

Project Report Template

1 INTRODUCTION

Overview

A brief description about your project

Purpose

The use of this project. What can be achieved using this.

2 Problem Definition & Design Thinking

Empathy Map

Paste the empathy map screenshot

Ideation & Brainstorming Map

Paste the Ideation & brainstorming map screenshot

3 RESULT

Final findings (Output) of the project along with screenshots.

4 ADVANTAGES & DISADVANTAGES

List of advantages and disadvantages of the proposed solution

5 APPLICATIONS

The areas where this solution can be applied

6 CONCLUSION

Conclusion summarizing the entire work and findings.

7 FUTURE SCOPE

Enhancements that can be made in the future.

8 APPENDIX

A. Source Code

Attach the code for the solution built.

1 INTRODUCTION

1.1 Overview

A loan is a sum of money that is borrowed and repaid over a period of time, typically with interest. There are various types of loans available to individuals and businesses, such as personal loans, mortgages, auto loans, student loans, business loans and many more. They are offered by banks, credit unions, and other financial institutions, and the terms of the loan, such as interest rate, repayment period, and fees, vary depending on the lender and the type of loan.

1.2 purpose

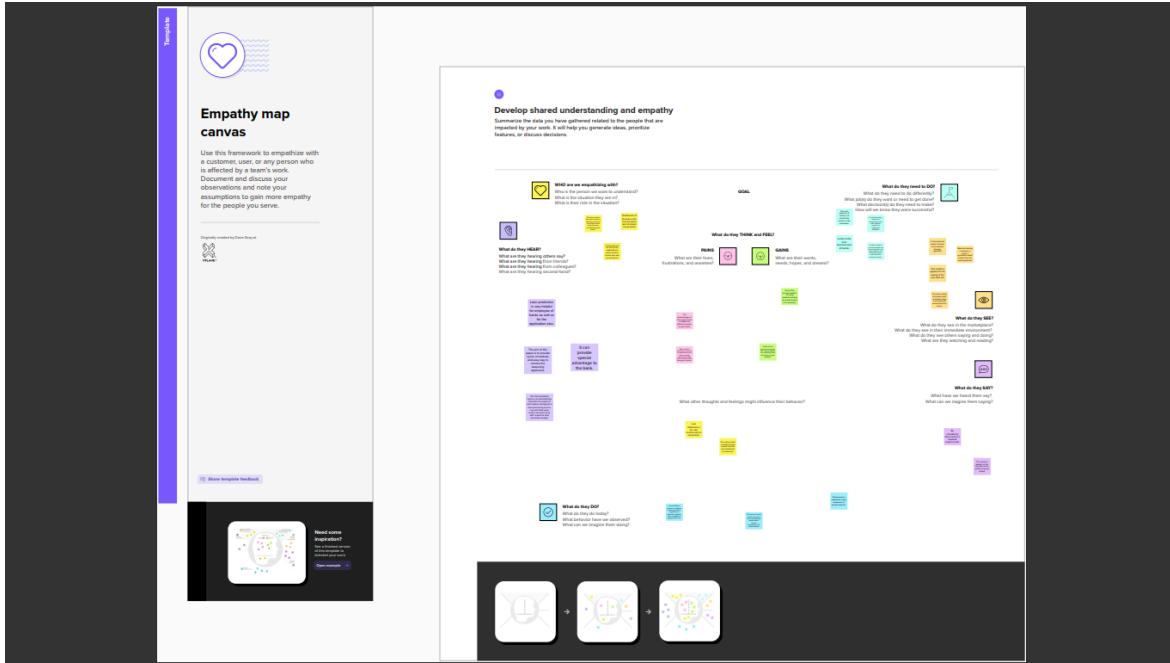
A personal loan is a type of unsecured loan that can be used for a variety of expenses such as home repairs, medical expenses, debt consolidation, and more. The loan amount, interest rate, and repayment period vary depending on the lender and the borrower's creditworthiness. To qualify for a personal loan, borrowers typically need to provide proof of income and have a good credit score.

Predicting personal loan approval using machine learning analyses a borrower's financial data and credit history to determine the likelihood of loan approval. This can help financial institutions to make more informed decisions about which loan applications to approve and which to deny.



