NAME: DIPIN ROKA

CLASS: BCA - A

ROLL NO: 2411021240007

SUBJECT: INTRODUCTION TO DATA SCIENCE

GITHUB LINK: https://github.com/DIPINROKA10/IDS-INTRODUCTION-TO-DATASCIENCE-

Assignment 1 Part 1: Theoretical Understanding

1. DEFINE DATA SCIENCE: WHAT IS DATA SCIENCE? DISCUSS ITS KEY COMPONENTS AND THE CRISP-DM PROCESS.

Ans Data Science is the study of data to extract meaningful insights using scientific methods, algorithms, and technology. It combines statistics, machine learning, and domain expertise to analyse complex data for decision-making.

Key Components of Data Science

- **Data Collection** – Gathering raw data from various sources.
- **Data Cleaning –** Removing errors and inconsistencies.
- **Exploratory Data Analysis (EDA)** – Identifying patterns and trends.
- **Feature Engineering** – Enhancing data for better model performance.
- **Modelling**- Applying machine learning to build prediction models.
- **Evaluation-** Measuring the accuracy of the model.
- **Deployment-** Application of the model in real time
- CRISP-DM Process
- CRISP-DM, or Cross-Industry Standard Process for Data Mining, is an approach to managing data science projects in a structured way:

Business Understanding- Defines the objectives of the project.

Data Understanding- Exploring patterns in data.

Data Preparation- Cleaning and transformation of data

Modelling- Applying machine learning techniques.

Evaluation- Measuring the performance of the model.

Deployment- Usage of the model in real applications.

This process ensures efficiency and reliability in data-driven solutions.

2.Explain how the CRISP-DM framework is applied in solving real-world problems (e.g., predicting customer churn or recommending movies).

Ans Applying the CRISP-DM Framework to Real-World Problems

The CRISP-DM framework is a systematic way to solve problems in data science. Below is how it can be applied to two real-world scenarios: predicting customer churn and recommending movies.

1. Predicting Customer Churn

Customer churn prediction will help a business identify potential customers to leave and take proactive steps in retaining them.

Business Understanding:

- Define objective: Identify customers at risk of churning.
- Understand business requirements and major churn drivers (e.g., service issues, pricing).
- Data Understanding:
- Collect customer information like demographics, transaction history, and service usage.
- Analyse trends such as the number of complaints and inactive periods.
- Data Preparation:
- Clean missing or inconsistent data.
- Convert categorical variables like subscription plans into numerical form.
- Modelling:
- Use machine learning models like logistic regression, decision trees, or neural networks to classify churn risk.
- Evaluation:

Evaluate the model's accuracy using precision, recall, and F1-score.

Compare various models to choose the best one.

Deployment:

Implement the model in a CRM system to give churn alerts.

Use the insights to create retention strategies, such as personalized offers.

2. Movie Recommendation System

Recommender systems help platforms like Netflix and Amazon suggest content based on user preferences.

Business Understanding:

Define the goal: Increase user engagement by providing personalized movie recommendations.

Understand user behaviour and preferences.

Data Understanding:

- Gather ratings, watch history, and genre preferences of the users.
- Determine the trends in movie popularity and user interaction.
- Data Preprocessing
- Remove missing ratings and normalize formats.
- Convert data into a structured format, such as a user-movie matrix.
- Modelling
- Apply collaborative filtering or content-based filtering algorithms.
- Train the model on past interactions to predict future preferences.
- Evaluation
- Measure the performance using metrics like Mean Absolute Error (MAE) and precision-recall.
- Optimize algorithms for better recommendations.
- Deployment Integrate the system with the streaming platform.
- Always evolve recommendations based on user activity.
- By using the CRISP-DM approach, businesses can construct effective data-driven solutions that drive customer experience.

2. Case Study Questions: From the case studies in the "Module 1 Case Studies" file, answer the following:

What is the main business objective of the Netflix Recommendation System?

Ans The main business objective of the Netflix Recommendation System is to increase user engagement and retention by providing personalized content suggestions. Based on user preferences, watch history, and behaviour patterns, Netflix aims to recommend movies and TV shows in order to:

>>Reduce search time for enhancing user satisfaction.

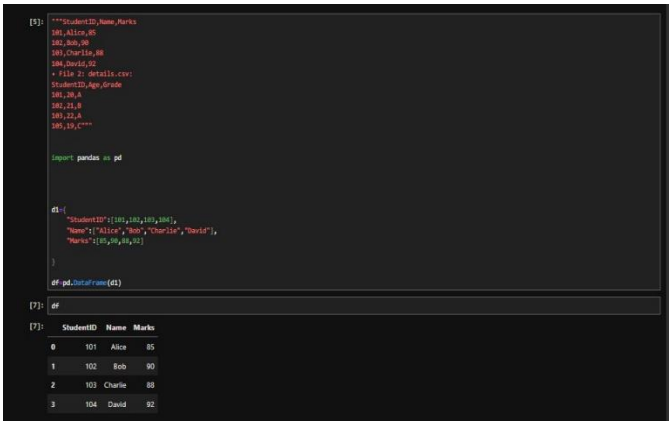
>>Improve customer retention by making viewers engage.

