**SCHOOL OF COMPUTER SCIENCE ENGINEERING AND INFORMATION SYSTEMS**

**Course code: PMCA507L**

**Course Title: Machine Learning**

**LABORATORY PROJECT**
**SLOT: A2**

# TITLE: CREDIT RATING PREDICTION

**TEAM MEMBERS:**

| | |
|---|---|
| SHUBHAM SINHA | 23MCA0111 |
| ANURAG SINHA | 23MCA0079 |
| BALAR DARSHAN VAJUBHAI | 23MCA0106 |

**FACULTY INFORMTION:**

**Dr. ARUN PANDIAN J**
- **Assistant Professor Sr. Grade 1**

# TABLE OF CONTENT

# INTRODUCTION

The Credit Risk Prediction System is a data-driven solution designed to assist financial institutions in evaluating the creditworthiness of loan applicants. With the increasing complexity of financial markets and the growing demand for loans, accurately assessing credit risk is crucial for mitigating financial losses and maintaining a healthy lending portfolio. Traditional methods of credit assessment often rely on manual review processes, which can be time-consuming and prone to errors. In response to these challenges, the Credit Risk Prediction System leverages machine learning techniques to automate the credit evaluation process and provide timely, data-driven insights.

By analysing historical data of loan applicants and their corresponding credit outcomes, the system learns patterns and relationships that influence credit risk. Through the implementation of logistic regression, a widely used classification algorithm, the system predicts whether an applicant falls into the category of 'good' or 'bad' credit risk based on their financial and personal attributes. This predictive model is trained on a diverse dataset encompassing factors such as credit history, income, employment status, and more.

The Credit Risk Prediction System is deployed as a user-friendly web application, allowing financial institutions to seamlessly integrate it into their existing loan approval workflows. Through a simple and intuitive interface, users can input their details and receive instant predictions regarding their credit classification. This empowers lenders to make informed decisions regarding loan approvals, streamline their processes, and minimize the risk of default.

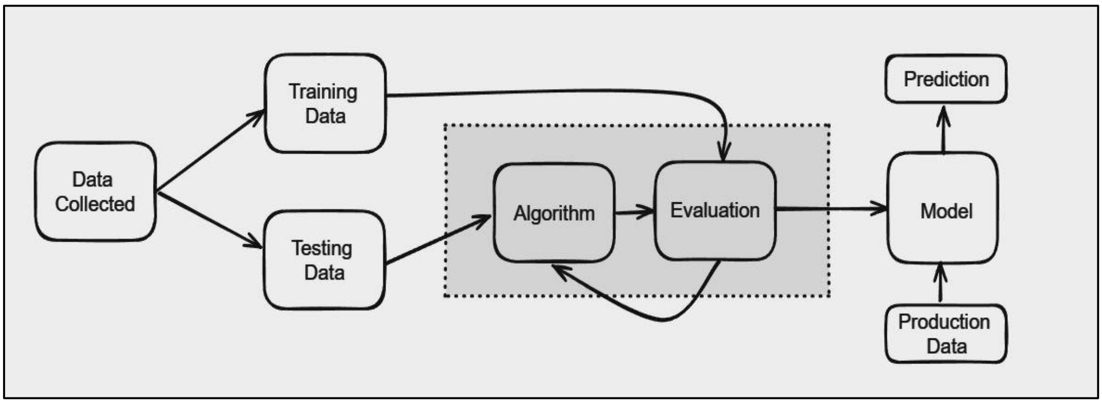
In this project, we explore the intersection of machine learning and finance, demonstrating the potential of predictive analytics in enhancing credit risk assessment. By harnessing the power of data-driven insights, the Credit Risk Prediction System aims to optimize lending practices, improve financial outcomes, and foster responsible lending practices in the ever-evolving landscape of finance.

## PROBLEM STATEMENT

Classify the customer's credit rating (good or bad) based on their personal and bank account details.

- Dataset: credit_rating.xlsx
- Target variable: Credit classification
- Constrain: Use any one of the following classification techniques: Logistic Regression,
- Decision Tree, Naïve Bayes, Support Vector Machine, and Multi-Layer Perceptron
- Performance Measure: Plot the confusion matrix, accuracy, and F1 score of the
- classification technique on test data.
- Deploy the model as a mobile or web application. Create a form for collect the user
- inputs.

## ARCHITECTURAL DIAGRAM



## RESULT & MODEL

We have developed a web-based credit card rating model that predicts the creditworthiness of applicants based on their submitted information. This project aims to provide a user-friendly interface for users to input their data, receive a credit rating prediction, and view the result conveniently.