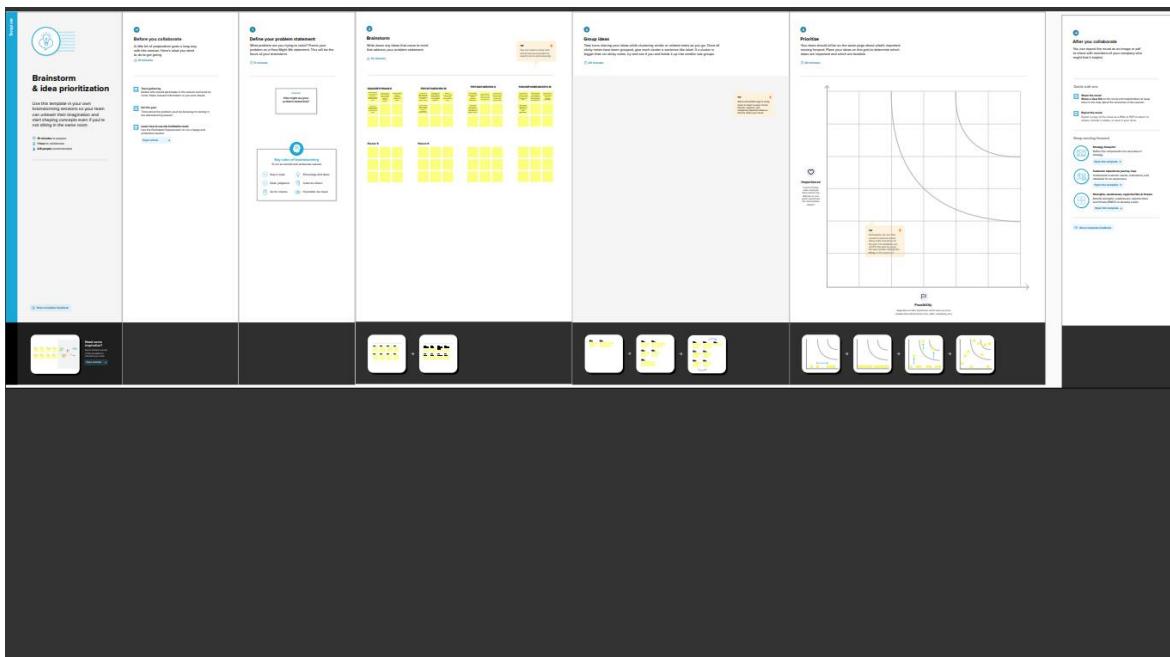
2. Problem Definition & Design Thinking

2.1 Empathy Map



The image shows a template for an Empathy Map canvas. On the left, there's a sidebar with a heart icon and the title "Empathy map canvas". Below it is a note: "Use this framework to empathize with a customer, user, or any person who is affected by a team's work. Document your findings, review your observations and note your assumptions to gain more empathy for the people you serve." At the bottom of the sidebar is a button labeled "Show response feedback". In the center, there's a large grid divided into four quadrants: "What are they **FEELING**" (top-left), "What are they **THINKING** and **FEELING**" (top-right), "What are they **NEEDING**" (bottom-right), and "What are they **SAYING**" (bottom-left). The top of the grid has a section titled "Develop shared understanding and empathy" with the sub-instruction: "Summarize the data you have gathered related to the people that are impacted by your work. It will help you generate ideas, prioritize features, or design decisions!" Below the main grid is a navigation bar with three icons: a heart, a brain, and a person.

2.2 Ideation & Brainstorm Map



The image shows a template for "Brainstorm & idea prioritization". It consists of several panels: "Before you collaborate" (with tips like "Establishing ground rules for your brainstorming session can help ensure that everyone is on the same page"), "Define your problem statement" (with a box for "Problem statement" and a "Define your problem statement" button), "Brainstorm" (with a 4x4 grid for "Initial ideas" and a "Prioritize" section), "Group Ideas" (with a "Group ideas" section), "Prioritize" (with a "Prioritize" section and a graph showing "Importance" vs "Priority"), and "After you collaborate" (with a "Sync up notes" section). At the bottom, there are two rows of small icons representing different steps in the process.



Project Report Template

3.RESULT

Untitled0.ipynb - Colaboratory

File Edit View Insert Runtime Tools Help All changes saved

Files

(x) sample_data

test_Y3wMUE5_7gLdaTN.csv

train_u6lujuX_CVtuZ9!(2).xls

+ Code + Text

```
import pandas as pd
import numpy as np
import pickle
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
import sklearn
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import GradientBoostingClassifier,RandomForestClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.model_selection import RandomizedSearchCV
import imblearn
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix, f1_score
```

data = pd.read_csv('loan_prediction.csv')

data

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	CooapplicantIncome	LoanAmount	Loan_Status
0	LP001002	Male	No	0	Graduate	No	5849	0.0	Nan	NaN
1	LP001003	Male	Yes	1	Graduate	No	4583	1508.0	128.0	
2	LP001005	Male	Yes	0	Graduate	Yes	3000	0.0	66.0	

0s completed at 12:26 AM

Type here to search

37°C Sunny ENG IN 12:29 AM 4/22/2023

Untitled0.ipynb - Colaboratory

File Edit View Insert Runtime Tools Help All changes saved

Files

(x) sample_data

test_Y3wMUE5_7gLdaTN.csv

train_u6lujuX_CVtuZ9!(2).xls

+ Code + Text

```
data = pd.read_csv('loan_prediction.csv')
```

data

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	CooapplicantIncome	LoanAmount	Loan_Status
0	LP001002	Male	No	0	Graduate	No	5849	0.0	Nan	NaN
1	LP001003	Male	Yes	1	Graduate	No	4583	1508.0	128.0	
2	LP001005	Male	Yes	0	Graduate	Yes	3000	0.0	66.0	
3	LP001006	Male	Yes	0	Not Graduate	No	2583	2358.0	120.0	
4	LP001008	Male	No	0	Graduate	No	6000	0.0	141.0	
...
609	LP002978	Female	No	0	Graduate	No	2900	0.0	71.0	
610	LP002979	Male	Yes	3+	Graduate	No	4106	0.0	40.0	
611	LP002983	Male	Yes	1	Graduate	No	6072	240.0	253.0	
612	LP002984	Male	Yes	2	Graduate	No	7583	0.0	187.0	
613	LP002990	Female	No	0	Graduate	Yes	4583	0.0	133.0	

614 rows x 13 columns

0s completed at 12:26 AM

Type here to search

37°C Sunny ENG IN 12:29 AM 4/22/2023



Project Report Template

The screenshot shows a Google Colab notebook titled "Untitled0.ipynb". The code cell contains the following Python code:

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 614 entries, 0 to 613
Data columns (total 13 columns):
 #   Column      Non-Null Count  Dtype  
--- 
 0   Loan_ID     614 non-null    object  
 1   Gender      601 non-null    object  
 2   Married     611 non-null    object  
 3   Dependents  599 non-null    object  
 4   Education   614 non-null    object  
 5   Self_Employed 582 non-null    object  
 6   ApplicantIncome 614 non-null    int64  
 7   CoapplicantIncome 614 non-null    float64 
 8   LoanAmount   592 non-null    float64 
 9   Loan_Amount_Term 600 non-null    float64 
 10  Credit_History 564 non-null    float64 
 11  Property_Area 614 non-null    object  
 12  Loan_Status  614 non-null    object  
dtypes: float64(4), int64(1), object(8)
memory usage: 62.5+ KB
```

```
data.isnull().sum()
```

The output of the second cell shows the count of null values for each column:

Column	Count
Loan_ID	0
Gender	13
Married	3

The screenshot shows a Google Colab notebook titled "Untitled0.ipynb". The code cell contains the following Python code:

```
data.isnull().sum()
```

The output of the first part of the cell shows the count of null values for each column:

Column	Count
Loan_ID	0
Gender	13
Married	3
Dependents	15
Education	0
Self_Employed	32
ApplicantIncome	0
CoapplicantIncome	0
LoanAmount	22
Loan_Amount_Term	14
Credit_History	50
Property_Area	0
Loan_Status	0

```
[ ] print(y.value_counts())
print(y_bal.value_counts())
```

The output of the second part of the cell shows a NameError:

```
NameError: Traceback (most recent call last)
<ipython-input-5-04921ec55d8e> in <cell line: 1>()
----> 1 print(y.value_counts())
      2 print(y_bal.value_counts())

