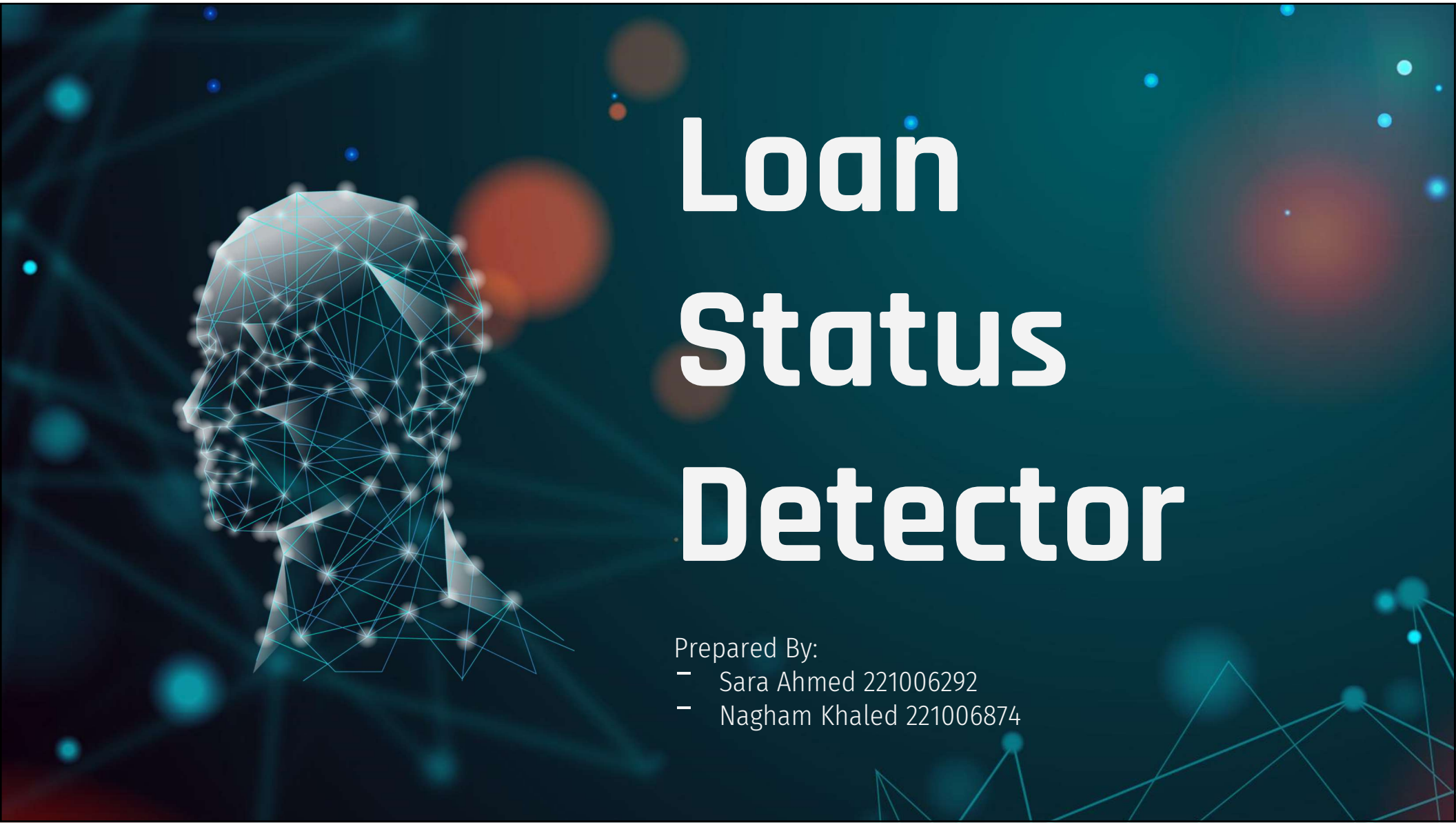


Loan Status Detector

Prepared By:

- Sara Ahmed 221006292
- Nagham Khaled 221006874

AI TECH AGENCY INFOGRAPHICS

1 Introduction

2 Dataset

3 Processing

4 Results

Introduction

- - This presentation explores the critical steps involved in data preprocessing and the application of a decision tree classifier to predict loan default.

