Introduction to Data Science

Assignment 1

Name: Anuska Ghosh

Course: BCA (Sec – A)

Registration Number: 2411021240012

GitHub Repository Link: https://github.com/anuskaghosh17/IDS_Python

Part 1: Theoretical Understanding

1.Define Data Science:

Q.What is Data Science? Discuss its key components and the CRISP-DM process.

Ans. Data Science is an interdisciplinary field that uses mathematics, statistics, computer science, and domain knowledge to extract meaningful insights from data.

It is all about asking the right questions and using data to find answers.

Key Components:

a. Data Collection: Gathering raw data from various sources.

b. Data Processing: Cleaning and organizing the data.

c. Data Analysis: Using statistical and computational methods to derive insights.

d. Visualization: Presenting results in an understandable format (e.g., graphs,

dashboards).

e. Decision-Making: Applying insights to solve real-world problems.

The CRISP-DM Process:

A structured approach to solving data science problems.

1. Business Understanding: Define objectives and questions.

2. Data Understanding: Explore the data to understand its structure.

3. Data Preparation: Clean, transform, and organize the data.

4. Modeling: Apply algorithms to analyze and predict.

5. Evaluation: Check if the model answers the problem.

6. Deployment: Implement the solution in real-world scenarios.

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

Ans. **Predicting Customer Churn in Telecom:-**

Problem Statement:

How can a telecom company predict which customers are likely to stop using their services?

Dataset:

Source: Telco Customer Churn Dataset. [https://www.kaggle.com/datasets/blastchar/telco-customer-churn]

Columns:

- Customer ID

- Demographics: Gender, age group, etc.

- Service details: Internet service, contract type, monthly charges.

- Churn (Yes/No).

CRISP-DM Process:

1. *Business Understanding:*

- Goal: Reduce churn rate by identifying at-risk customers.

- Impact: Increase revenue by targeting retention efforts.

2. *Data Understanding:*

- Analyze churn rate across demographics.

- Understand correlations between service features and churn.

3. *Data Preparation:*

- Handle missing values (e.g., impute missing charges).

- Convert categorical variables (e.g., gender, contract type) to numerical data.

- Scale numerical features like monthly charges.

4. *Modeling:*

- Train a classification model (e.g., Logistic Regression, Random Forest) to predict

churn.

- Use features like contract type, monthly charges, and tenure.

5. *Evaluation:*

- Use metrics like Accuracy, Precision, Recall, and F1 Score.

- Evaluate the model on a confusion matrix to understand false positives and

negatives.

6. *Deployment:*

- Provide alerts for high-risk customers to customer service teams.

- Implement proactive offers and discounts to retain customers.

Netflix Recommendation System:-

Problem Statement:

How can Netflix recommend personalized movies or TV shows to users based on their preferences?

Dataset:

Source: MovieLens Dataset (Free public dataset). [https://grouplens.org/datasets/movielens/]

Columns:

- User ID

- Movie ID

- Rating (1-5)

- Timestamp

- Movie metadata (title, genres, release year).

CRISP-DM Process:

1. *Business Understanding:*

- Goal: Improve user engagement by suggesting content they're likely to enjoy.

- Impact: Increased user satisfaction and retention.

2. *Data Understanding:*

- Explore the dataset: Number of users, movies, and ratings.

- Analyze distribution of ratings and popular genres.

3. *Data Preparation:*

- Handle missing data in movie metadata.

- Transform timestamp into human-readable format.

- One-hot encode genres for analysis.

4. *Modeling:*

- Use Collaborative Filtering to predict user preferences:

(I) Find similar users and suggest movies they liked.

(II) Train a recommendation model (e.g., Singular Value Decomposition).

- Alternative: Content-based filtering using movie metadata.

5. *Evaluation:*

- Split data into training and test sets.

- Use metrics like Root Mean Square Error (RMSE) to measure model accuracy.

6. *Deployment:*

- Integrate the model into a recommendation engine.

- Provide real-time recommendations on the Netflix platform.