NameError: name 'y' is not defined
```



Project Report Template

Untitled0.ipynb - Colaboratory

File Edit View Insert Runtime Tools Help All changes saved

Files

{x} sample_data test_Y3wMUE5_7gLdaTN.csv train_u6lujuX_CVtuZ9i (2).xls

+ Code + Text

```
from imblearn.combine import SMOTETomek
```

```
[ ] data.describe()
```

	ApplicantIncome	CoplicantIncome	LoanAmount	Loan_Amount_Term	Credit_History
count	614.000000	614.000000	592.000000	600.00000	564.000000
mean	5403.459283	1621.245798	146.412162	342.00000	0.842199
std	6109.041673	2926.248369	85.587325	65.12041	0.364878
min	150.000000	0.000000	9.000000	12.00000	0.000000
25%	2877.500000	0.000000	100.000000	360.00000	1.000000
50%	3812.500000	1188.500000	128.000000	360.00000	1.000000
75%	5795.000000	2297.250000	168.000000	360.00000	1.000000
max	81000.000000	41667.000000	700.000000	480.00000	1.000000

```
[ ] plt.figure(figsize=(12,5))
plt.subplot(121)
sns.distplot(data['ApplicantIncome'], color='r')
plt.subplot(122)
sns.distplot(data['Credit_History'])
```

0s completed at 12:26 AM

Type here to search 37°C Sunny ENG 12:31 AM IN 4/22/2023

Untitled0.ipynb - Colaboratory

File Edit View Insert Runtime Tools Help All changes saved

Files

{x} sample_data test_Y3wMUE5_7gLdaTN.csv train_u6lujuX_CVtuZ9i (2).xls

+ Code + Text

```
sns.distplot(data['Credit_History'])
plt.show()
```

```
<ipython-input-13-4b78f43a4171>:3: UserWarning:
```

'distplot' is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either 'displot' (a figure-level function with similar flexibility) or 'histplot' (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see <https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751>

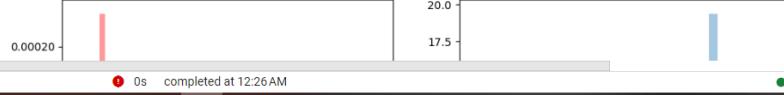
```
sns.distplot(data['ApplicantIncome'], color='r')
<ipython-input-13-4b78f43a4171>:5: UserWarning:
```

'distplot' is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either 'displot' (a figure-level function with similar flexibility) or 'histplot' (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see <https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751>

```
sns.distplot(data['Credit_History'])
```

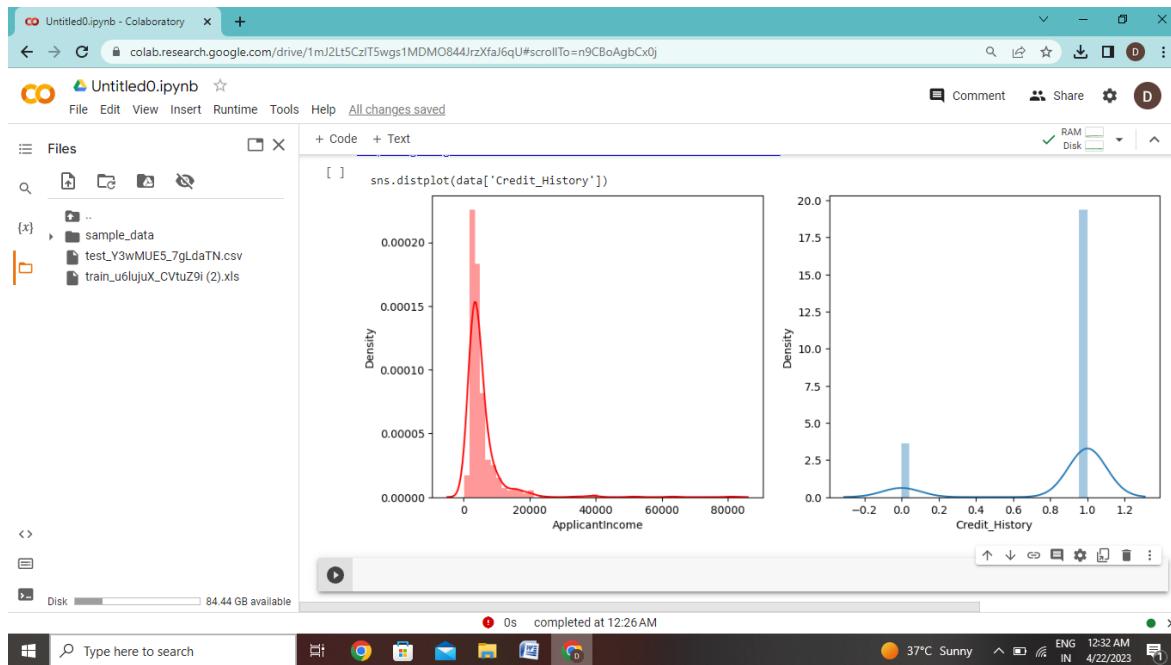


0s completed at 12:26 AM

Type here to search 37°C Sunny ENG 12:31 AM IN 4/22/2023



Project Report Template



```
#plotting the count plot
plt.figure(figsize=(18,4))
plt.subplot(1,4,1)
sns.countplot(data['Gender'])
plt.subplot(1,4,2)
sns.countplot(data['Education'])
plt.show()

[Err] NameError: name 'plt' is not defined
```

The screenshot shows a Jupyter notebook cell in Google Colab. The user has attempted to run the following Python code:

```
#plotting the count plot
plt.figure(figsize=(18,4))
plt.subplot(1,4,1)
sns.countplot(data['Gender'])
plt.subplot(1,4,2)
sns.countplot(data['Education'])
plt.show()
```

An error message is displayed: "NameError: name 'plt' is not defined". Below the error, there is a "SEARCH STACK OVERFLOW" button. The notebook also lists other cells and files in the sidebar.



Project Report Template

The screenshot shows a Google Colab notebook titled "Untitled0.ipynb". The code cell contains several plotting commands using the sns library:

```
sns.countplot(data['married'], hue=data['gender'])
plt.subplot(132)
sns.countplot(data['self_employed'],hue=data['education'])
plt.subplot(133)
sns.countplot(data['property_Area'],hue=data['Loan_Amount_term'])

-----  
NameError: name 'plt' is not defined
```

Another cell shows a similar error:

