

Team Members:

Sanjay Pratap T K (1905046)

Nidin Sree (1905034)

Sagolsem Hironika Devi (1905044)

Jeiranjeni V (1905020)

Veipunii lana Ng (1905057)

Team id: PNT2022TMID52586

The screenshot shows the IBM Watson Studio interface. The top navigation bar includes 'Service Details - IBM Cloud', 'IBM Watson Studio', and the project ID 'IBM-Project-23299-1659877556'. The main header shows the project name 'Loan credibility model deployment' and a 'Launch IDE' button. The 'Assets' tab is active, displaying a list of assets:

Name	Last modified
Decision Tree Model Notebook	Now Modified by you
Random forest classifier Notebook	3 minutes ago Modified by you
loan_prediction.csv	13 minutes ago Modified by you

On the right, the 'About this project' sidebar shows the project name, description, collaborators (SANJAY PRATAP T K (you)), controls (Restricted eligibility), cloud object storage (130.3 KB used), IBM Cloud account details, and SAML federation.

The screenshot shows the IBM Watson Studio IDE interface. The top navigation bar includes 'Service Details - IBM Cloud', 'Decision Tree Model - IBM Watson Studio', and the project ID 'IBM-Project-23299-1659877556'. The main header shows the project name 'Loan credibility model deployment' and the notebook name 'Decision Tree Model'. The notebook is open, showing the following code:

```
The dataset is already download in .csv format
```

IMPORTING THE PACKAGE

```
In [1]: import numpy as np
import pandas as pd

import warnings
warnings.filterwarnings('ignore')
```

Load the dataset

```
In [2]: df=pd.read_csv("C:\\Users\\jey\\Downloads\\loan_prediction.csv")

In [3]: df
```

The output of the code is a DataFrame with the following columns: Loan_ID, Gender, Married, Dependents, Education, Self_Employed, ApplicantIncome, CoapplicantIncome, LoanAmount, Loan_Amount_Term, Credit_History, Property_Area, Loan_Status.

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	CoapplicantIncome	LoanAmount	Loan_Amount_Term	Credit_History	Property_Area	Loan_Status
0	LP001002	Male	No	0	Graduate	No	5849	0.0	NaN	360.0	1.0	Urban	Y
1	LP001003	Male	Yes	1	Graduate	No	4583	1508.0	128.0	360.0	1.0	Rural	N
2	LP001005	Male	Yes	0	Graduate	Yes	3000	0.0	66.0	360.0	1.0	Urban	Y
3	LP001006	Male	Yes	0	Not Graduate	No	2583	2358.0	120.0	360.0	1.0	Urban	Y
4	LP001008	Male	No	0	Graduate	No	6000	0.0	141.0	360.0	1.0	Urban	Y

Service Details - IBM CloudDecision Tree Model - IBM WatsonStudioIBM Watson StudioIBM Project-23299-1659877556

dataplatform.cloud.ibm.com/analytics/notebooks/v2/3082ee4d-551c-48d4-b75b-e2d1dc7842d1?projectId=be499232-04bb-4f67-924d-38af27af4374&context=cpsdaas

IBM Watson StudioSearch in your workspacesBuySANJAY PRATAP T K's Acc...Dallas

Projects / Loan credibility model deployment / Decision Tree Model

FileEditViewInsertCellKernelHelp

RunFormatMarkdown

Treating the Null Value

We will fill the missing values in numeric data type using the mean value of that particular column and categorical data type using the most repeated value

```
In [6]: numerical_features = df.select_dtypes(include=[np.number]).columns
categorical_features = df.select_dtypes(include=[np.object]).columns

In [7]: numerical_features
Out[7]: Index(['ApplicantIncome', 'CoapplicantIncome', 'LoanAmount',
              'Loan_Amount_Term', 'Credit_History'],
              dtype='object')

In [8]: categorical_features
Out[8]: Index(['Loan_ID', 'Gender', 'Married', 'Dependents', 'Education',
              'Self_Employed', 'Property_Area', 'Loan_Status'],
              dtype='object')

In [9]: df['Gender'] = df['Gender'].fillna(df['Gender'].mode()[0])
df['Married'] = df['Married'].fillna(df['Married'].mode()[0])
#replace + with non value
df['Dependents'] = df['Dependents'].str.replace('+','')
df['Dependents'] = df['Dependents'].fillna(df['Dependents'].mode()[0])
df['LoanAmount'] = df['LoanAmount'].fillna(df['LoanAmount'].mode()[0])
df['Self_Employed'] = df['Self_Employed'].fillna(df['Self_Employed'].mode()[0])
df['Loan_Amount_Term'] = df['Loan_Amount_Term'].fillna(df['Loan_Amount_Term'].mode()[0])
df['Credit_History'] = df['Credit_History'].fillna(df['Credit_History'].mode()[0])

In [10]: #check in the null values now
```

27°C Haze18:33 19-11-2022

Service Details - IBM CloudDecision Tree Model - IBM WatsonStudioIBM Watson StudioIBM Project-23299-1659877556

dataplatform.cloud.ibm.com/analytics/notebooks/v2/3082ee4d-551c-48d4-b75b-e2d1dc7842d1?projectId=be499232-04bb-4f67-924d-38af27af4374&context=cpsdaas

