Kmit

# SONET

# Data Science
(Level-I)

**Ajeet K. Jain, M. Narsimlu**

(ML TEAM)- SONET, KMIT, Hyderabad

# Session – 29

This session deals with

Data Encoding

Data Science Project Life cycle

Introduction to Case Study

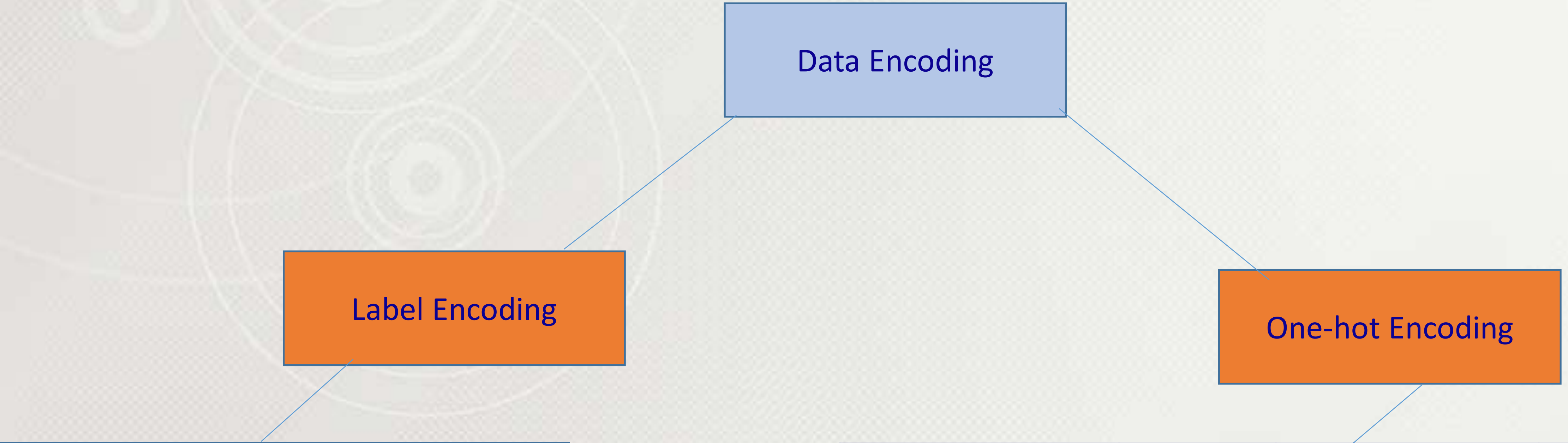
# Data Encoding

- Many machine learning algorithms cannot operate on label data directly. They require all input variables and output variables to be numeric.
- In general, this is mostly a constraint of the efficient implementation of machine learning algorithms rather than hard limitations on the algorithms themselves.
- This means that categorical data must be converted to a numerical form.

# Label Encoding

- Numerical variable will to assign a unique number to each possible  outcome of the variable and replace the variables values with its corresponding number.
- Ex:

| Purpose | loan_purpose_cat |
|---|---|
| Business loan | 0 |
| Buy house | 1 |
| Buy a car | 2 |
| Debt Consolidation | 3 |
| Educational Expenses | 4 |



Data Encoding

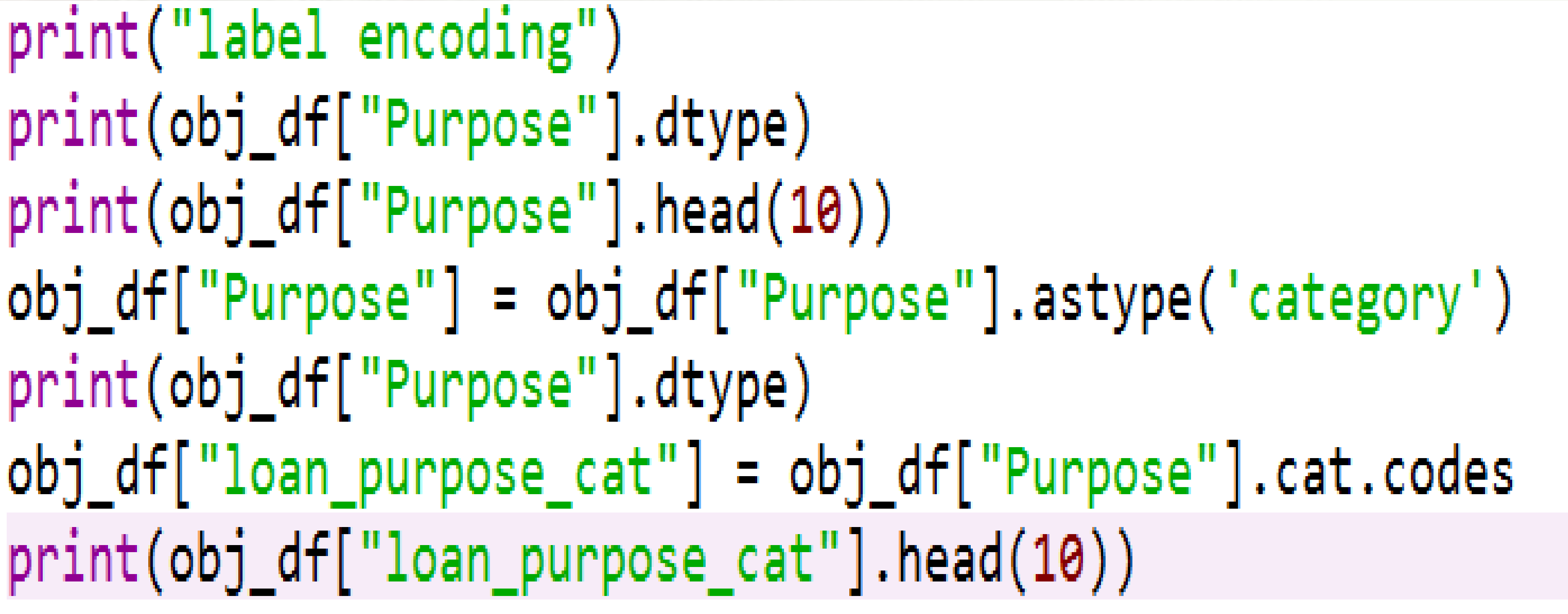
Label Encoding

One-hot Encoding

Ex: Business loan->0,Buy house->1,Buy a car->2 etc..

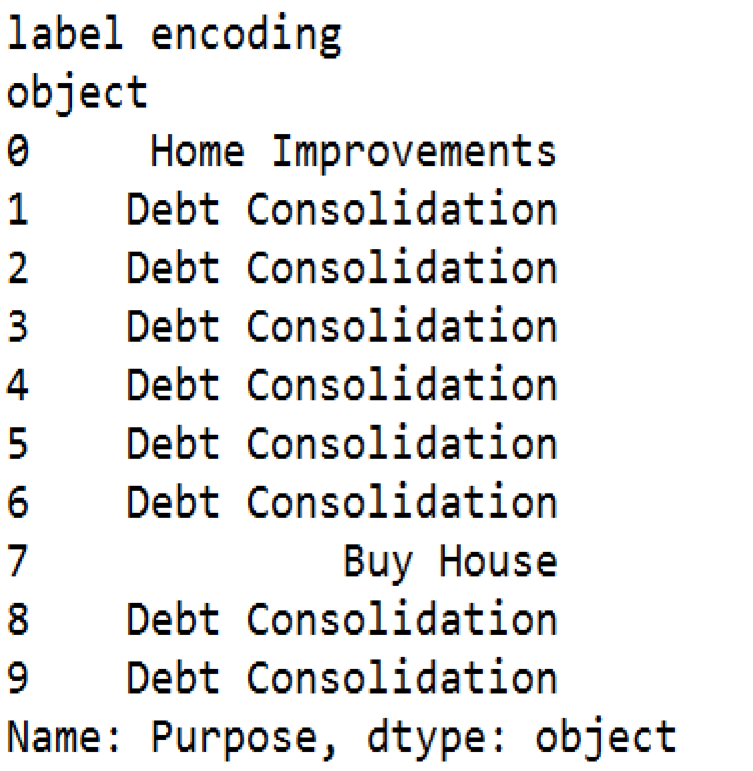
| Home Ownership | H_Rent | H_H_Mort | H_Own Home |
|---|---|---|---|
| Rent | 1 | 0 | 0 |
| Rent | 0 | 1 | 0 |
| Own Home | 1 | 0 | 0 |
| Home Mortgage | 0 | 0 | 1 |
| Home Mortgage | 0 | 0 | 1 |