```
[ ] sns.swarmplot(data['Gender'],data['ApplicantIncome'], hue = data['loan_status'])  
-----  
NameError: name 'sns' is not defined
```

The status bar at the bottom indicates the notebook completed at 12:26 AM.

The screenshot shows a Google Colab notebook titled "Untitled0.ipynb". The code cell contains data manipulation code:

```
[ ] y = data['Loan_Status']
x = data.drop(columns=['Loan_Status'],axis=1)

-----  
NameError: name 'data' is not defined
```

Another cell shows a similar error:

```
[ ] model_history = classifier.fit(x_train, y_train, batch_size=100,validation_split=0.2, epochs=100)  
-----  
NameError: name 'classifier' is not defined
```

The status bar at the bottom indicates the notebook completed at 12:34 AM.



Project Report Template

```
[ ] x_train, x_test, y_train, y_test= train_test_split(x_bal, y_bal, test_size=0.33, random_state=42)

NameError: name 'train_test_split' is not defined

[ ] def decisionTree(x_train, x_test, y_train, y_test):
    dt=DecisiontreeClassifier()
    dt.fit(x_train,y_train)
    ypred= dt.predict(x_test)
    print('***DecisiontreeClassifier***')
    print('Confusion matrix')
    print(confusion_matrix(y_test,ypred))
    print('Classification report')
    print(classification_report(y_test,ypred))

File "<ipython-input-3-8bbb464be328>", line 1
    def decisionTree (x_train, x_test, y_train, y_test)
                                         ^
SyntaxError: invalid syntax
```

```
[ ] def RandomForestClassifiest(x_train, x_test, y_train, y_test):
    rf =RandomForestClassifier()
    rf .fit(x_train,y_train)
    ypred= rf.predict(x_test)
    print('***RandomForestClassifier***')
    print('Confusion matrix')
    print(confusion_matrix(y_test,ypred))
    print('Classification report')
    print(classification_report(y_test,ypred))

[ ] #printing the values of y before balancing the data and after
print(y.value_counts())
print(y_bal.value_counts())

NameError: name 'y' is not defined
```



Project Report Template

The screenshot shows a Google Colab notebook titled "Untitled0.ipynb". The code cell contains two functions: KNN and xgboost. The KNN function imports KNeighborsClassifier from sklearn.neighbors, fits it to training data, and prints classification reports. The xgboost function imports GradientBoostingClassifier from sklearn.ensemble, fits it to training data, and prints classification reports. A second code cell at the bottom imports tensorflow and keras Sequential models.

```
[ ] def KNN(x_train, x_test, y_train, y_test):
    knn = KNeighborsClassifier()
    knn.fit(x_train,y_train)
    yPred = knn.predict(x_test)
    print('***KNeighborsClassifier***')
    print('Confusion matrix')
    print(confusion_matrix(y_test,yPred))
    print('Classification report')
    print(classification_report(y_test,yPred))

[ ] def xgboost(x_train, x_test, y_train, y_test):
    xg = GradientBoostingClassifier()
    xg.fit(x_train,y_train)
    ypred = xg.predict(x_test)
    print('***GradientBoostingClassifier***')
    print('Confusion matrix')
    print(confusion_matrix(y_test,ypred))
    print('Classification report')
    print(classification_report(y_test,ypred))

[ ] # Importing the keras libraries and packages
import tensorflow
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
```

The screenshot shows a Google Colab notebook titled "Untitled0.ipynb". The code cell imports tensorflow and keras Sequential models, defines a Sequential classifier, adds an input layer, two hidden layers (100 and 50 units), and a sigmoid output layer. It then compiles the model with adam optimizer, binary_crossentropy loss, and accuracy metric. Finally, it fits the model to training data with a validation split of 0.2 and 100 epochs. A NameError traceback is shown at the end of the execution.

```
[ ] # Importing the keras libraries and packages
import tensorflow
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense

[ ] classifier = Sequential()

[ ] # Adding the input layer and the first hidden layer
classifier.add(Dense(units=100, activation='relu',input_dim=11))

[ ] classifier.add(Dense(units=50, activation='relu'))

[ ] classifier.add(Dense(units=1, activation='sigmoid'))

[ ] classifier.compile(optimizer='adam',loss='binary_crossentropy',metrics=['accuracy'])

[ ] model_history = classifier.fit(x_train, y_train, batch_size=100, validation_split=0.2, epochs=100)

-----
NameError: Traceback (most recent call last)
<ipython-input-24-e8bc7f4e494f> in <cell line: 1>
----> 1 model_history = classifier.fit(x_train, y_train, batch_size=100, validation_split=0.2, epochs=100)
```



Project Report Template

The screenshot shows two consecutive code cells in a Google Colab notebook. Both cells contain the same line of code: `y_pred = classifier.predict(X_test)`. The first cell results in a `NameError: name 'x_train' is not defined` because the variable `x_train` has not been defined. The second cell results in a `NameError: name 'classifier' is not defined` because the classifier object has not been defined. The user has attempted to search for the error on Stack Overflow by clicking the "SEARCH STACK OVERFLOW" button.

```
[ ] NameError: name 'x_train' is not defined
[ ] SEARCH STACK OVERFLOW

[ ] y_pred = classifier.predict(X_test)

[ ] -----
[ ] NameError: name 'x_train' is not defined
[ ] Traceback (most recent call last)
[ ] <ipython-input-18-20469ee708c2> in <cell line: 1>()
[ ]     y_pred = classifier.predict(X_test)

[ ] NameError: name 'classifier' is not defined
[ ] SEARCH STACK OVERFLOW

[ ] [237] y_pred
[ ] -----
[ ] File "<ipython-input-19-6081b40c1724>", line 1
[ ]     [237] y_pred
[ ]         ^
[ ] SyntaxError: invalid syntax
[ ] SEARCH STACK OVERFLOW

[ ] [238] y_pred = (y_pred > 0.5)
[ ] -----
[ ] 0s completed at 12:26 AM
```

The bottom part of the screenshot shows the Windows taskbar with the search bar containing "Type here to search". The system tray indicates it's 38°C, sunny, and the date is 4/22/2023.

The screenshot shows a third code cell in the notebook. The user has modified the code to `y_pred = (y_pred > 0.5)`. This results in a `SyntaxError: invalid syntax`. The user has again attempted to search for the error on Stack Overflow by clicking the "SEARCH STACK OVERFLOW" button.