>>Optimise content discovery and increase watch time.

>>Reduce churn by ensuring that users like what they're seeing.

The recommendation system of Netflix ensures that the company remains competitive by delivering an extremely personalized viewing experience.

and decision-making.

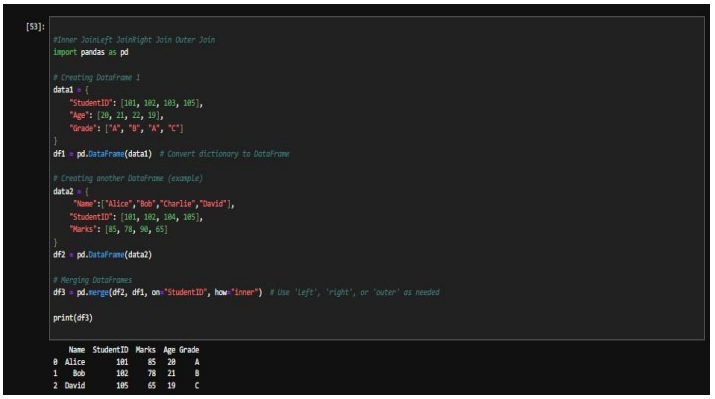


```
[5]: """StudentID,Name,Marks
101,Alice,85
102,Bob,90
103,Charlie,88
104,David,92
+ File 2: details.csv:
StudentID,Age,Grade
101,20,A
102,21,B
103,22,A
105,19,C"""

import pandas as pd

d1={
    "StudentID":[101,102,103,104],
    "Name":["Alice","Bob","Charlie","David"],
    "Marks":[85,90,88,92]
}
df=pd.DataFrame(d1)
```

[7]:
```
df
```

[7]:

	StudentID	Name	Marks
0	101	Alice	85
1	102	Bob	90
2	103	Charlie	88
3	104	David	92



[53]:
```
#Inner JoinLeft JoinRight Join Outer Join
import pandas as pd

# Creating DataFrame 1
data1 = {
    "StudentID": [101, 102, 103, 105],
    "Age": [20, 21, 22, 19],
    "Grade": ["A", "B", "A", "C"]
}
df1 = pd.DataFrame(data1)  # Convert dictionary to DataFrame

# Creating another DataFrame (example)
data2 = {
    "Name":["Alice","Bob","Charlie","David"],
    "StudentID": [101, 102, 104, 105],
    "Marks": [85, 78, 90, 65]
}
df2 = pd.DataFrame(data2)

# Merging DataFrames
df3 = pd.merge(df2, df1, on="StudentID", how="inner")  # Use 'left', 'right', or 'outer' as needed

print(df3)
```

```
    Name  StudentID  Marks  Age Grade
0  Alice        101     85   20     A
1    Bob        102     78   21     B
2  David        105     65   19     C
```

[23]:
```
#Left join
df3 = pd.merge(df2, df1, on="StudentID", how="left")  # Use 'left', 'right', or 'outer' as needed
df3
```

[23]:

	Name	StudentID	Marks	Age	Grade
0	Alice	101	85	20.0	A
1	Bob	102	78	21.0	B
2	Charlie	104	90	NaN	NaN
3	David	105	65	19.0	C

[25]:
```
#rightjoin
df3= pd.merge(df2,df1, on="StudentID", how="right")
df3
```

[25]:

	Name	StudentID	Marks	Age	Grade
0	Alice	101	85.0	20	A
1	Bob	102	78.0	21	B
2	NaN	103	NaN	22	A
3	David	105	65.0	19	C

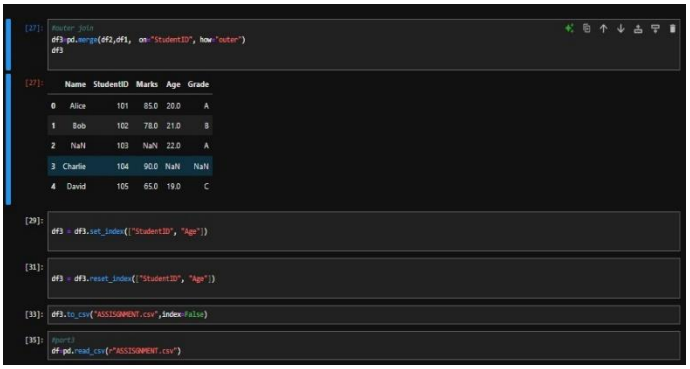
[41]:
```
df.tail()#by default prints the last 5
```

