NAME :- ROHITH BALAN

COURSE :- B.C.A.-A

REG.NO :- 2411021240022

GITHUB LINK :-rohithbalan1907/IDS-ASSIGNMENT

1.What is Data Science?

Data science is a field that focuses on extracting insights and knowledge from structured and unstructured data using techniques such as statistics, machine learning, and data analysis. It involves multiple steps, including data collection, cleaning, exploratory analysis, machine learning, and visualization. Businesses use data science for decision-making, healthcare benefits from it for disease prediction, and finance utilizes it for fraud detection. With the increasing availability of big data, cloud computing, and AI, data science continues to shape industries and drive innovation.

Understanding Data Science

Data science is a discipline that utilizes techniques such as statistics, machine learning, and data analysis to extract meaningful insights from data. It

plays a crucial role in decision-making and problem-solving across industries like business, healthcare, finance, and entertainment. In essence, data science involves gathering, cleaning, analyzing, and applying data to drive informed decisions. With the vast amounts of data generated daily from sources like social media, transactions, and sensors, data science helps transform raw information into actionable knowledge, improving efficiency and customer satisfaction.

Core Components of Data Science

To extract useful insights, data science follows several key steps:

1. **Data Collection** – Gathering raw data from various sources such as sensors, websites, customer interactions, surveys, and APIs, ensuring data quality for accuracy.
2. **Data Cleaning & Preparation** – Formatting data properly, handling missing values, and eliminating errors to make it analysis-ready.
3. **Exploratory Data Analysis (EDA)** – Identifying trends, patterns, and outliers using visualizations like graphs and charts.

4. **Feature Engineering** – Selecting or creating key variables to improve predictive accuracy.
5. **Model Building** – Developing machine learning models like decision trees and neural networks for predictions.
6. **Evaluation** – Measuring the model's accuracy using metrics such as precision and recall.
7. **Deployment & Monitoring** – Implementing the model in real-world applications and continuously updating it as needed.

CRISP-DM: A Structured Approach to Data Science

The **Cross-Industry Standard Process for Data Mining (CRISP-DM)** provides a systematic framework for handling data challenges through six stages:

1. **Business Understanding** – Defining the problem and objectives.
2. **Data Understanding** – Collecting and analyzing data to assess its quality.
3. **Data Preparation** – Cleaning and formatting data for analysis.
4. **Modeling** – Using machine learning techniques to develop predictive models.

5. **Evaluation** – Testing the model's performance for accuracy.
6. **Deployment** – Applying the model in real-world scenarios.

Real-World Applications

Data science is widely used to enhance business performance and customer experience. For instance, telecom companies analyze customer data to predict and prevent churn. Similarly, platforms like Netflix utilize data science to improve content recommendations.

Netflix's Recommendation System

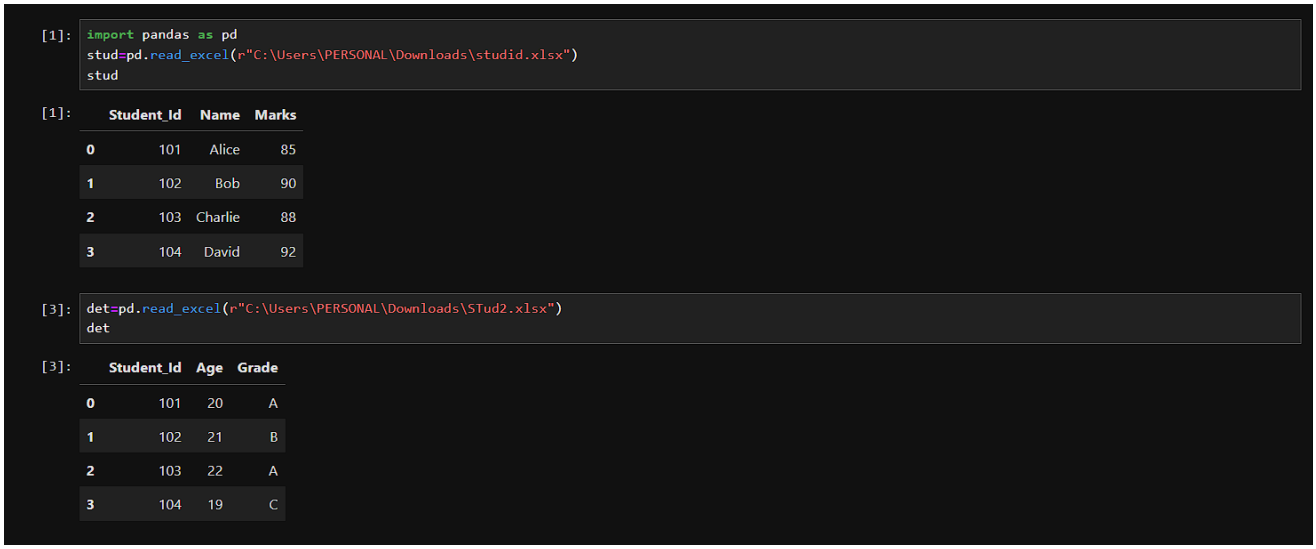
Netflix's recommendation algorithm is designed to keep users engaged by suggesting content tailored to their preferences. Using machine learning, it analyzes user behavior to recommend personalized TV shows and movies.