```
[ ] [238] y_pred = (y_pred > 0.5)
[ ] -----
[ ] File "<ipython-input-17-1c05ff2a0234>", line 1
[ ]     [238] y_pred = (y_pred > 0.5)
[ ]         ^
[ ] SyntaxError: invalid syntax
[ ] SEARCH STACK OVERFLOW

[ ] [244] def predict_exit(sample_value):
[ ] -----
[ ] File "<ipython-input-3-4812d23bfba8>", line 1
[ ]     [244] def predict_exit(sample_value):
[ ]         ^
[ ] SyntaxError: invalid syntax
[ ] SEARCH STACK OVERFLOW

[ ] -----
[ ] sample_value = np.array(sample_value)
[ ] -----
[ ] NameError: name 'np' is not defined
[ ] Traceback (most recent call last)
[ ] <ipython-input-13-750e45c05bb8> in <cell line: 1>()
[ ]     sample_value = np.array(sample_value)
[ ]     ^
[ ] NameError: name 'np' is not defined
[ ] -----
[ ] 0s completed at 12:26 AM
```

The bottom part of the screenshot shows the Windows taskbar with the search bar containing "Type here to search". The system tray indicates it's 38°C, sunny, and the date is 4/22/2023.



Project Report Template

```
[ ] # Predictions
# Value Order 'CreditScore','Age','Tenure','Balance','NumoOfProducts','HasCrCard','IsActiveMember','EstimatedSalary','France','Germany'
sample_value = [[1,1, 0, 1, 1, 4276, 1542,145, 240, 1,1]]
if predict_exit(sample_value)>0.5:
    print('Prediction: High chance of Loan Approval!')
else:
    print('Prediction: Low chance of Loan Approval.')

-----
NameError: Traceback (most recent call last)
<ipython-input-9-c52150fce86> in <cell line: 4>()
      2 # Value Order
      'CreditScore','Age','Tenure','Balance','NumoOfProducts','HasCrCard','IsActiveMember','EstimatedSalary','France','Germany',
      3 sample_value = [[1,1, 0, 1, 1, 4276, 1542,145, 240, 1,1]]
----> 4 if predict_exit(sample_value)>0.5:
      5     print('Prediction: High chance of Loan Approval!')
      6 else:

NameError: name 'predict_exit' is not defined

SEARCH STACK OVERFLOW
```

```
[ ] def compareModel(x_train,X_test,y_train,y_test):
| decisionTree(x_train,x_test,y_train,y_test)
| print('*'*100)
```

0s completed at 12:26 AM

Type here to search 38°C Sunny ENG 12:42 AM IN 4/22/2023


```
[ ] def compareModel(x_train,X_test,y_train,y_test):
| decisionTree(x_train,x_test,y_train,y_test)
| print('*'*100)
| RandomForest(x_train,x_test,y_train,y_test)
| print('*'*100)
| XGB(x_train,x_test,y_train,y_test)
| print('*'*100)
| KNN(x_train,x_test,y_train,y_test)
| print('*'*100)

[ ] compareModel(x_train,x_test,y_train,y_test)

-----
NameError: Traceback (most recent call last)
<ipython-input-22-3eb19b242d41> in <cell line: 1>()
----> 1 compareModel(x_train,x_test,y_train,y_test)

NameError: name 'x_train' is not defined

SEARCH STACK OVERFLOW
```

```
[ ] yPred = classifier.predict(x_test)
print(accuracy_score(y_pred,y_test))
print("ANN Model")
```

0s completed at 12:26 AM

Type here to search 38°C Sunny ENG 12:42 AM IN 4/22/2023



Project Report Template

```
RandomForest(x_train,x_test,y_train,y_test)
print('*'*100)
XGB(x_train,x_test,y_train,y_test)
print('*'*100)
KNN(x_train,x_test,y_train,y_test)
print('*'*100)

[ ] compareModel(x_train,x_test,y_train,y_test)

-----
NameError: Traceback (most recent call last)
<ipython-input-22-3eb19b242d41> in <cell line: 1>()
      1 compareModel(x_train,x_test,y_train,y_test)

NameError: name 'x_train' is not defined

SEARCH STACK OVERFLOW
yPred = classifier.predict(x_test)
print(accuracy_score(y_pred,y_test))
print("ANN Model")
print("Confusion_Matrix")
print(confusion_matrix(y_test,y_pred))
print("Classification_Report")
print(classification_report(y_test,y_pred))
```

4. ADVANTAGES & DISADVANTAGES

ADVANTAGE:

Accuracy—one of the primary benefits of using machine learning for credit scoring is **its accuracy**. Unlike human manual processing, ML-based models are automated and less likely to make mistakes. This means that loan processing becomes not only faster but more accurate, too, cutting costs on the whole.

DISADVANTAGE:

- damage Interest rates can be higher than alternatives.
- More eligibility requirements.
- Fees and penalties can be high.
- Additional monthly payment.
- Increased debt load.
- Higher payments than credit cards.

5. APPLICATIONS

The objective of the data is to use Gender, Marital Status, Education, Number of Dependents, Income, Loan Amount, Credit History, and other factors and predict the approval probability of each application.

6.CONCLUSION

From a proper analysis of positive points and constraints on the member, it can be safely concluded that the product is a considerably productive member. This use is working duly and meeting to all Banker requisites. This member can be freely plugged in numerous other systems. There have been mathematics cases of computer glitches, violations in content and most important weight of features is fixed in automated prophecy system, so in the near future the so – called software could be made more secure, trustworthy and dynamic weight conformation. In near future this module of prophecy can be integrated with the module of automated processing system. The system is trained on old training dataset in future software can be made resembling that new testing date should also take part in training data after some fix time.

7.FUTURE SCOPE

In this report author has used Random Forest approach for building a model. In this report two or more classifiers are combined together and identify a perfect model for loan prediction. Ensemble method compares two or more models and identifies a perfect model from two or more models for better loan prediction which makes banking sector to make a right choice for approval of loan application.



Project Report Template

8.APPENDIX

The screenshot shows a Google Colab notebook titled "Untitled0.ipynb". The code cell contains imports for pandas, numpy, pickle, matplotlib.pyplot, seaborn, and various sklearn classifiers and utilities. The output cell shows the loading of a CSV file named "loan_prediction.csv" and displays its first two rows:

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	CooapplicantIncome	LoanAmount	Loan_Status
0	LP001002	Male	No	0	Graduate	No	5849	0.0	Nan	Approved
1	LP001003	Male	Yes	1	Graduate	No	4583	1508.0	128.0	Approved

The status bar at the bottom indicates the operation completed at 12:26 AM.

The screenshot shows a Google Colab notebook titled "Untitled0.ipynb". The code cell contains imports for pandas and the CSV file "loan_prediction.csv". The output cell displays the full dataset, indicating 614 rows and 13 columns. The data structure is identical to the one shown in the previous screenshot, with the last column "Loan_Status" being inferred as "Approved" for all entries.

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	CooapplicantIncome	LoanAmount	Loan_Status
0	LP001002	Male	No	0	Graduate	No	5849	0.0	Nan	Approved
1	LP001003	Male	Yes	1	Graduate	No	4583	1508.0	128.0	Approved
2	LP001005	Male	Yes	0	Graduate	Yes	3000	0.0	66.0	Approved
3	LP001006	Male	Yes	0	Not Graduate	No	2583	2358.0	120.0	Approved
4	LP001008	Male	No	0	Graduate	No	6000	0.0	141.0	Approved
...
609	LP002978	Female	No	0	Graduate	No	2900	0.0	71.0	Approved
610	LP002979	Male	Yes	3+	Graduate	No	4106	0.0	40.0	Approved
611	LP002983	Male	Yes	1	Graduate	No	8072	240.0	253.0	Approved
612	LP002984	Male	Yes	2	Graduate	No	7583	0.0	187.0	Approved
613	LP002990	Female	No	0	Graduate	Yes	4583	0.0	133.0	Approved

The status bar at the bottom indicates the operation completed at 12:26 AM.



Project Report Template

Untitled0.ipynb - Colaboratory

File Edit View Insert Runtime Tools Help All changes saved

Comment Share Reconnect

Files

Connecting to a runtime to enable file browsing.