[41]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
763	10	101	76	48	180	32.9	0.171	63	0
764	2	122	70	27	0	36.8	0.340	27	0
765	5	121	72	23	112	26.2	0.245	30	0
766	1	126	60	0	0	30.1	0.349	47	1
767	1	93	70	31	0	30.4	0.315	23	0

[43]:
```
print(df.shape)
```

```
(768, 9)
```



```
[27]: #outer join
df3=pd.merge(df2,df1, on="StudentID", how="outer")
df3
```

[27]:

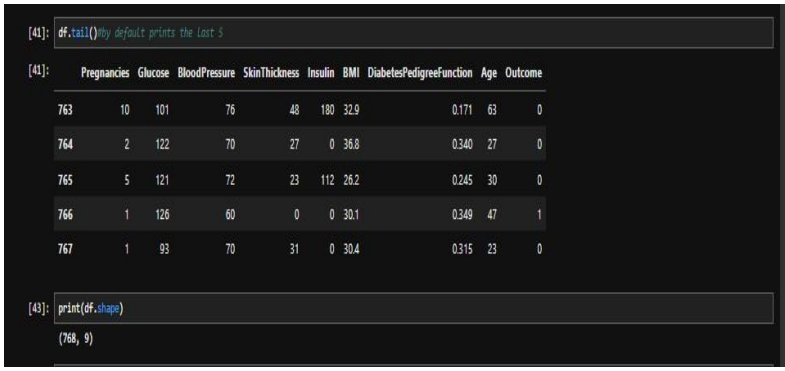
	Name	StudentID	Marks	Age	Grade
0	Alice	101	85.0	20.0	A
1	Bob	102	78.0	21.0	B
2	NaN	103	NaN	22.0	A
3	Charlie	104	90.0	NaN	NaN
4	David	105	65.0	19.0	C

```
[29]: df3 = df3.set_index(["StudentID", "Age"])
```

```
[31]: df3 = df3.reset_index(["StudentID", "Age"])
```

```
[33]: df3.to_csv("ASSISGNMENT.csv",index=False)
```

```
[35]: #part1
df=pd.read_csv("ASSISGNMENT.csv")
```



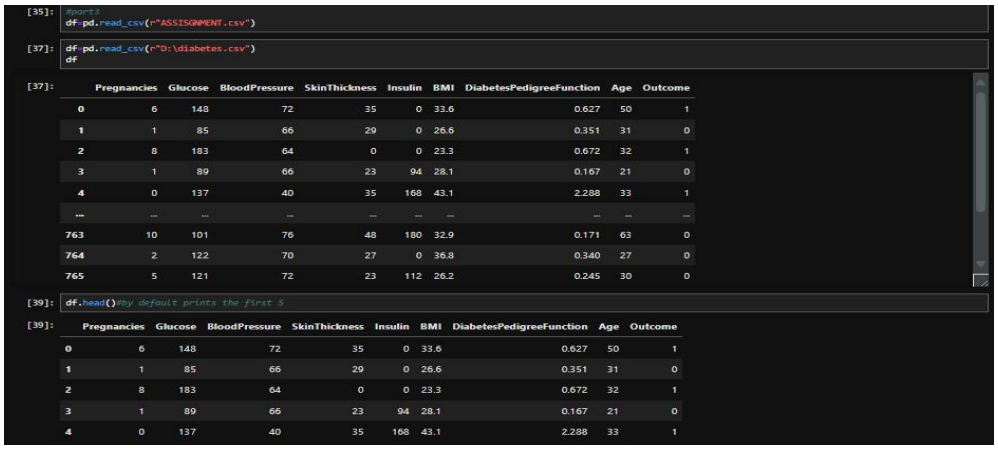
```
[41]: df.tail()#by default prints the last 5
```

[41]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
763	10	101	76	48	180	32.9	0.171	63	0
764	2	122	70	27	0	36.8	0.340	27	0
765	5	121	72	23	112	26.2	0.245	30	0
766	1	126	60	0	0	30.1	0.349	47	1
767	1	93	70	31	0	30.4	0.315	23	0

```
[43]: print(df.shape)
```

(768, 9)



```
[35]: #part1
df=pd.read_csv("ASSISGNMENT.csv")
```

```
[37]: df=pd.read_csv(r"D:\diabetes.csv")
df
```