# Exercise-1

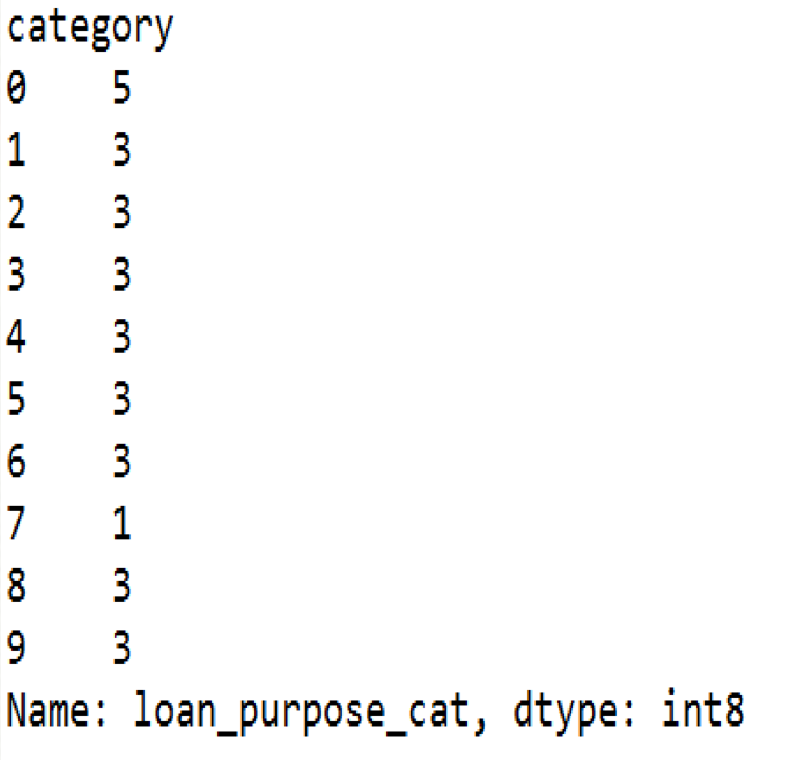


```
print("label encoding")
print(obj_df["Purpose"].dtype)
print(obj_df["Purpose"].head(10))
obj_df["Purpose"] = obj_df["Purpose"].astype('category')
print(obj_df["Purpose"].dtype)
obj_df["loan_purpose_cat"] = obj_df["Purpose"].cat.codes
print(obj_df["loan_purpose_cat"].head(10))
```

# Output



```
label encoding
object
0      Home Improvements
1     Debt Consolidation
2     Debt Consolidation
3     Debt Consolidation
4     Debt Consolidation
5     Debt Consolidation
6     Debt Consolidation
7              Buy House
8     Debt Consolidation
9     Debt Consolidation
Name: Purpose, dtype: object
```



```
category
0    5
1    3
2    3
3    3
4    3
5    3
6    3
7    1
8    3
9    3
Name: loan_purpose_cat, dtype: int8
```

# Binary Encoding

There are two columns of data where the values are words used to represent numbers

Pandas makes it easy for us to directly replace the text values with their numeric equivalent by using replace .

# Exercise-2

Read loan data set and perform following tasks

1.Load “Loan Status”

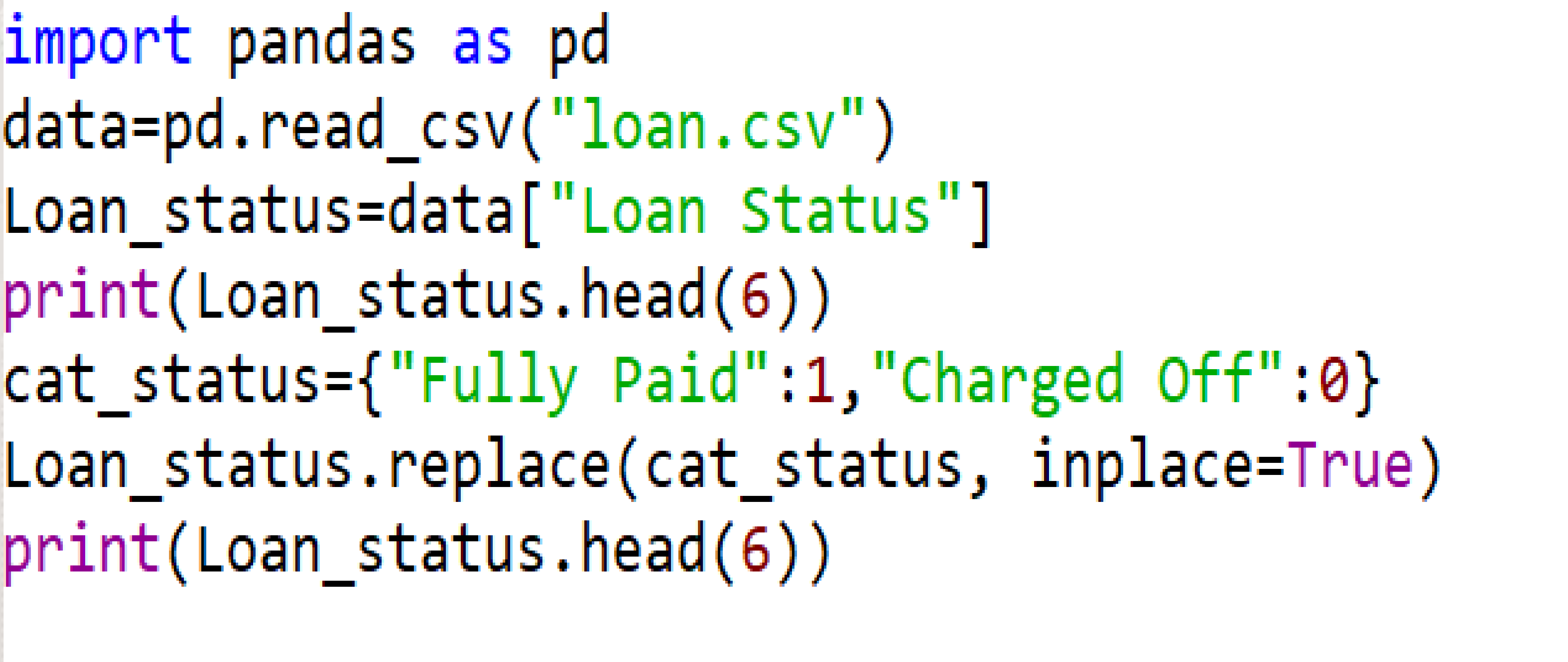
2.Display top 6 records

3.Create a dictionary to replace "Fully Paid“ with 1 and "Charged Off“ with 0

4.Apply replace function to replace the categorical data with numerical

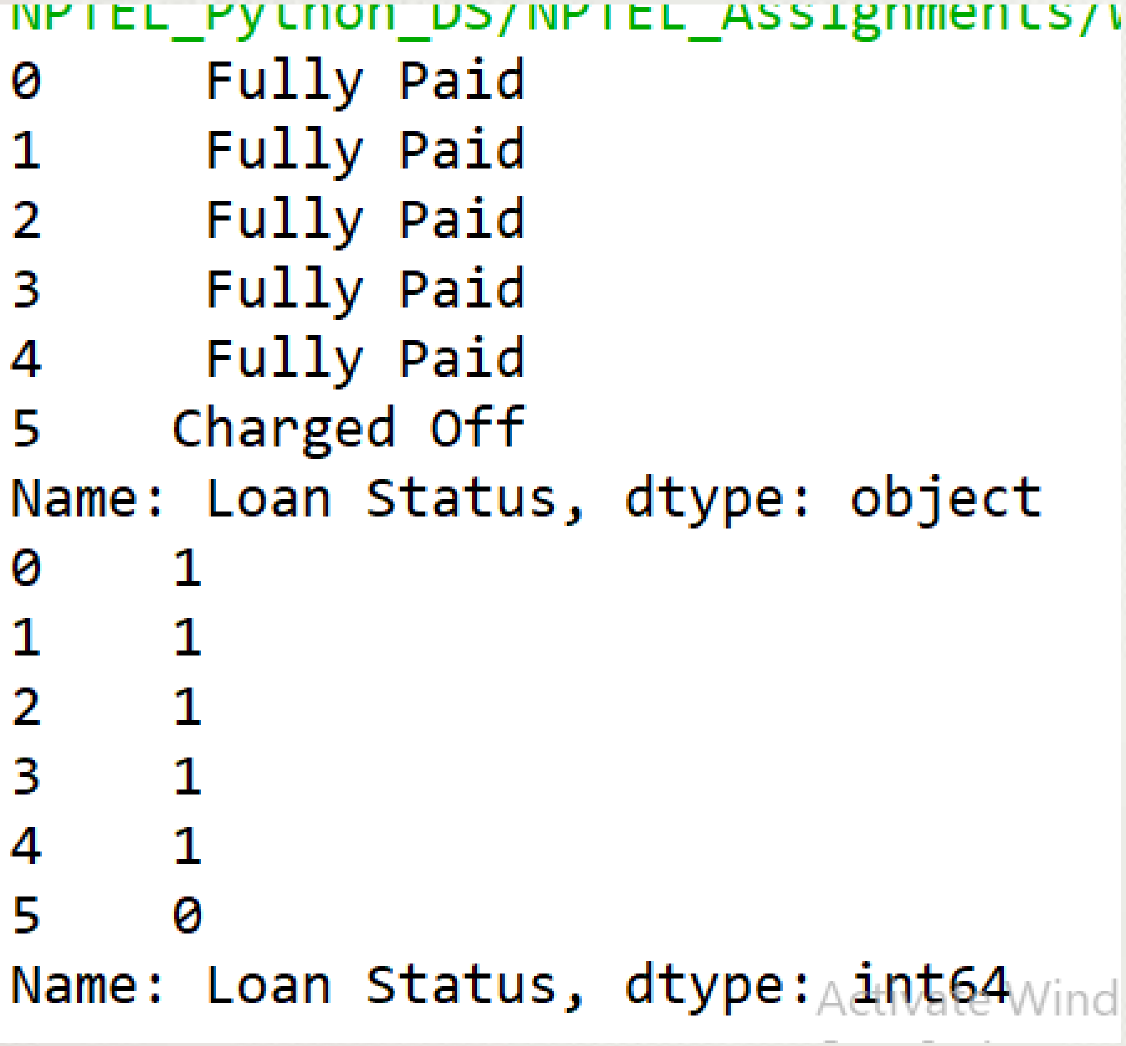
5.Display the top 6 replaced numerical values