IBM Watson StudioSearch in your workspacesBuySANJAY PRATAP T K's Acc...Dallas

Projects / Loan credibility model deployment / Decision Tree Model

FileEditViewInsertCellKernelHelp

RunFormatMarkdown

Handling Categorical Values

```
In [11]: df.select_dtypes(include='object').columns
Out[11]: Index(['Loan_ID', 'Gender', 'Married', 'Dependents', 'Education',
              'Self_Employed', 'Property_Area', 'Loan_Status'],
              dtype='object')

In [12]: df['Gender'].unique()
Out[12]: array(['Male', 'Female'], dtype=object)

In [13]: df['Gender'].replace({'Male':1,'Female':0},inplace=True)

In [14]: df['Married'].unique()
Out[14]: array(['No', 'Yes'], dtype=object)

In [15]: df['Married'].replace({'Yes':1,'No':0},inplace=True)

In [16]: df['Dependents'].unique()
Out[16]: array(['0', '1', '2', '3'], dtype=object)

In [17]: df['Dependents'].replace({'0':0,'1':1,'2':2,'3':3},inplace=True)

In [18]: df['Self_Employed'].unique()
Out[18]: array(['No', 'Yes'], dtype=object)
```

27°C Haze18:33 19-11-2022

Projects / Loan credibility model deployment / Decision Tree Model

Balancing The Dataset

```
In [29]: from imblearn.combine import SMOTETomek

In [30]: smote = SMOTETomek(0.90)

In [31]: #dividing the dataset into dependent and independent y and x respectively
y = df['Loan_Status']
x = df.drop(columns=['Loan_Status'],axis=1)

In [32]: #creating the new x and y for balance data
x_bal,y_bal = smote.fit_resample(x,y)

In [33]: #printing the value before and after balancing
print(y.value_counts())
print(y_bal.value_counts())

1    422
0    192
Name: Loan_Status, dtype: int64

1    357
0    314
Name: Loan_Status, dtype: int64
```

Scaling The Data

Projects / Loan credibility model deployment / Decision Tree Model

Evaluating the model

```
In [41]: from sklearn.tree import DecisionTreeClassifier
model = DecisionTreeClassifier(max_depth=4,splitter='best',criterion='entropy')

In [42]: model.fit(x_train,y_train)

Out[42]: DecisionTreeClassifier(criterion='entropy', max_depth=4)
In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

In [43]: y_predict= model.predict(x_test)
y_predict

Out[43]: array([1, 0, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1, 0, 1, 0, 1, 0,
0, 0, 0, 0, 0, 0, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 1, 0, 0, 1, 1, 0,
0, 0, 1, 0, 1, 0, 0, 1, 0, 1, 0, 0, 0, 1, 1, 1, 1, 1, 0, 1, 0,
0, 1, 1, 1, 1, 0, 1, 1, 1, 0, 0, 0, 0, 1, 1, 0, 1, 1, 1, 0, 0,
1, 0, 1, 1, 0, 0, 1, 0, 1, 0, 1, 0, 1, 1, 1, 0, 0, 0, 0, 1, 1,
1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 0, 0, 1, 0,
1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 0, 0, 1, 0, 1, 0, 0, 0, 1, 0,
1, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 1, 0, 1, 1, 1, 1,
0, 1, 1, 1, 1, 0, 1, 1, 0, 0, 1, 1, 0, 1, 0, 1, 1, 1, 1,
1, 0, 1, 1, 0, 1, 0, 0, 0, 1, 1, 0, 1, 1, 1, 0, 1, 1, 1,
1, 0, 1, 1, 0, 1, 0, 0, 0, 1, 1, 0, 1, 1, 1, 0, 1, 1, 0, 0,
1, 1], dtype=int64)

In [44]: y_predict_train = model.predict(x_train)
```

Service Details - IBM CloudDecision Tree Model - IBM WatsonStudioIBM Project-23299-165987755/

datapatform.cloud.ibm.com/analitics/notebooks/v2/3082ee4d-551c-48d4-b75b-e2d1dc7842d1?projectId=be499232-04bb-4167-924d-38af27af4374&context=cpdaas

IBM Watson StudioSearch in your workspacesBuySANJAY PRATAP T K's Acc...DallasSR

Projects / Loan credibility model deployment / Decision Tree Model

FileEditViewInsertCellKernelHelp

Not Trusted | Python 3.9

RunFormatCode

Confusion matrix

```
In [47]: pd.crosstab(y_test,y_predict)
Out[47]:
```

	col_0	0	1
Loan_Status			
0	81	30	
1	7	104	

```
In [48]: print(classification_report(y_test,y_predict))
```

	precision	recall	f1-score	support
0	0.92	0.73	0.81	111
1	0.78	0.94	0.85	111
accuracy			0.83	222
macro avg	0.85	0.83	0.83	222
weighted avg	0.85	0.83	0.83	222

```
In [ ]:
```

27°C Haze

ENG IN18:34 19-11-2022