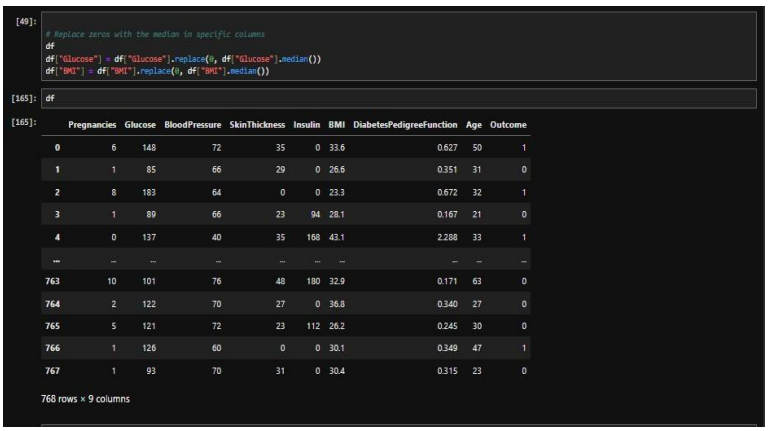
[37]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	6	148	72	35	0	33.6	0.627	50	1
1	1	85	66	29	0	26.6	0.351	31	0
2	8	183	64	0	0	23.3	0.672	32	1
3	1	89	66	23	94	28.1	0.167	21	0
4	0	137	40	35	168	43.1	2.288	33	1
...	...	...	...	...	...	...	...	...	...
763	10	101	76	48	180	32.9	0.171	63	0
764	2	122	70	27	0	36.8	0.340	27	0
765	5	121	72	23	112	26.2	0.245	30	0

```
[39]: df.head()#by default prints the first 5
```

[39]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	6	148	72	35	0	33.6	0.627	50	1
1	1	85	66	29	0	26.6	0.351	31	0
2	8	183	64	0	0	23.3	0.672	32	1
3	1	89	66	23	94	28.1	0.167	21	0
4	0	137	40	35	168	43.1	2.288	33	1



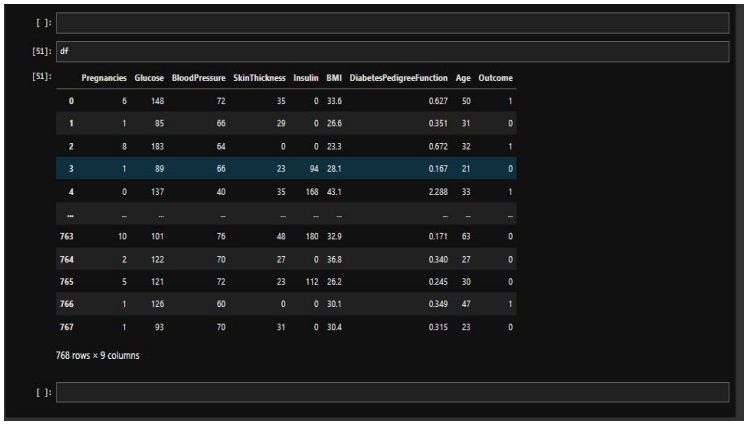
```
[49]: # Replace zeros with the median in specific columns
df
df["Glucose"] = df["Glucose"].replace(0, df["Glucose"].median())
df["BMI"] = df["BMI"].replace(0, df["BMI"].median())
```

```
[165]: df
```

[165]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	6	148	72	35	0	33.6	0.627	50	1
1	1	85	66	29	0	26.6	0.351	31	0
2	8	183	64	0	0	23.3	0.672	32	1
3	1	89	66	23	94	28.1	0.167	21	0
4	0	137	40	35	168	43.1	2.288	33	1
...	...	...	...	...	...	...	...	...	...
763	10	101	76	48	180	32.9	0.171	63	0
764	2	122	70	27	0	36.8	0.340	27	0
765	5	121	72	23	112	26.2	0.245	30	0
766	1	126	60	0	0	30.1	0.349	47	1
767	1	93	70	31	0	30.4	0.315	23	0

768 rows × 9 columns



```
[ ]:
```

```
[51]: df
```

[51]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	6	148	72	35	0	33.6	0.627	50	1
1	1	85	66	29	0	26.6	0.351	31	0
2	8	183	64	0	0	23.3	0.672	32	1
3	1	89	66	23	94	28.1	0.167	21	0
4	0	137	40	35	168	43.1	2.288	33	1
...	...	...	...	...	...	...	...	...	...
763	10	101	76	48	180	32.9	0.171	63	0
764	2	122	70	27	0	36.8	0.340	27	0
765	5	121	72	23	112	26.2	0.245	30	0
766	1	126	60	0	0	30.1	0.349	47	1
767	1	93	70	31	0	30.4	0.315	23	0

768 rows × 9 columns

```
[ ]:
```