# Solution



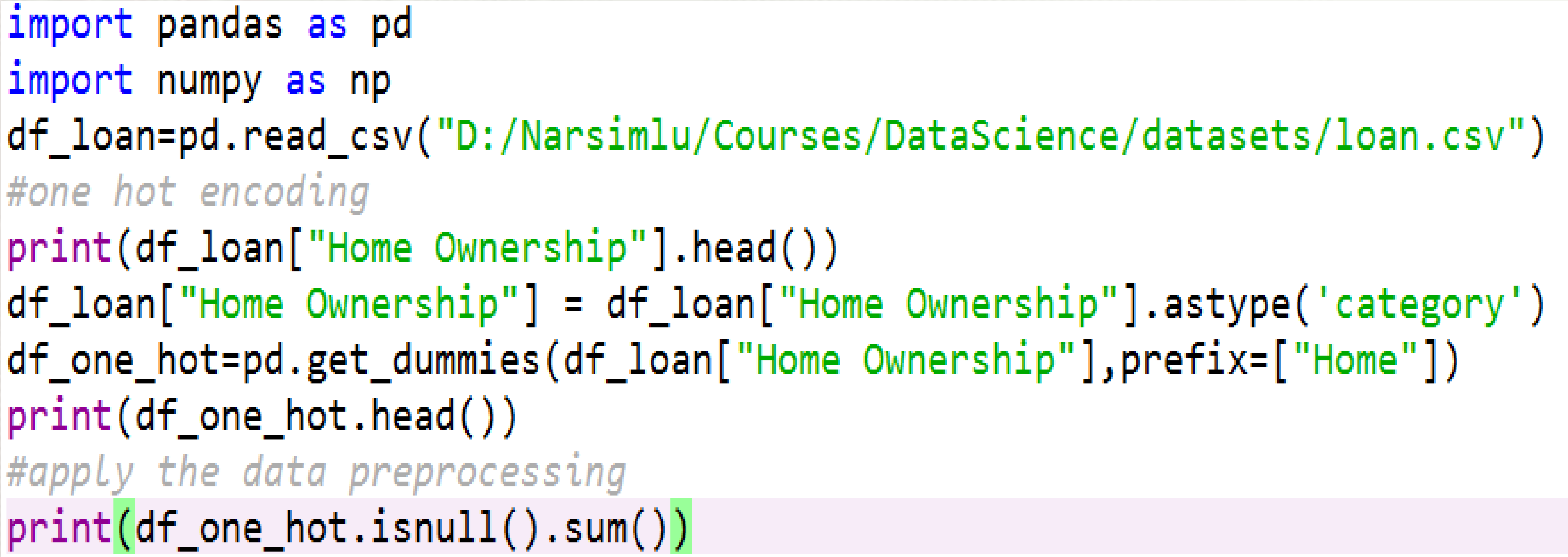
```python
import pandas as pd
data=pd.read_csv("loan.csv")
Loan_status=data["Loan Status"]
print(Loan_status.head(6))
cat_status={"Fully Paid":1,"Charged Off":0}
Loan_status.replace(cat_status, inplace=True)
print(Loan_status.head(6))
```

# Output



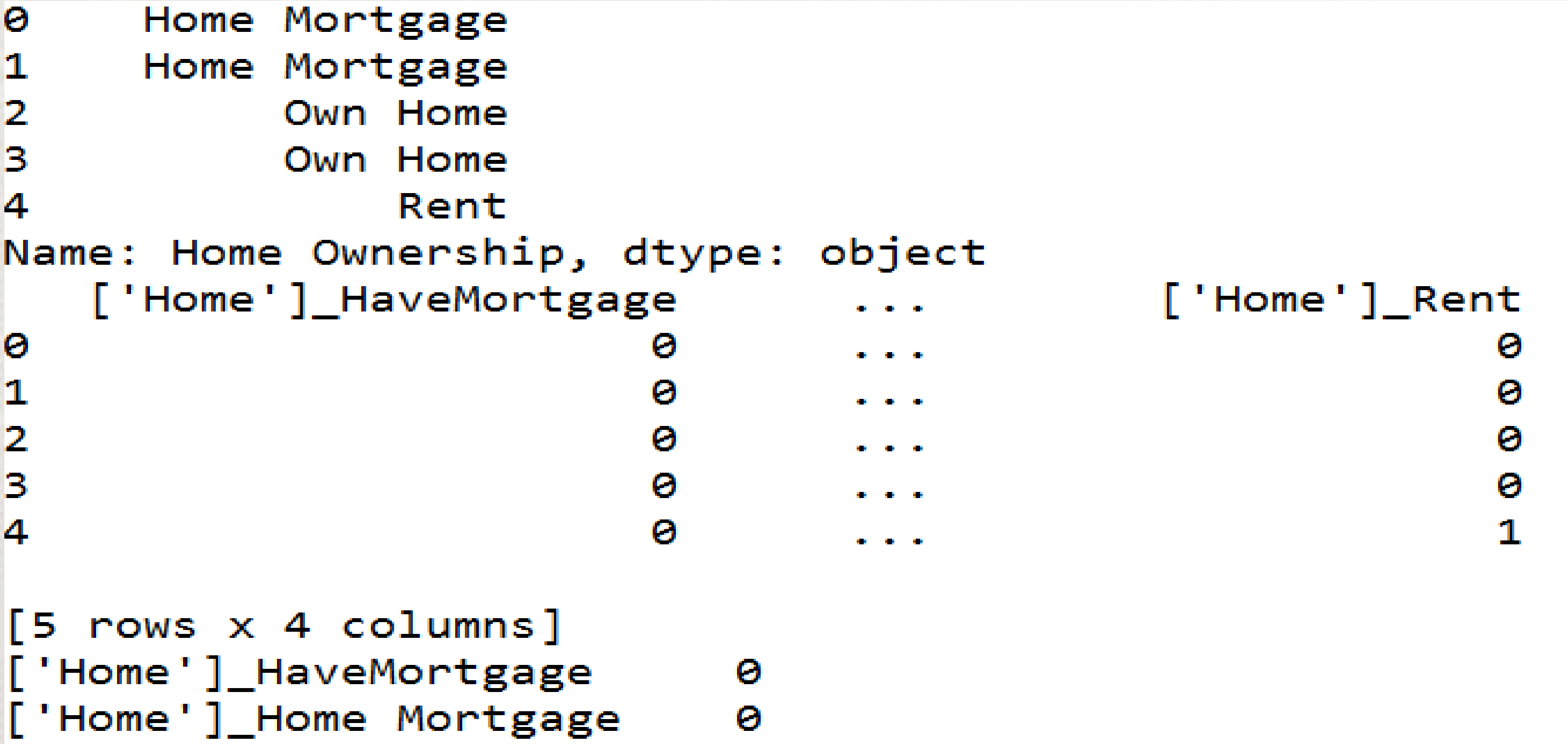
```
NPTEL_Python_DS/NPTEL_Assignments/
0      Fully Paid
1      Fully Paid
2      Fully Paid
3      Fully Paid
4      Fully Paid
5     Charged Off
Name: Loan Status, dtype: object
0     1
1     1
2     1
3     1
4     1
5     0
Name: Loan Status, dtype: int64
```

# One Hot Encoding



```python
import pandas as pd
import numpy as np
df_loan=pd.read_csv("D:/Narsimlu/Courses/DataScience/datasets/loan.csv")
#one hot encoding
print(df_loan["Home Ownership"].head())
df_loan["Home Ownership"] = df_loan["Home Ownership"].astype('category')
df_one_hot=pd.get_dummies(df_loan["Home Ownership"],prefix=["Home"])
print(df_one_hot.head())
#apply the data preprocessing
print(df_one_hot.isnull().sum())
```