- - The goal of the project is to check loan status.

DataSet

```
dataset = pd.read_csv('D:\Downloads\loan_data.csv’)
dataset.columns=dataset.columns.str.strip()
```

person_age	person_gender	person_education	person_income	person_emp_exp	person_home_ownership	loan_amnt
22	female	Master	71948	0	RENT	35000
21	female	High School	12282	0	OWN	1000
25	female	High School	12438	3	MORTGAGE	5500
23	female	Bachelor	79753	0	RENT	35000
24	male	Master	66135	1	RENT	35000

loan_intent	loan_int_rate	loan_percent_income	cb_person_cred_hist_length	credit_score	previous_loan_defaults_on_file	loan_status
PERSONAL	16.02	0.49	3	561	No	1
EDUCATION	11.14	0.08	2	504	Yes	0
MEDICAL	12.87	0.44	3	635	No	1
MEDICAL	15.23	0.44	2	675	No	1
MEDICAL	14.27	0.53	4	586	No	1

Processing

```
# Splitting the Labels from the data inputs
X = dataset.iloc[:, :-1].values
Y = dataset.iloc[:, 13].values

# Convert strings to numericals
from sklearn.preprocessing import LabelEncoder
LabelEncoder_Y = LabelEncoder()
Y = LabelEncoder_Y.fit_transform(Y)


LabelEncoder_X = LabelEncoder()
X[:,1] = LabelEncoder_X.fit_transform(X[:, 2])
X[:,2]= LabelEncoder_X.fit_transform(X[:, 3])
X[:,5]= LabelEncoder_X.fit_transform(X[:, 6])
X[:,7]= LabelEncoder_X.fit_transform(X[:, 8])
X[:,-1]= LabelEncoder_X.fit_transform(X[:, -1])
```

Processing

```
# Splitting the data into training and testing
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(X, Y, test_size=0.2, random_state=0)

# Scaling the features to be able to make the values closer
from sklearn.preprocessing import StandardScaler
sc_X = StandardScaler()
x_train = sc_X.fit_transform(x_train)
x_test = sc_X.transform(x_test)

# Model Building
from sklearn import tree
DT = tree.DecisionTreeClassifier(criterion='entropy', random_state=0)
y_pred = DT.fit(x_train, y_train).predict(x_test)
```

Results

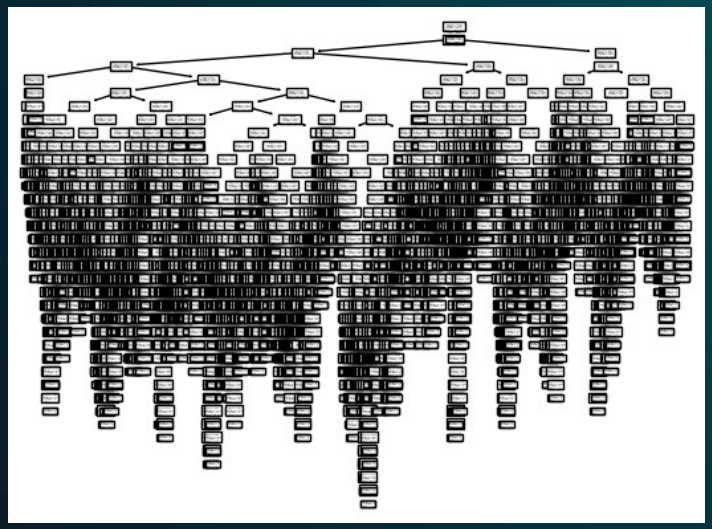
```
# Get Accuracy
from sklearn.metrics import accuracy_score
tree_score = accuracy_score(y_pred, y_test)

# Print the Decision Tree
import matplotlib.pyplot as plt
tree.plot_tree(DT)
plt.show()

# Print the Accuracy
print(f"Output of Decision Tree {tree_score}")
```

Results

Decision Tree:



Output of Decision Tree 0.8831111111111111

Thank You!