Key Factors Behind Its Effectiveness:

1. **User Behavior Tracking** – Netflix monitors what users watch, like, and skip.

2. **Personalized Recommendations** – It compares viewing patterns with those of similar users to suggest relevant content.
3. **Increased Watch Time** – Engaging recommendations encourage users to spend more time on the platform.
4. **Simplified Content Discovery** – Instead of searching, users receive ready-to-watch suggestions.

This recommendation system plays a vital role in Netflix's success by boosting engagement, reducing churn rates, and increasing revenue. Without intelligent recommendations, users may struggle to find appealing content, leading them to leave the platform.



```
[1]: import pandas as pd
     stud=pd.read_excel(r"C:\Users\PERSONAL\Downloads\studid.xlsx")
     stud
```

[1]:

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

```
[3]: det=pd.read_excel(r"C:\Users\PERSONAL\Downloads\STud2.xlsx")
     det
```

[3]:

	Student_Id	Age	Grade
0	101	20	A
1	102	21	B
2	103	22	A
3	104	19	C

[4]:
```
innerjoin=pd.merge(stud,det,how='inner',on='Student_Id')
innerjoin
```

[4]:

	Student_Id	Name	Marks	Age	Grade
0	101	Alice	85	20	A
1	102	Bob	90	21	B
2	103	Charlie	88	22	A
3	104	David	92	19	C

[5]:
```
leftjoin=pd.merge(stud,det,how='left',on='Student_Id')
leftjoin
```

[5]:

	Student_Id	Name	Marks	Age	Grade
0	101	Alice	85	20	A
1	102	Bob	90	21	B
2	103	Charlie	88	22	A
3	104	David	92	19	C

[6]:
```
rightjoin=pd.merge(stud,det,how='right',on='Student_Id')
rightjoin
```

[6]:

	Student_Id	Name	Marks	Age	Grade
0	101	Alice	85	20	A
1	102	Bob	90	21	B
2	103	Charlie	88	22	A
3	104	David	92	19	C

[7]:
```
outerjoin=pd.merge(stud,det,how='outer',on='Student_Id')
outerjoin
```

[7]:

	Student_Id	Name	Marks	Age	Grade
0	101	Alice	85	20	A
1	102	Bob	90	21	B
2	103	Charlie	88	22	A
3	104	David	92	19	C

[8]:
```
outerjoin.set_index('Student_Id', inplace=True)
outerjoin
```

[8]:

Student_Id	Name	Marks	Age	Grade
101	Alice	85	20	A
102	Bob	90	21	B
103	Charlie	88	22	A
104	David	92	19	C

[9]:
```
outerjoin.reset_index(inplace=True)
outerjoin
```

[9]:

	Student_Id	Name	Marks	Age	Grade
0	101	Alice	85	20	A
1	102	Bob	90	21	B
2	103	Charlie	88	22	A
3	104	David	92	19	C

[11]:
```
outerjoin.to_csv('outer.csv',index=False)
outerjoin
```

[11]:

	Student_Id	Name	Marks	Age	Grade
0	101	Alice	85	20	A
1	102	Bob	90	21	B
2	103	Charlie	88	22	A
3	104	David	92	19	C

[12]:
```
diabetes=pd.read_csv(r"C:\Users\PERSONAL\Downloads\diabetes (1).csv")
diabetes
```

[12]:

	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

[11]:
```
outerjoin.to_csv('outer.csv',index=False)
outerjoin
```

[11]:

	Student_Id	Name	Marks	Age	Grade
0	101	Alice	85	20	A
1	102	Bob	90	21	B
2	103	Charlie	88	22	A
3	104	David	92	19	C

[12]:
```
diabetes=pd.read_csv(r"C:\Users\PERSONAL\Downloads\diabetes (1).csv")
diabetes
```

[12]:

	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

[13]:
```
diabetes.head()
```

[13]:

	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

[14]:
```
diabetes.shape
```

[14]:
```
(768, 9)
```

[15]:
```
diabetes.describe()
```

[15]:

	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

[16]:
```
diabetes['Pregnancies'] = diabetes['Pregnancies'].replace(0, diabetes['Pregnancies'].median())
diabetes
```

[16]:

	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	3	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

[17]:
```
diabetes['Insulin'] = diabetes['Insulin'].replace(0, diabetes['Insulin'].median())
diabetes
```

[17]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	6	148	72	35	30.5	33.6	0.627	50	1
1	1	85	66	29	30.5	26.6	0.351	31	0
2	8	183	64	0	30.5	23.3	0.672	32	1
3	1	89	66	23	94.0	28.1	0.167	21	0
4	3	137	40	35	168.0	43.1	2.288	33	1
...	...	...	...	...	...	...	...	...	...
763	10	101	76	48	180.0	32.9	0.171	63	0
764	2	122	70	27	30.5	36.8	0.340	27	0
765	5	121	72	23	112.0	26.2	0.245	30	0
766	1	126	60	0	30.5	30.1	0.349	47	1
767	1	93	70	31	30.5	30.4	0.315	23	0

768 rows × 9 columns