# Output



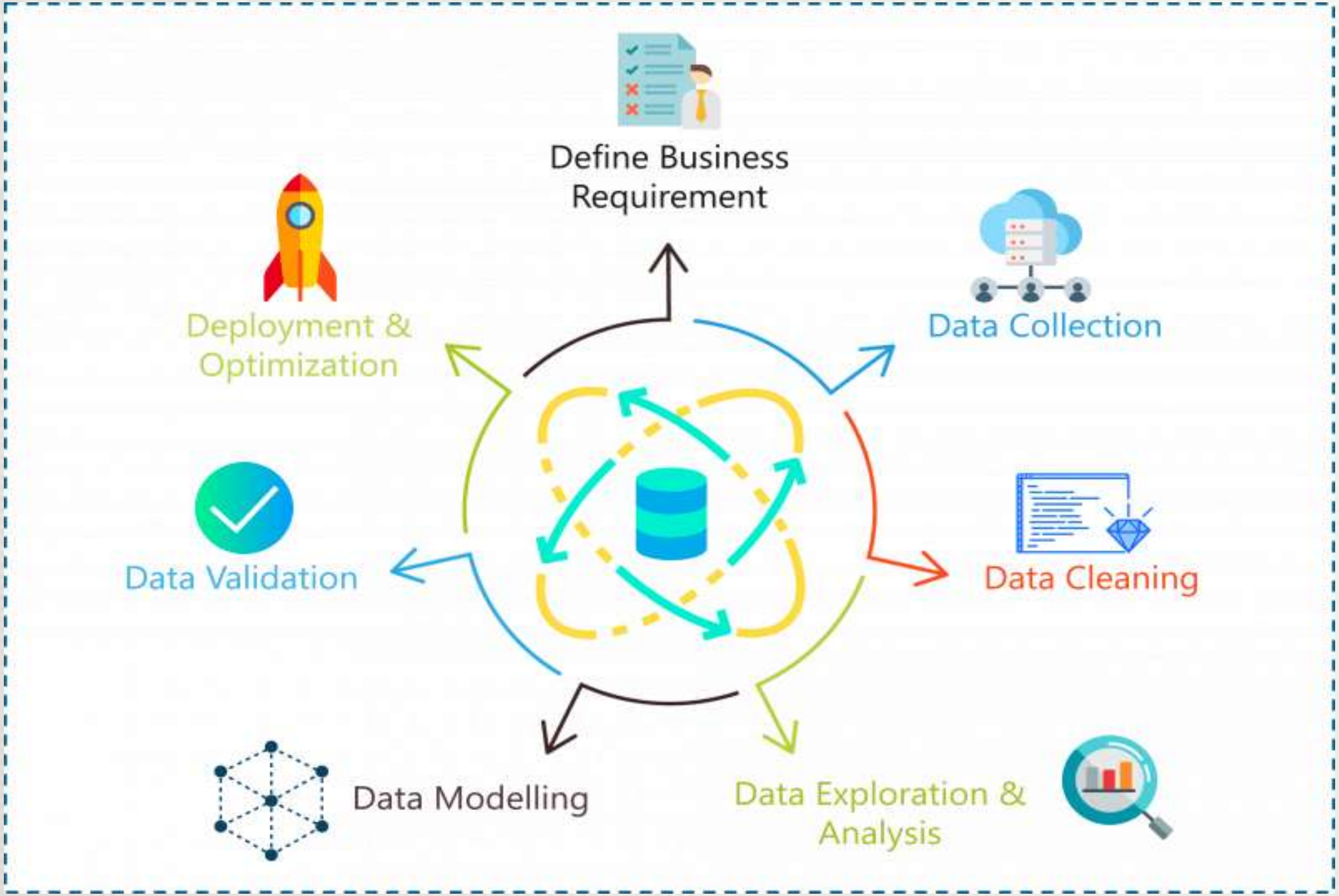
```
0    Home Mortgage
1    Home Mortgage
2         Own Home
3         Own Home
4             Rent
Name: Home Ownership, dtype: object
   ['Home']_HaveMortgage      ...        ['Home']_Rent
0                      0      ...                    0
1                      0      ...                    0
2                      0      ...                    0
3                      0      ...                    0
4                      0      ...                    1

[5 rows x 4 columns]
['Home']_HaveMortgage     0
['Home']_Home Mortgage    0
```

# Data Science-Project Life Cycle

A business problem statement in Data Science can be solved by following the phases

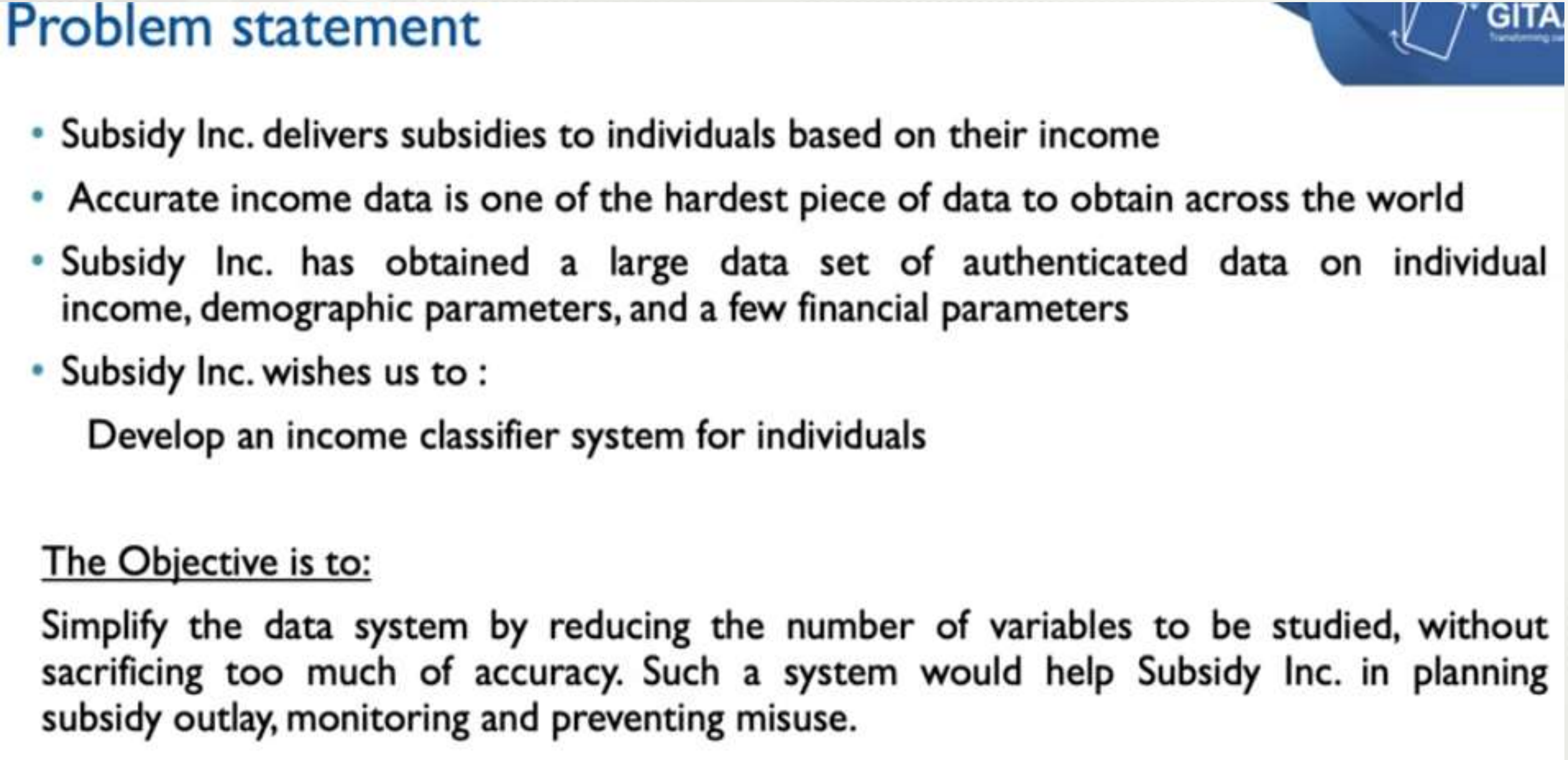
1. Define Problem Statement/ Business Requirement
2. Data Collection
3. Data Cleaning /preparation
4. Data Exploration & Analysis
5. Data Modelling
6. Deployment & Optimization

# Project Life Cycle -Architecture



Define Business Requirement
Data Collection
Data Cleaning
Data Exploration & Analysis
Data Modelling
Data Validation
Deployment & Optimization