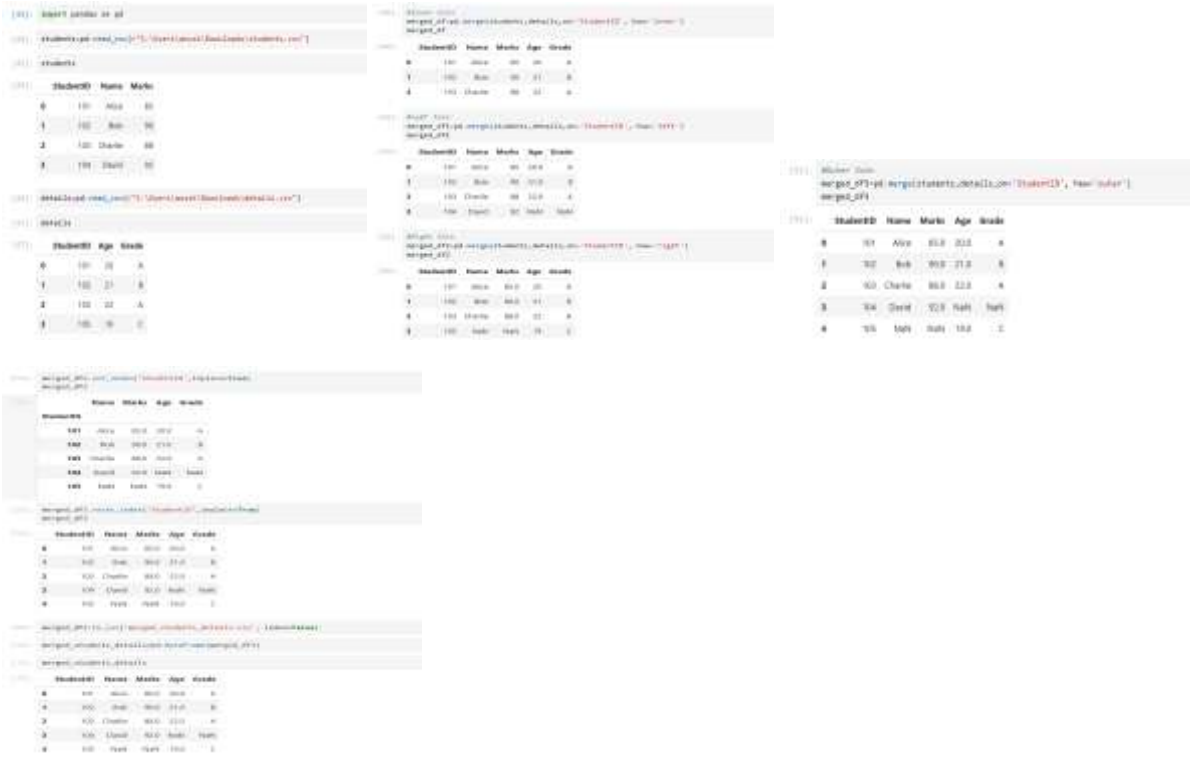
2. Case Study Questions:

- From the case studies in the "Module 1 Case Studies" file, answer the following:

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

Ans. : The main business objective of the Netflix Recommendation System is to improve user engagement by suggesting content they're likely to enjoy. This will increase the user satisfaction and retention.

Part 2: Data Manipulation and Joins



Types of Joins and difference between each of them:

a. Inner Join: Only take the matching records from both students and details files.
b. Left Join: All records from the students file (left) and matching records from the details file (right).
c. Right Join: All records from the details file (right) and matching records from the students file (left).
d. Outer Join: Take all the records from both students and details files.

Merged CSV File: http://localhost:8888/lab/tree/IDS/merged_students_details.csv

	StudentID	Name	Marks	Age	Grade
1	101	Alice	85.0	20.0	A
2	102	Bob	90.0	21.0	B
3	103	Charlie	88.0	22.0	A
4	104	David	92.0		
5	105			19.0	C

Part 3: Exploratory Data Analysis (EDA)

```
import pandas as pd
```

```
df=pd.read_csv("C:\Users\pavan\Downloads\diabetes.csv")
```

```
df
```

	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

```
df.head(5)
```

	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

```
df.shape
```

```
(768, 9)
```

```
df.describe()
```

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
count	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000
mean	3.845052	120.894531	69.105469	20.536458	79.799479	31.992578	0.471876	33.240885	0.348958
std	3.369578	31.972618	19.355807	15.952218	115.244002	7.884160	0.331329	11.760232	0.476951
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.078000	21.000000	0.000000
25%	1.000000	99.000000	62.000000	0.000000	0.000000	27.300000	0.243750	24.000000	0.000000
50%	3.000000	117.000000	72.000000	23.000000	30.500000	32.000000	0.372500	29.000000	0.000000
75%	6.000000	140.250000	80.000000	32.000000	127.250000	36.600000	0.626250	41.000000	1.000000
max	17.000000	199.000000	122.000000	99.000000	846.000000	67.100000	2.420000	81.000000	1.000000

```
df.isna()
```

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	False	False	False	False	False	False	False	False	False
1	False	False	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False	False	False
3	False	False	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False	False	False
...	...	...	...	...	...	...	...	...	...
763	False	False	False	False	False	False	False	False	False
764	False	False	False	False	False	False	False	False	False
765	False	False	False	False	False	False	False	False	False
766	False	False	False	False	False	False	False	False	False
767	False	False	False	False	False	False	False	False	False

768 rows × 9 columns

```
BMI_median = df['BMI'].median()
```

```
BMI_median
```

```
32.0
```

```
df['BMI'] = df['BMI'].fillna(BMI_median)
```

```
df
```

	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