```
[ ] data.info()
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 614 entries, 0 to 613
Data columns (total 13 columns):
 #   Column           Non-Null Count  Dtype  
--- 
 0   Loan_ID          614 non-null    object  
 1   Gender           601 non-null    object  
 2   Married          611 non-null    object  
 3   Dependents       599 non-null    object  
 4   Education        614 non-null    object  
 5   Self_Employed    582 non-null    object  
 6   ApplicantIncome  614 non-null    int64  
 7   CoapplicantIncome 614 non-null    float64 
 8   LoanAmount       592 non-null    float64 
 9   Loan_Amount_Term 600 non-null    float64 
 10  Credit_History   564 non-null    float64 
 11  Property_Area    614 non-null    object  
 12  Loan_Status       614 non-null    object  
dtypes: float64(4), int64(1), object(8)
memory usage: 62.5+ KB
```

```
[ ] data.isnull().sum()
```

0s completed at 12:26 AM

Type here to search

36°C Mostly sunny ENG 1:44 AM IN 4/22/2023

Untitled0.ipynb - Colaboratory

File Edit View Insert Runtime Tools Help All changes saved

Comment Share Reconnect

Files

Connecting to a runtime to enable file browsing.

```
[ ] data.isnull().sum()
Loan_ID          0
Gender           13
Married          3
Dependents       15
Education         0
Self_Employed    32
ApplicantIncome   0
CoapplicantIncome 0
LoanAmount        22
Loan_Amount_Term 14
Credit_History    50
Property_Area     0
Loan_Status        0
dtype: int64
```

```
[ ] print(y.value_counts())
print(y_bal.value_counts())

-----
```

```
NameError: Traceback (most recent call last)
<ipython-input-5-04921ec55d8e> in <cell line: 1>()
----> 1 print(y.value_counts())
      2 print(y_bal.value_counts())

NameError: name 'y' is not defined
```

0s completed at 12:26 AM

Type here to search

36°C Mostly sunny ENG 1:45 AM IN 4/22/2023



Project Report Template

Untitled0.ipynb - Colaboratory +

File Edit View Insert Runtime Tools Help All changes saved

Files

Connecting to a runtime to enable file browsing.

```
[ ] from imblearn.combine import SMOTETomek
```

```
[ ] data.describe()
```

	ApplicantIncome	CoefficientIncome	LoanAmount	Loan_Amount_Term	Credit_History
count	614.000000	614.000000	592.000000	600.000000	564.000000
mean	5403.459283	1621.245798	146.412162	342.000000	0.842199
std	6109.041673	2926.248369	85.587325	65.12041	0.364678
min	150.000000	0.000000	9.000000	12.00000	0.000000
25%	2877.500000	0.000000	100.000000	360.000000	1.000000
50%	3812.500000	1188.500000	128.000000	360.000000	1.000000
75%	5795.000000	2297.250000	168.000000	360.000000	1.000000
max	81000.000000	41667.000000	700.000000	480.000000	1.000000

```
[ ] plt.figure(figsize=(12,5))
plt.subplot(121)
sns.distplot(data['ApplicantIncome'], color='r')
plt.subplot(122)
sns.distplot(data['Credit_History'])
plt.show()
```

0s completed at 12:26 AM

Type here to search

36°C Mostly sunny 1:45 AM ENG IN 4/22/2023

Untitled0.ipynb - Colaboratory +

File Edit View Insert Runtime Tools Help All changes saved

Files

Connecting to a runtime to enable file browsing.