# Case Study -01



## Problem statement

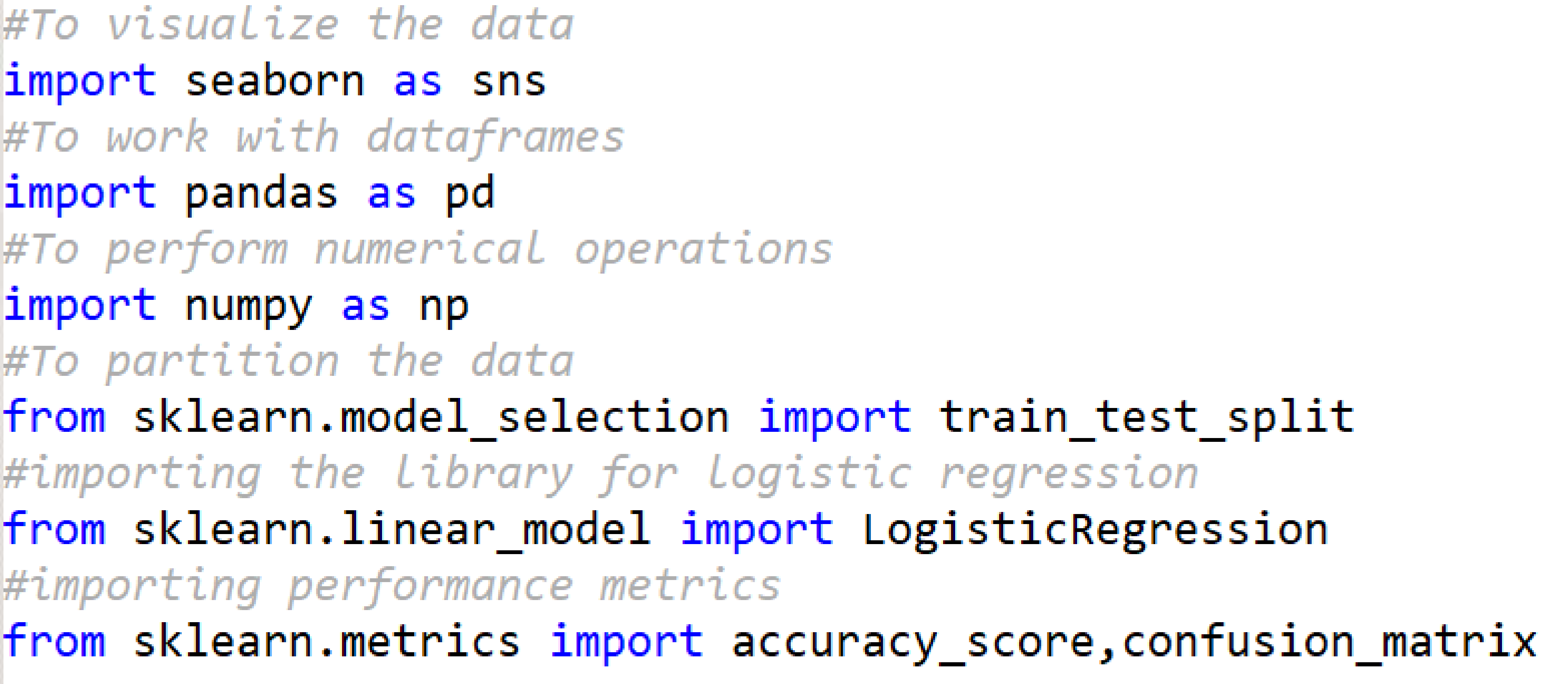
- Subsidy Inc. delivers subsidies to individuals based on their income
- Accurate income data is one of the hardest piece of data to obtain across the world
- Subsidy Inc. has obtained a large data set of authenticated data on individual income, demographic parameters, and a few financial parameters
- Subsidy Inc. wishes us to :

  Develop an income classifier system for individuals

<u>The Objective is to:</u>

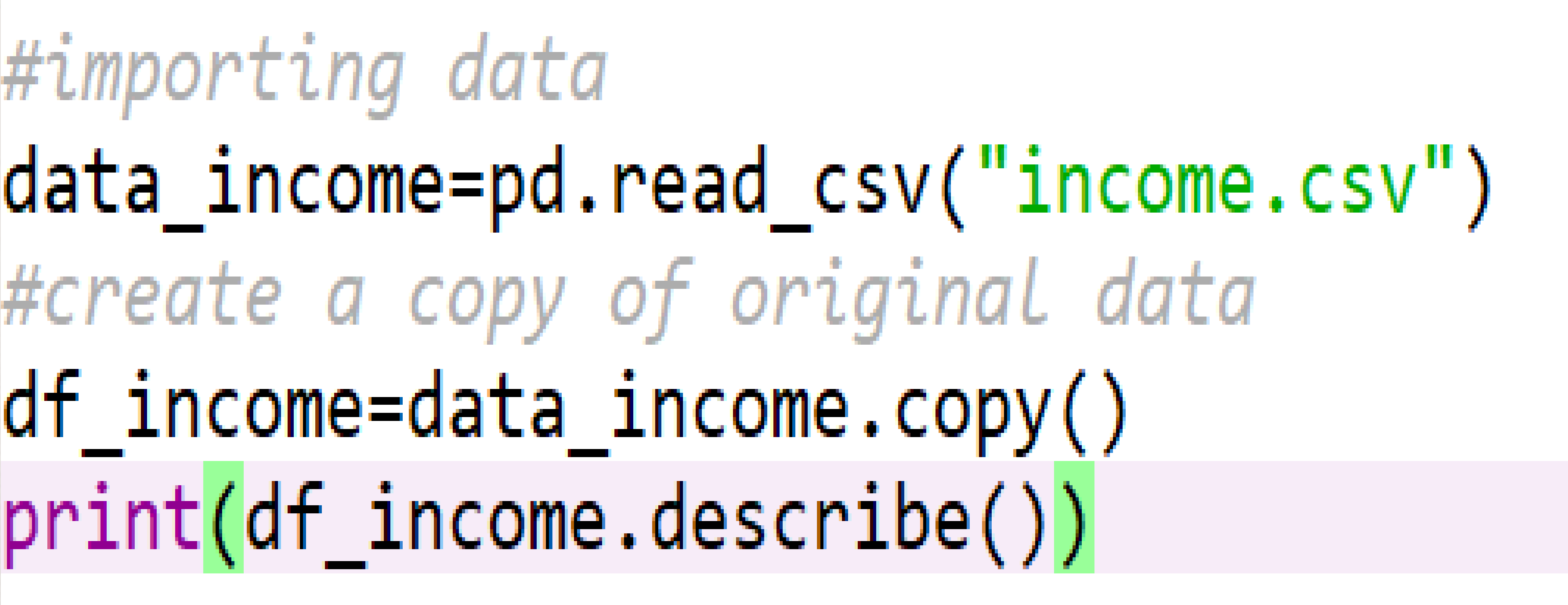
Simplify the data system by reducing the number of variables to be studied, without sacrificing too much of accuracy. Such a system would help Subsidy Inc. in planning subsidy outlay, monitoring and preventing misuse.

# Required modules



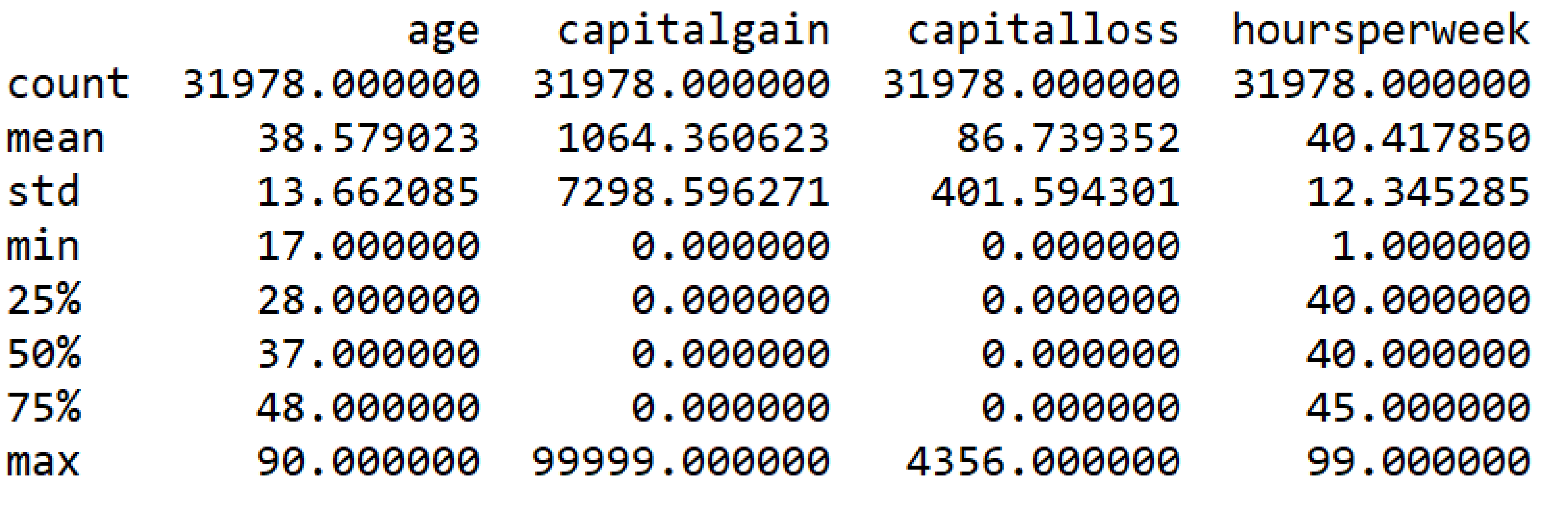
```
#To visualize the data
import seaborn as sns
#To work with dataframes
import pandas as pd
#To perform numerical operations
import numpy as np
#To partition the data
from sklearn.model_selection import train_test_split
#importing the library for logistic regression
from sklearn.linear_model import LogisticRegression
#importing performance metrics
from sklearn.metrics import accuracy_score,confusion_matrix
```

