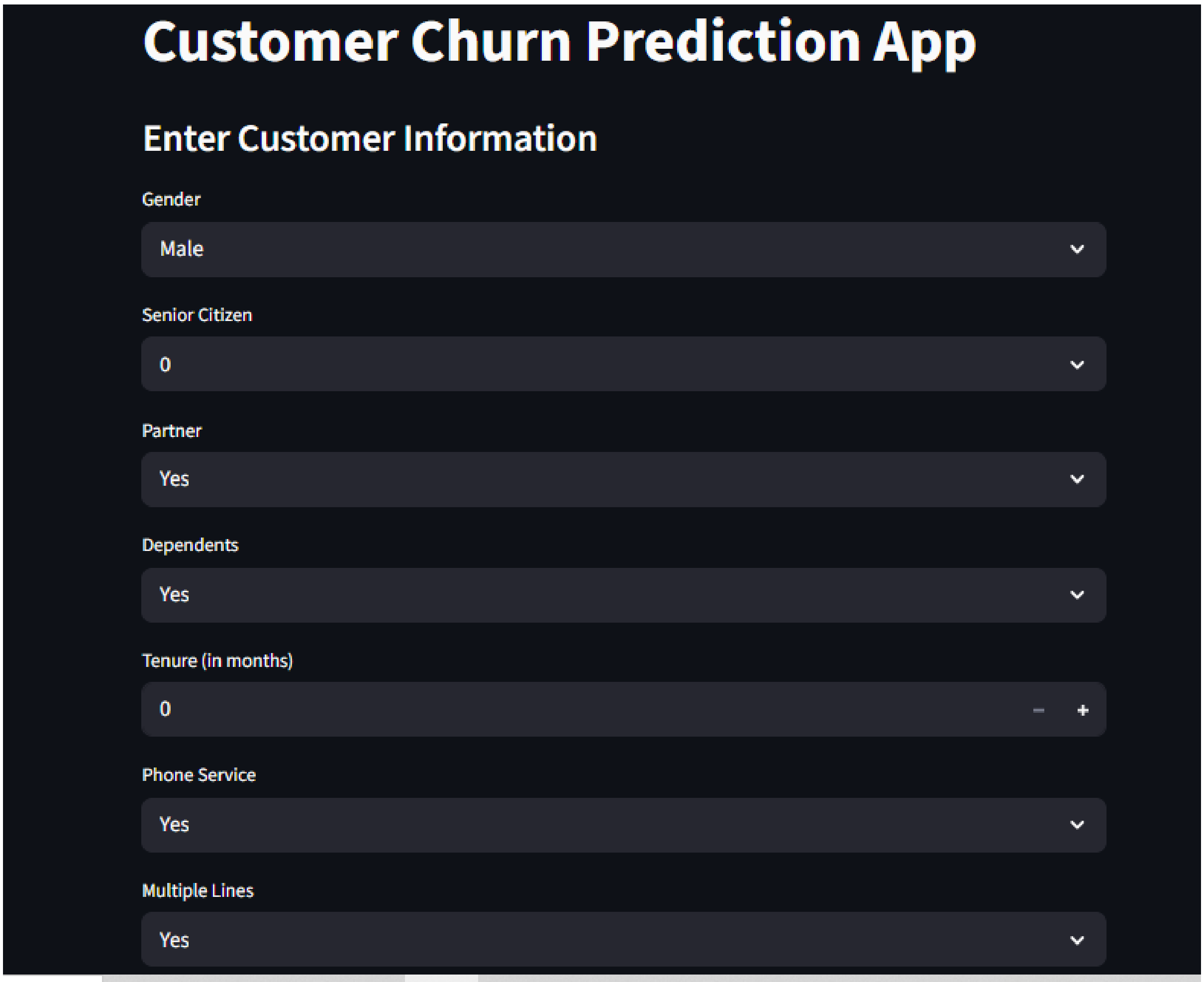
- pandas
- numpy
- scikit-learn
- xgboost
- joblib
- streamlit (for deployment)

Steps Followed:

1. **Data Preprocessing:**
   - Handled categorical variables using **One-Hot Encoding**.
   - Scaled numeric columns (`tenure`, `MonthlyCharges`, `TotalCharges`) if required.
2. **Model Building:**
   - Models tried: Logistic Regression, Random Forest, XGBoost.
   - **XGBoostClassifier** provided the best performance.
3. **Model Training & Testing:**
   - Split dataset into training (80%) and testing (20%).
   - Performed hyperparameter tuning.
4. **Model Evaluation:**
   - Evaluated using **accuracy, precision, recall, F1-score, and AUC-ROC**.
   - **XGBoost achieved around 82% accuracy.**
5. **Model Deployment:**
   - Saved the trained model using `joblib`.
   - Developed a web app using **Streamlit**.
   - Deployed temporarily via **ngrok** or **LocalTunnel** from Google Colab.

Deployment Approach:

- Built an interactive **Streamlit web app**.
- Allowed users to input customer attributes and receive churn predictions.
- Deployed temporarily using **ngrok** or **LocalTunnel** from Google Colab for demonstration purposes.



Customer Churn Prediction App

Enter Customer Information

Gender: Male

Senior Citizen: 0

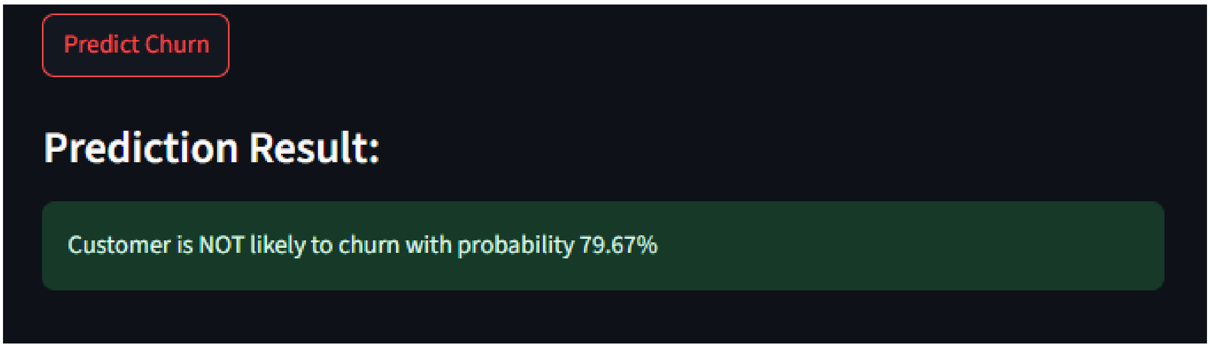
Partner: Yes

Dependents: Yes

Tenure (in months): 0

Phone Service: Yes

Multiple Lines: Yes



Prediction Result:

Customer is NOT likely to churn with probability 79.67%

Conclusion:

- The machine learning model can predict customer churn with reasonable accuracy.
- SQL was effectively used for data cleaning, validation, and basic analysis.
- The deployment via Streamlit provides an accessible interface for stakeholders to use the prediction model.