```
[ ] <ipython-input-13-4b78f43a4171>:3: UserWarning:
```

```
'distplot' is a deprecated function and will be removed in seaborn v0.14.0.
```

```
Please adapt your code to use either 'displot' (a figure-level function with similar flexibility) or 'histplot' (an axes-level function for histograms).
```

```
For a guide to updating your code to use the new functions, please see https://github.com/mwaskom/seaborn/blob/v0.14.0/doc/user/deprecate\_distplot.rst
```

```
sns.distplot(data['ApplicantIncome'], color='r')
```

```
<ipython-input-13-4b78f43a4171>:5: UserWarning:
```

```
'distplot' is a deprecated function and will be removed in seaborn v0.14.0.
```

```
Please adapt your code to use either 'displot' (a figure-level function with similar flexibility) or 'histplot' (an axes-level function for histograms).
```

```
For a guide to updating your code to use the new functions, please see https://github.com/mwaskom/seaborn/blob/v0.14.0/doc/user/deprecate\_distplot.rst
```

```
sns.distplot(data['Credit_History'])
```

0s completed at 12:26 AM

Type here to search

36°C Mostly sunny 1:45 AM ENG IN 4/22/2023



Project Report Template

Untitled0.ipynb - Colaboratory +

File Edit View Insert Runtime Tools Help All changes saved

Comment Share Reconnect

Files Connecting to a runtime to enable file browsing.

Please adapt your code to use either 'distplot' (a figure-level function with similar flexibility) or 'histplot' (an axes-level function for histograms). For a guide to updating your code to use the new functions, please see <https://gist.github.com/maskom/de44147ed2974457ad6372750bbe5751>

```
sns.distplot(data['Credit_History'])
```

0s completed at 12:26 AM

Type here to search 36°C Mostly sunny ENG IN 1:46 AM 4/22/2023

Untitled0.ipynb - Colaboratory +

File Edit View Insert Runtime Tools Help All changes saved

Comment Share Reconnect

Files Connecting to a runtime to enable file browsing.

```
#plotting the count plot
plt.figure(figsize=(18,4))
plt.subplot(1,4,1)
sns.countplot(data['Gender'])
plt.subplot(1,4,2)
sns.countplot(data['Education'])
plt.show()
```

```
NameError: Traceback (most recent call last)
<ipython-input-1-1b5a6288f675> in <cell line: 2>()
      1 #plotting the count plot
----> 2 plt.figure(figsize=(18,4))
      3 plt.subplot(1,4,1)
      4 sns.countplot(data['Gender'])
      5 plt.subplot(1,4,2)

NameError: name 'plt' is not defined
```

SEARCH STACK OVERFLOW

```
[ ] plt.figure(figsize=(20,5))
plt.subplot(131)
sns.countplot(data['married'], hue=data['gender'])
plt.subplot(132)
sns.countplot(data['self_employed'],hue=data['education'])
plt.subplot(133)
```

0s completed at 12:26 AM

Type here to search 36°C Mostly sunny ENG IN 1:46 AM 4/22/2023



Project Report Template

The screenshot shows a Google Colab notebook titled "Untitled0.ipynb". The code cell contains the following Python code:

```
[ ] sns.countplot(data['self_employed'],hue=data['education'])
plt.subplot(131)
sns.countplot(data['property_Area'],hue=data['Lone_Amount_term'])

[ ] sns.swarmplot(data['Gender'],data['ApplicantIncome'], hue = data['loan_status'])
```

Two errors are displayed:

- NameError: name 'plt' is not defined
- NameError: name 'sns' is not defined

The Colab interface includes a "SEARCH STACK OVERFLOW" button and a status bar at the bottom indicating "0s completed at 12:26 AM".

The screenshot shows a Google Colab notebook titled "Untitled0.ipynb". The code cell contains the following Python code:

```
[ ] y = data['Loan_Status']
x = data.drop(columns=['Loan_Status'],axis=1)

[ ] model_history = classifier.fit(x_train, y_train, batch_size=100,validation_split=0.2, epochs=100)
```

Two errors are displayed:

- NameError: name 'data' is not defined
- NameError: name 'classifier' is not defined

The Colab interface includes a "SEARCH STACK OVERFLOW" button and a status bar at the bottom indicating "0s completed at 12:26 AM".



Project Report Template

Untitled0.ipynb - Colaboratory

File Edit View Insert Runtime Tools Help All changes saved

Files

Connecting to a runtime to enable file browsing.

```
[ ] x_train, x_test, y_train, y_test= train_test_split(x_bal, y_bal, test_size=0.33, random_state=42)
```

NameError: name 'train_test_split' is not defined

SEARCH STACK OVERFLOW

```
[ ] def decisionTree(x_train, x_test, y_train, y_test):
    dt=DecisiontreeClassifier()
    dt.fit(x_train,y_train)
    ypred = dt.predict(x_test)
    print('***DecisionTreeClassifier***')
    print('Confusion matrix')
    print(confusion_matrix(y_test,ypred))
    print('Classification report')
    print(classification_report(y_test,ypred))

File "<ipython-input-21-f40bc3e5743c>" line 1
      def decisionTree (x_train, x_test, y_train, y_test)
                                         ^
SyntaxError: invalid syntax
```

SEARCH STACK OVERFLOW

0s completed at 12:26 AM

Type here to search

36°C Mostly sunny ENG 1:47 AM IN 4/22/2023

Untitled0.ipynb - Colaboratory

File Edit View Insert Runtime Tools Help All changes saved

Files

Connecting to a runtime to enable file browsing.

```
[ ] def RandomForestClassifiest(x_train, x_test, y_train, y_test):
    rf =RandomForestClassifier()
    rf .fit(x_train,y_train)
    ypred= rf.predict(x_test)
    print('***RandomForestClassifier***')
    print('Confusion matrix')
    print(confusion_matrix(y_test,ypred))
    print('Classification report')
    print(classification_report(y_test,ypred))

[ ] #printing the values of y before balancing the data and after
print(y.value_counts())
print(y_bal.value_counts())
```

```
NameError: Traceback (most recent call last)
<ipython-input-13-65280d18cdbe> in <cell line: 2>()
      1 #printing the values of y before balancing the data and after
----> 2 print(y.value_counts())
      3 print(y_bal.value_counts())

NameError: name 'y' is not defined
```

SEARCH STACK OVERFLOW

0s completed at 12:26 AM

Type here to search

36°C Mostly sunny ENG 1:47 AM IN 4/22/2023



Project Report Template

The screenshot shows a Google Colab notebook titled "Untitled0.ipynb". The code cell contains two definitions: KNN and XGBoost. The KNN function imports KNeighborsClassifier from sklearn.neighbors, fits it to training data, and prints classification reports. The XGBoost function imports GradientBoostingClassifier from sklearn.ensemble, fits it to training data, and prints classification reports. A note in the code indicates that the code is connecting to a runtime to enable file browsing.