# Data Description



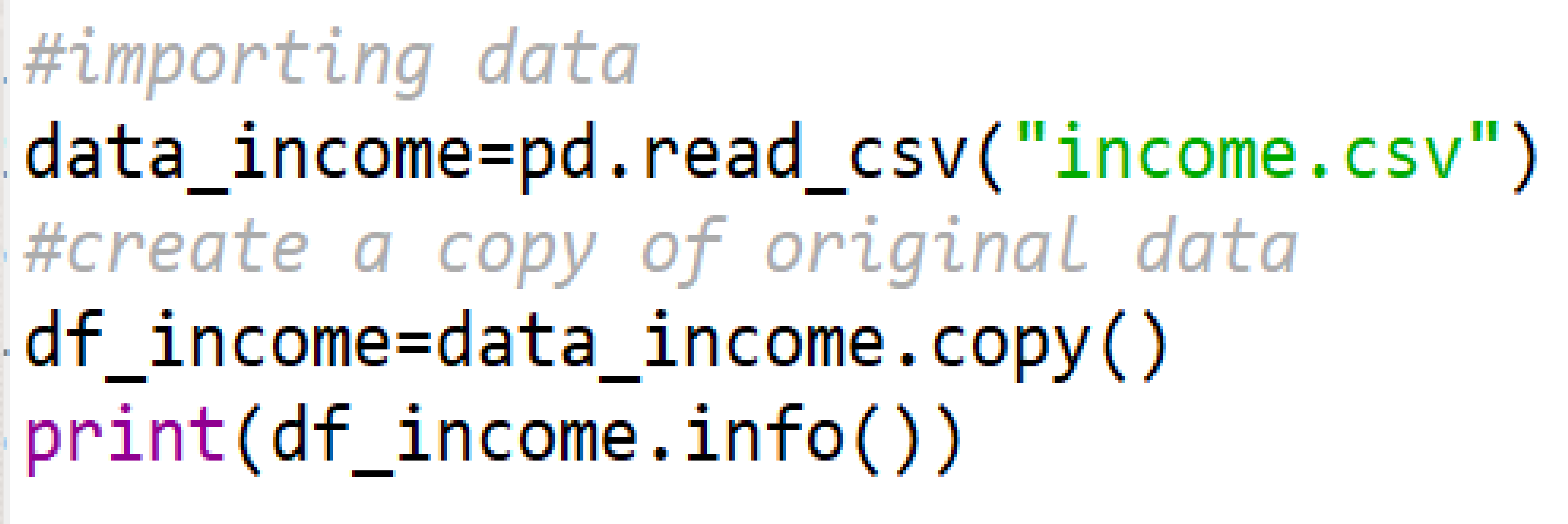
```
#importing data
data_income=pd.read_csv("income.csv")
#create a copy of original data
df_income=data_income.copy()
print(df_income.describe())
```

# Numerical Data Description



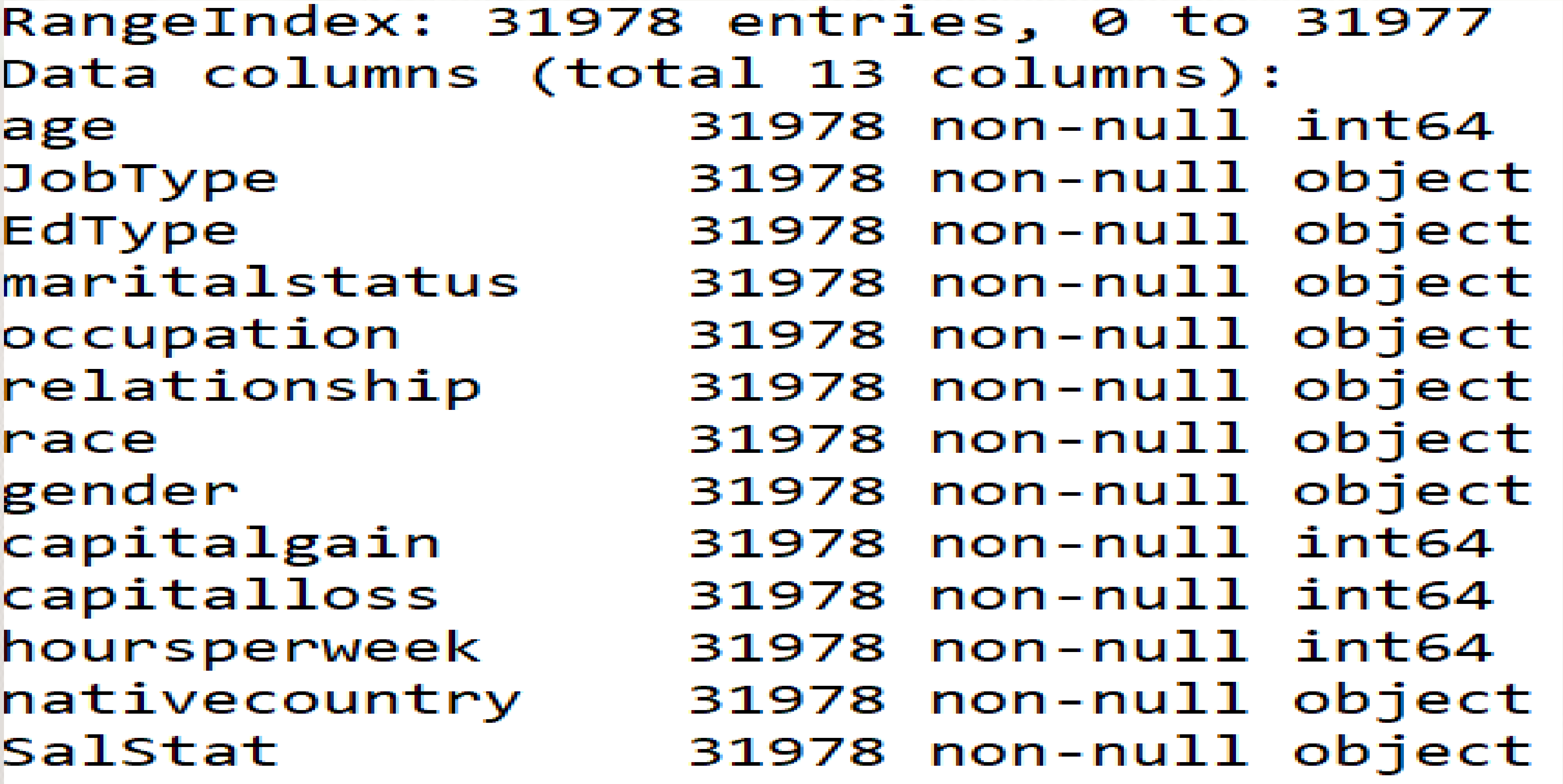
```
                age   capitalgain   capitalloss  hoursperweek
count  31978.000000  31978.000000  31978.000000  31978.000000
mean      38.579023   1064.360623     86.739352     40.417850
std       13.662085   7298.596271    401.594301     12.345285
min       17.000000      0.000000      0.000000      1.000000
25%       28.000000      0.000000      0.000000     40.000000
50%       37.000000      0.000000      0.000000     40.000000
75%       48.000000      0.000000      0.000000     45.000000
max       90.000000  99999.000000   4356.000000     99.000000
```

# Information about Data Set



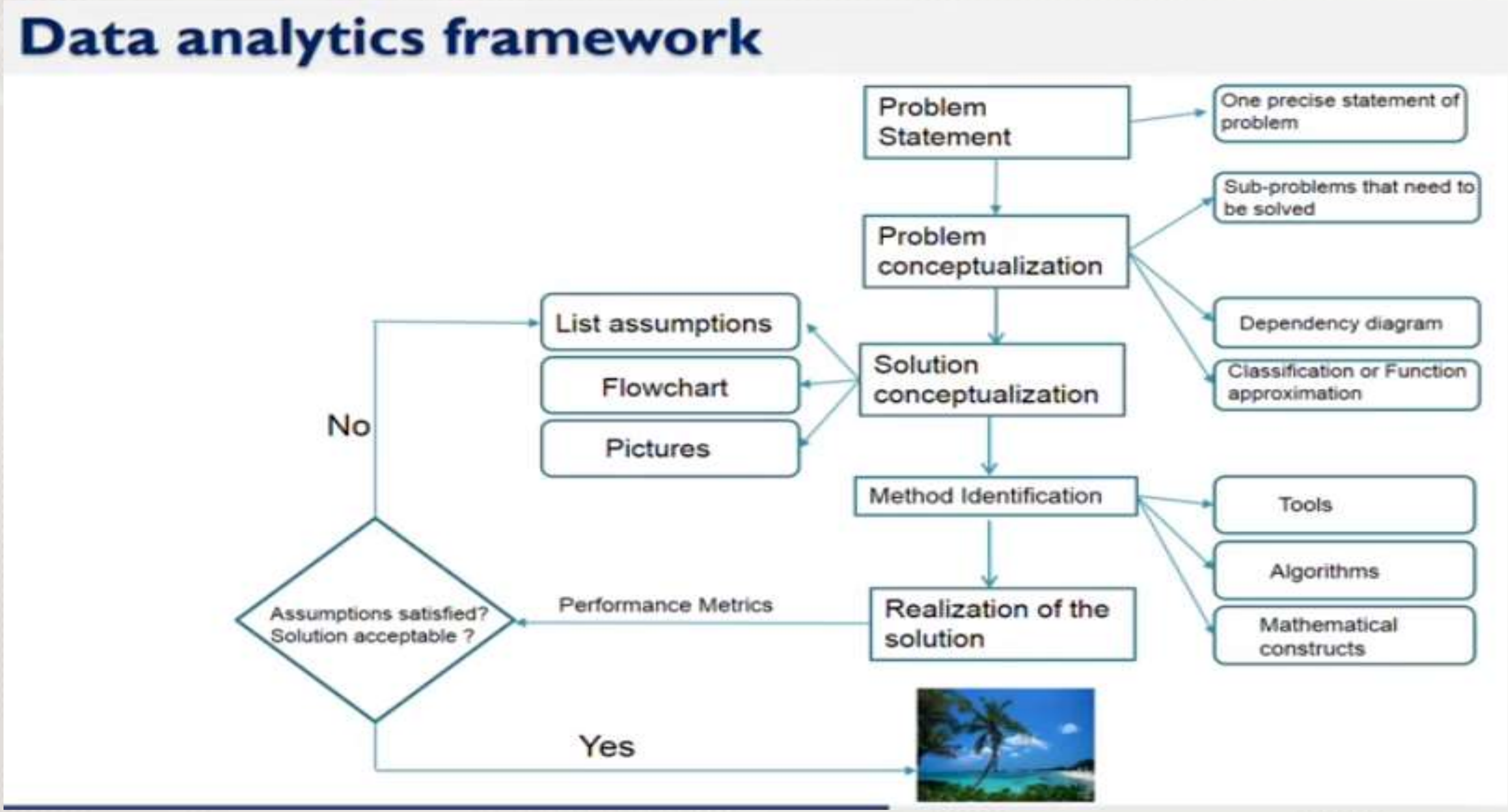
```
#importing data
data_income=pd.read_csv("income.csv")
#create a copy of original data
df_income=data_income.copy()
print(df_income.info())
```

# Output



```
RangeIndex: 31978 entries, 0 to 31977
Data columns (total 13 columns):
age              31978 non-null int64
JobType          31978 non-null object
EdType           31978 non-null object
maritalstatus    31978 non-null object
occupation       31978 non-null object
relationship     31978 non-null object
race             31978 non-null object
gender           31978 non-null object
capitalgain      31978 non-null int64
capitalloss      31978 non-null int64
hoursperweek     31978 non-null int64
nativecountry    31978 non-null object
SalStat          31978 non-null object
```

# Data Analytics Frame work



## Data analytics framework

Problem Statement

One precise statement of problem

Problem conceptualization

Sub-problems that need to be solved

Dependency diagram

Classification or Function approximation

Solution conceptualization

List assumptions

Flowchart

Pictures

Method Identification

Tools

Algorithms

Mathematical constructs

Realization of the solution

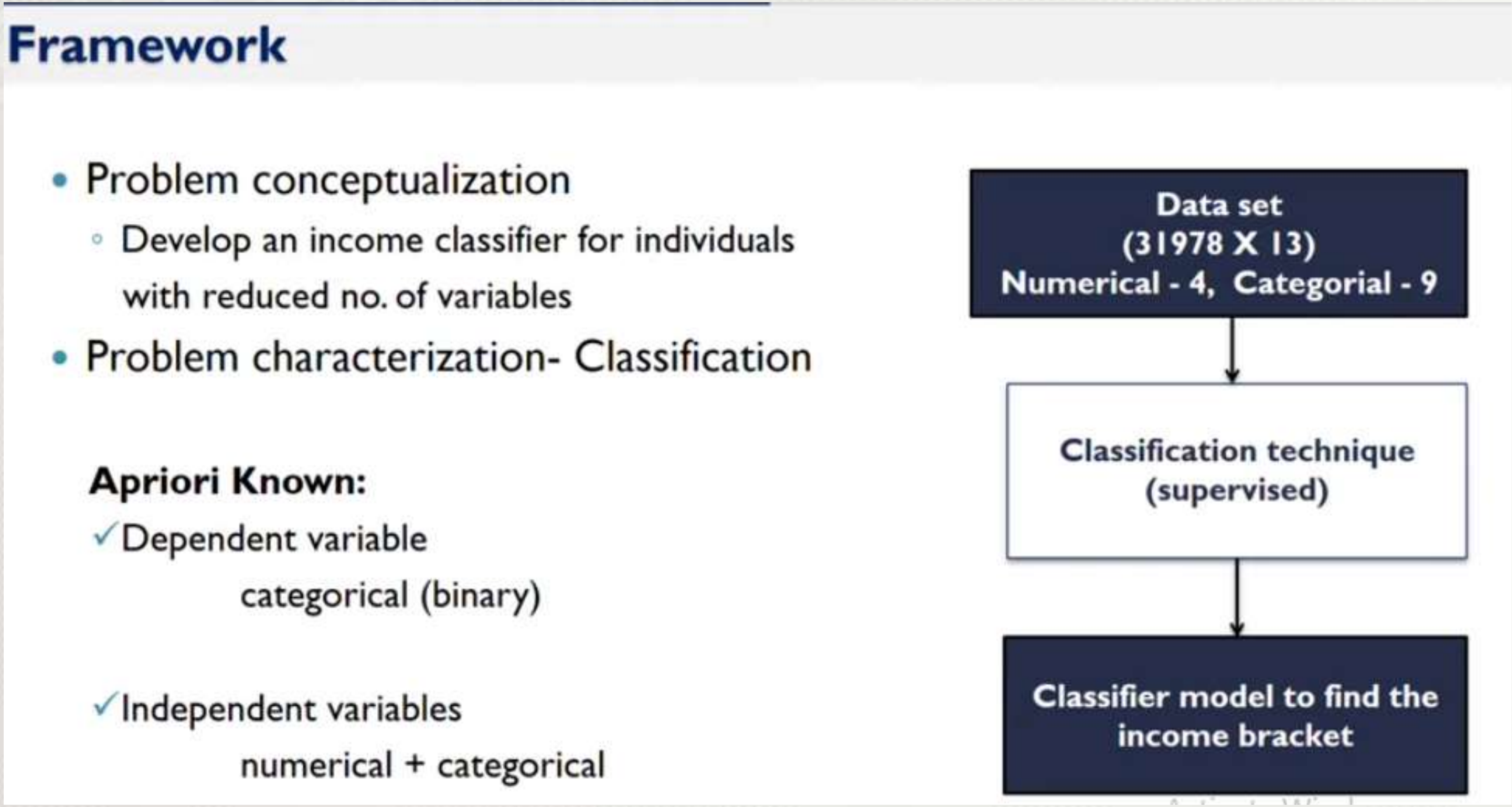
Performance Metrics

Assumptions satisfied? Solution acceptable ?

No

Yes

# Problem conceptualization



## Framework

- Problem conceptualization
  - Develop an income classifier for individuals with reduced no. of variables
- Problem characterization- Classification

**Apriori Known:**

✓Dependent variable

categorical (binary)

✓Independent variables

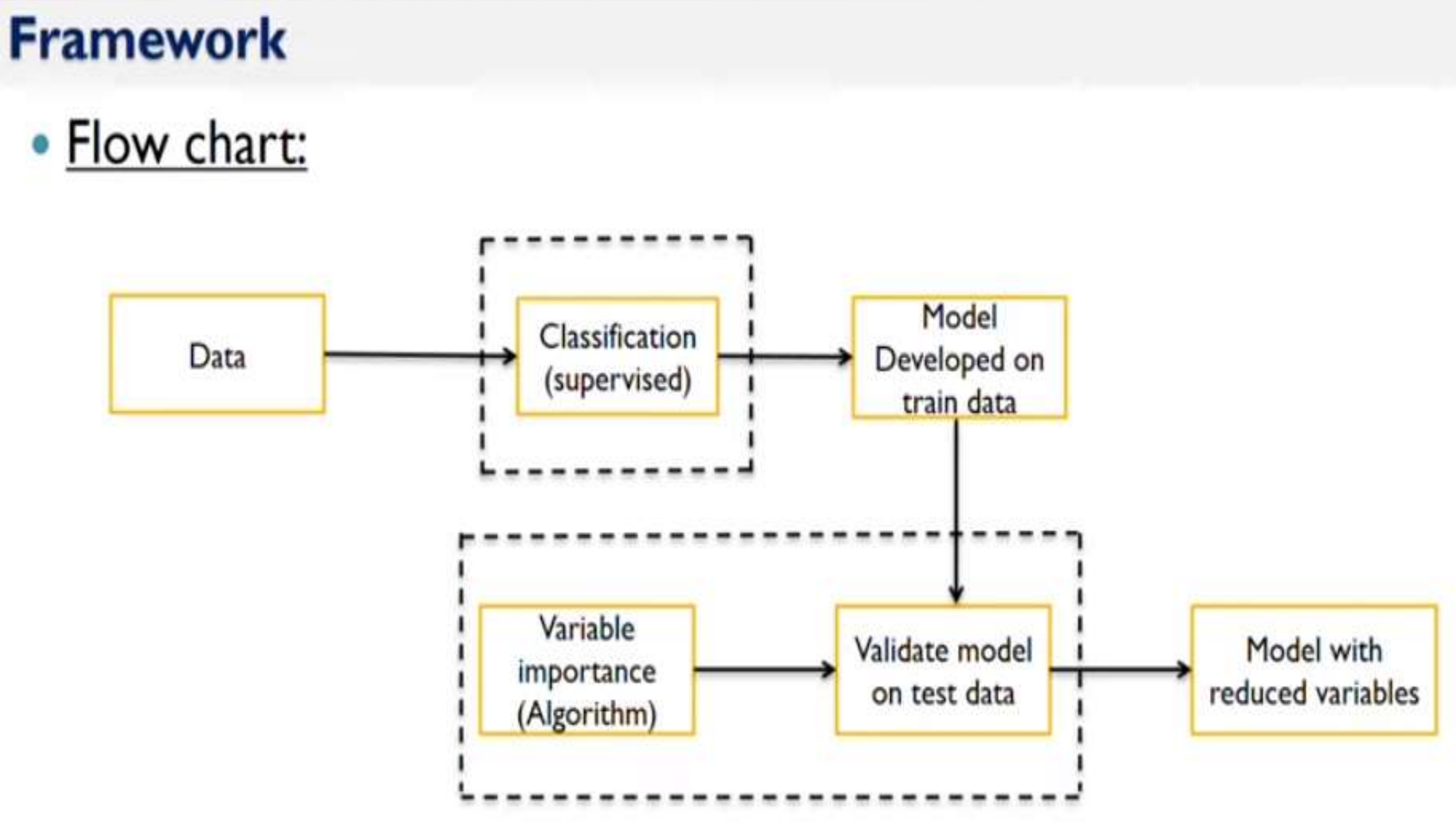
numerical + categorical

Data set
(31978 X 13)
Numerical - 4, Categorial - 9

Classification technique
(supervised)