```
[ ] def KNN(x_train, x_test, y_train, y_test):
    knn = KNeighborsClassifier()
    knn.fit(x_train,y_train)
    yPred = knn.predict(x_test)
    print('***Kneighborsclassifier***')
    print('Confusion matrix')
    print(confusion_matrix(y_test,yPred))
    print('Classification report')
    print(classification_report(y_test,yPred))

[ ] def xgboost(x_train, x_test, y_train, y_test):
    xg = GradientBoostingClassifier()
    xg.fit(x_train,y_train)
    ypred = xg.predict(x_test)
    print('***GradientBoostingClassifier***')
    print('Confusion matrix')
    print(confusion_matrix(y_test,ypred))
    print('Classification report')
    print(classification_report(y_test,ypred))

[ ] # Importing the keras libraries and packages
import tensorflow
from tensorflow.keras.models import Sequential
```

The screenshot shows a Google Colab notebook titled "Untitled0.ipynb". The code cell defines a Sequential model from tensorflow.keras.models. It adds three layers: a Dense layer with 100 units and ReLU activation, another Dense layer with 50 units and ReLU activation, and a final Dense layer with 1 unit and sigmoid activation. The model is compiled with Adam optimizer, binary_crossentropy loss, and accuracy metric. A NameError occurs when attempting to fit the model with 'x_train' undefined.

```
[ ] import tensorflow
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense

[ ] classifier = Sequential()

[ ] # Adding the input layer and the first hidden layer
classifier.add(Dense(units=100, activation='relu',input_dim=11))

[ ] classifier.add(Dense(units=50, activation='relu'))

[ ] classifier.add(Dense(units=1, activation='sigmoid'))

[ ] classifier.compile(optimizer='adam',loss='binary_crossentropy',metrics=['accuracy'])

[ ] model_history = classifier.fit(x_train, y_train, batch_size=100, validation_split=0.2, epochs=100)

-----
NameError: Traceback (most recent call last)
<ipython-input-24-e8bcf74e494f> in <cell line: 1>()
----> 1 model_history = classifier.fit(x_train, y_train, batch_size=100, validation_split=0.2, epochs=100)

NameError: name 'x_train' is not defined
```



Project Report Template

Untitled0.ipynb - Colaboratory

File Edit View Insert Runtime Tools Help All changes saved

Files

Connecting to a runtime to enable file browsing.

```
[ ] y_pred = classifier.predict(X_test)

NameError: Traceback (most recent call last)
<ipython-input-18-20469ee708c2> in <cell line: 1>()
      1 y_pred = classifier.predict(X_test)

NameError: name 'classifier' is not defined
```

SEARCH STACK OVERFLOW

```
[ ] [237] y_pred

File "<ipython-input-19-6081b40c1724>", line 1
[237] y_pred
          ^
SyntaxError: invalid syntax
```

SEARCH STACK OVERFLOW

```
[ ] [238] y_pred = (y_pred > 0.5)
y_pred

File "<ipython-input-17-1c05ff2a0234>", line 1
[238] y_pred = (y_pred > 0.5)
          ^
SyntaxError: invalid syntax
```

0s completed at 12:26 AM

Type here to search

36°C Mostly sunny ENG 1:48 AM IN 4/22/2023

Untitled0.ipynb - Colaboratory

File Edit View Insert Runtime Tools Help All changes saved

Files

Connecting to a runtime to enable file browsing.

```
[ ] SyntaxError: invalid syntax
```

SEARCH STACK OVERFLOW

```
[ ] [244] def predict_exit(sample_value):

File "<ipython-input-3-4012d23fbfa8>", line 1
[244] def predict_exit(sample_value):
          ^
SyntaxError: invalid syntax
```

SEARCH STACK OVERFLOW

```
[ ] sample_value = np.array(sample_value)

sample_value = sample_value.reshape(1, -1)

NameError: Traceback (most recent call last)
<ipython-input-13-750e45c05bb8> in <cell line: 1>()
      1 sample_value = np.array(sample_value)
      2
      3 sample_value = sample_value.reshape(1, -1)

NameError: name 'np' is not defined
```

0s completed at 12:26 AM

Type here to search

36°C Mostly sunny ENG 1:48 AM IN 4/22/2023



Project Report Template

```
NameError: name 'np' is not defined
[ ] NameError: name 'predict_exit' is not defined
[ ] # Predictions
# Value Order
'CreditScore','Age','Tenure','Balance','NumOfProducts','HasCrCard','IsActiveMember','EstimatedSalary','France','Germany','Spain'
sample_value = [[1,1,0,1,1,4276,1542,145,248,1,1]]
if predict_exit(sample_value)>0.5:
    print('Prediction: High chance of Loan Approval!')
else:
    print('Prediction: Low chance of Loan Approval.')

-----
NameError: Traceback (most recent call last)
<ipython-input-9-c52150fce86> in <cell line: 4>()
      2 # Value Order
      3 sample_value = [[1,1,0,1,1,4276,1542,145,248,1,1]]
----> 4 if predict_exit(sample_value)>0.5:
      5     print('Prediction: High chance of Loan Approval!')
      6 else:

NameError: name 'predict_exit' is not defined
```

```
[ ] def compareModel(x_train,x_test,y_train,y_test):
decisionTree(x_train,x_test,y_train,y_test)
print('*'*100)
RandomForest(x_train,x_test,y_train,y_test)
print('*'*100)
XGB(x_train,x_test,y_train,y_test)
print('*'*100)
KNN(x_train,x_test,y_train,y_test)
print('*'*100)

[ ] compareModel(x_train,x_test,y_train,y_test)

-----
NameError: Traceback (most recent call last)
<ipython-input-22-3eb19b242d41> in <cell line: 1>()
----> 1 compareModel(x_train,x_test,y_train,y_test)

NameError: name 'x_train' is not defined

[ ] yPred = classifier.predict(x_test)
print(accuracy_score(y_pred,y_test))
print("ANN Model")
```



Project Report Template

The screenshot shows a Google Colab notebook titled "Untitled0.ipynb". The code cell contains the following Python code:

```
[ ] yPred = classifier.predict(x_test)
print(accuracy_score(y_pred,y_test))
print("ANN Model")
print("Confution_Matrix")
print(confusion_matrix(y_test,y_pred))
print("Classification Report")
print(classification_report(y_test,y_pred))
```

A NameError occurs at the first line of the code, indicating that the variable 'classifier' is not defined.

Below the code cell, the status bar shows "0s completed at 12:26 AM". The taskbar at the bottom of the screen includes icons for File Explorer, Google Chrome, Mail, Task View, and others, along with system information like temperature (36°C), weather (Mostly sunny), and date/time (1:49 AM IN 4/22/2023).