Classifier model to find the income bracket

# Framework

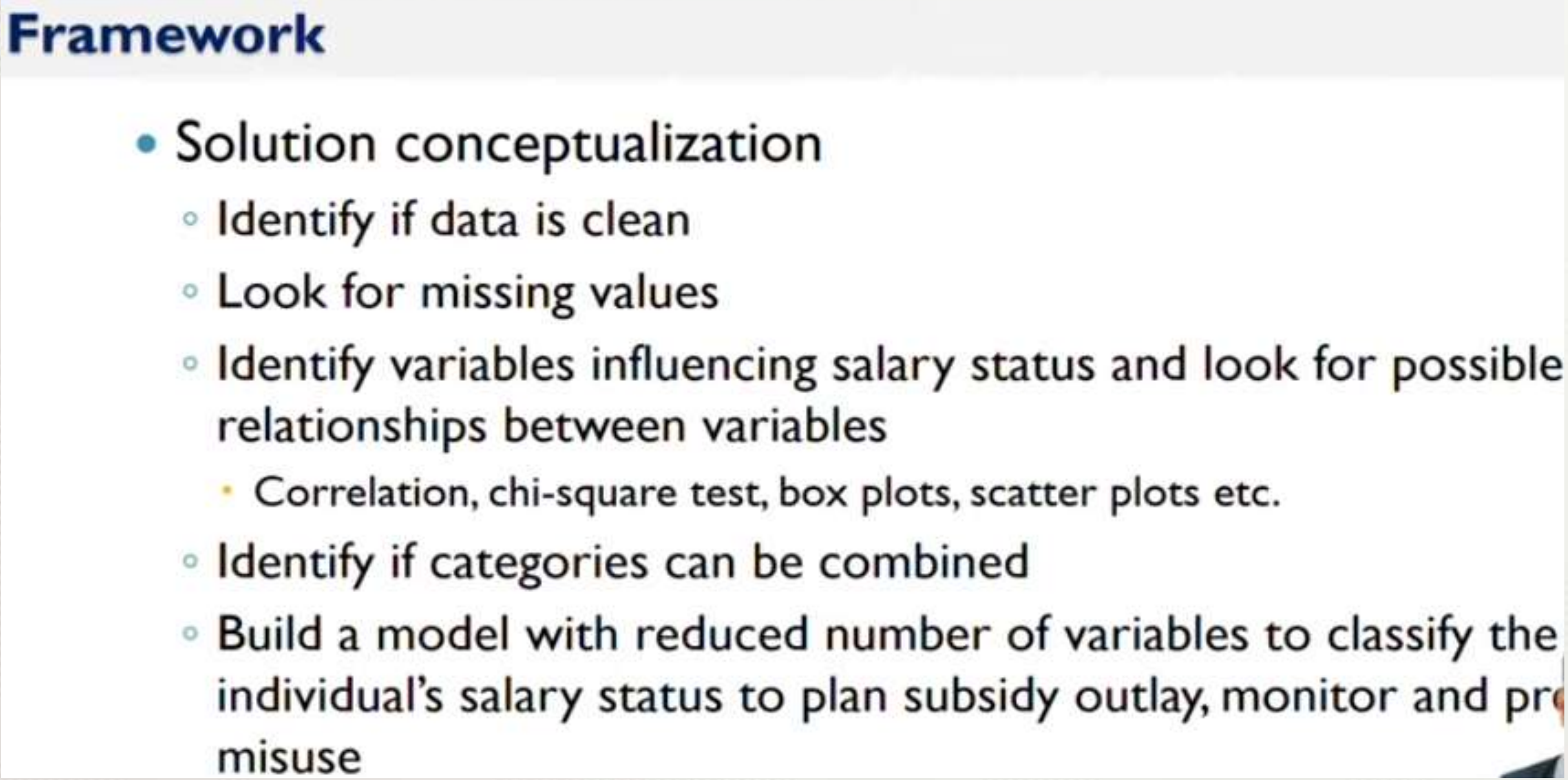


## Framework

- Flow chart:

Data
Classification (supervised)
Model Developed on train data
Variable importance (Algorithm)
Validate model on test data
Model with reduced variables

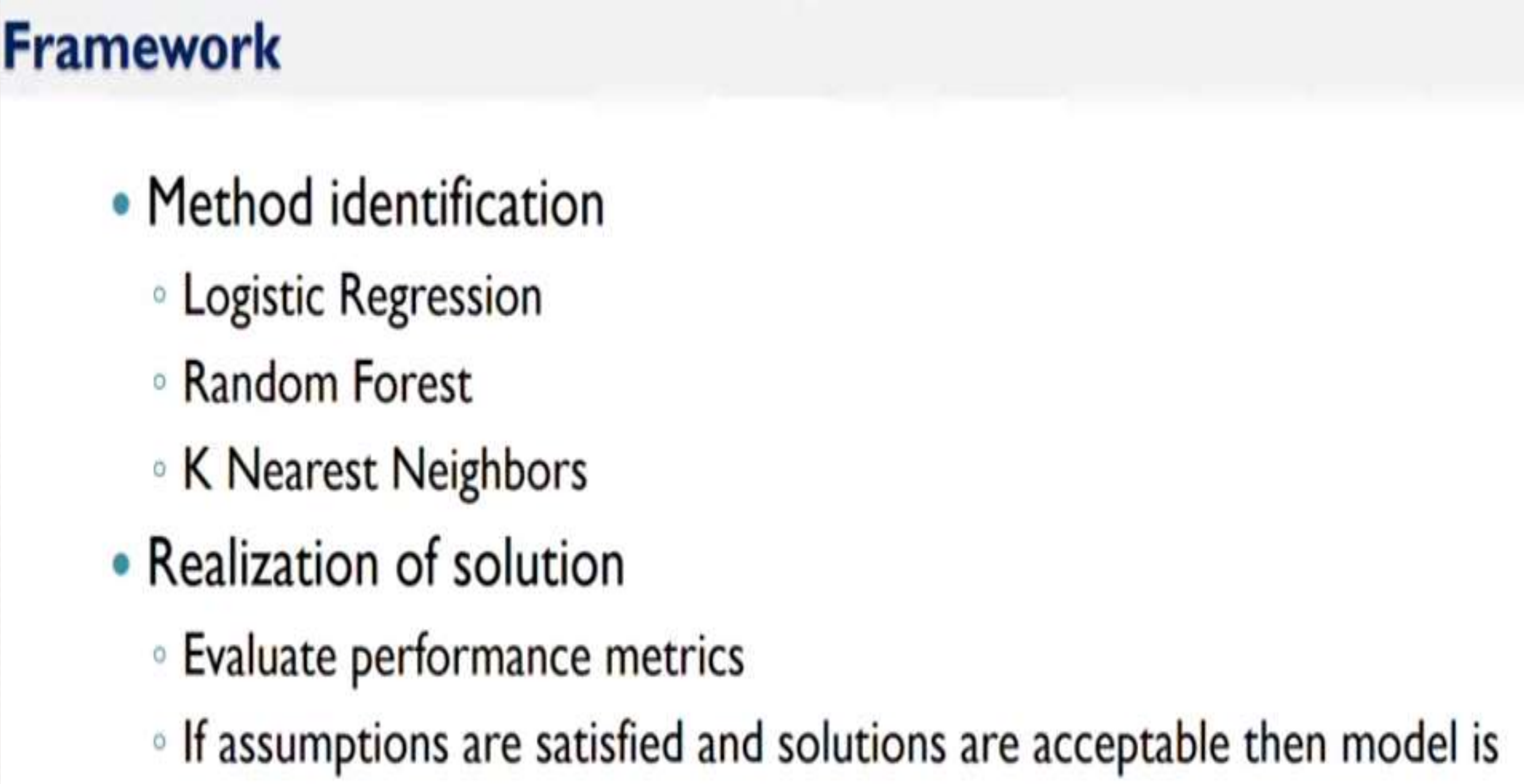
# Solution conceptualization



## Framework

- Solution conceptualization
  - Identify if data is clean
  - Look for missing values
  - Identify variables influencing salary status and look for possible relationships between variables
    - Correlation, chi-square test, box plots, scatter plots etc.
  - Identify if categories can be combined
  - Build a model with reduced number of variables to classify the individual's salary status to plan subsidy outlay, monitor and pr misuse

# Method Indentification



## Framework

- Method identification
  - Logistic Regression
  - Random Forest
  - K Nearest Neighbors
- Realization of solution
  - Evaluate performance metrics
  - If assumptions are satisfied and solutions are acceptable then model is

# Conclusion

You are aware of

Data Encoding

Project Life Cycle

We will proceed with

Case Study

# THANK YOU