**Technologies Used :**

- Backend Technologies:

  - **Python:** We utilized Python as the primary programming language for backend development due to its simplicity, flexibility, and extensive libraries for data processing and web development.

  - **Flask:** We chose Flask as the web framework for its lightweight nature, simplicity, and ease of integration with other libraries and frameworks. Flask allowed us to quickly develop RESTful APIs and serve web pages with minimal boilerplate code.

  - **Django:** Django, a high-level Python web framework, was used for certain components of the project, particularly for user authentication, administration panel, and database management. Django's batteries-included approach provided us with built-in solutions for common web development tasks, such as URL routing, form handling, and security features. Additionally, Django templates were utilized to render HTML files, allowing us to create dynamic web pages with reusable components.

  - **Scikit-learn:** We used Scikit-learn, a powerful machine learning library in Python, for building and training the credit card rating model. Scikit-learn provides a wide range of tools for data preprocessing, model selection, and evaluation.

- **Joblib:** Joblib was used for serializing and deserializing Python objects, particularly for saving and loading trained machine learning models.

- **Pandas**: Pandas, a popular data manipulation library in Python, was used for data handling and preprocessing tasks such as loading datasets, cleaning data, and feature engineering.

- Frontend Technologies:

  - **HTML:** We used HTML for structuring the content of web pages, including forms, tables, and other elements.
  - **SCSS:** SCSS (Sassy CSS) was chosen for styling the web pages due to its enhanced features and flexibility compared to traditional CSS.

## How to Run :

**Github Link For Code :** https://github.com/darshan-balar2400/CREDITY

**Step 1 : install virtualenv**

```
pip install virtualenv
```

**Step 2 : Create Python virtual environment**

```
virtualenv venv
```

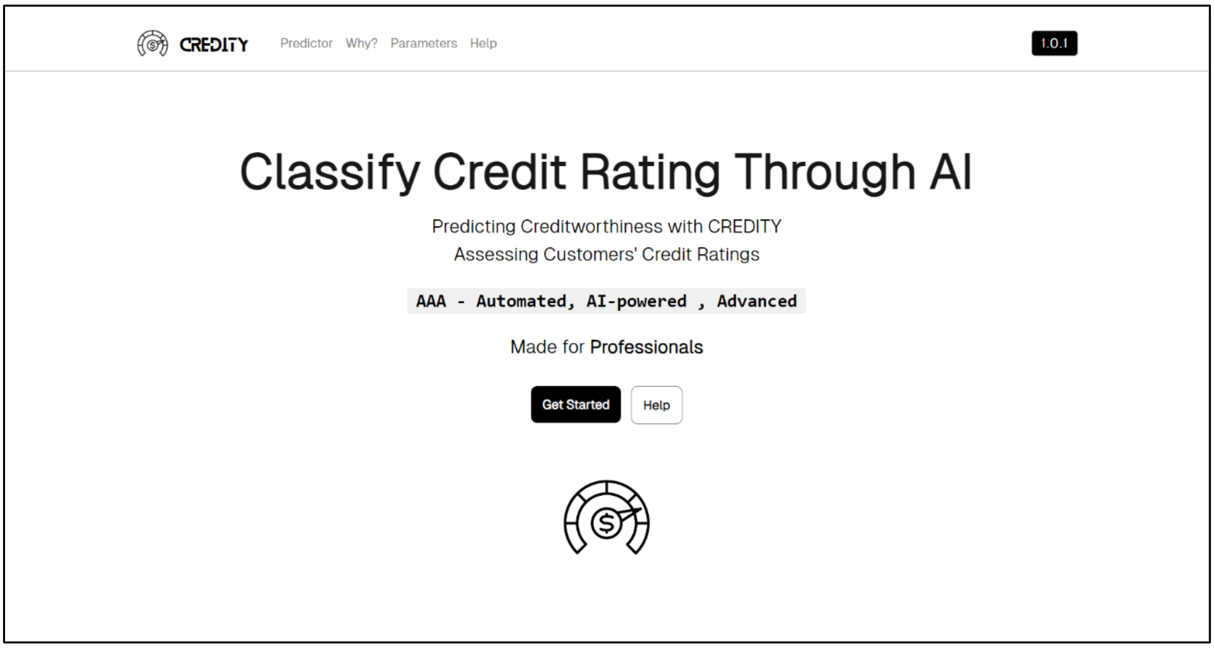
**Step 3 : Activate virtual environment**

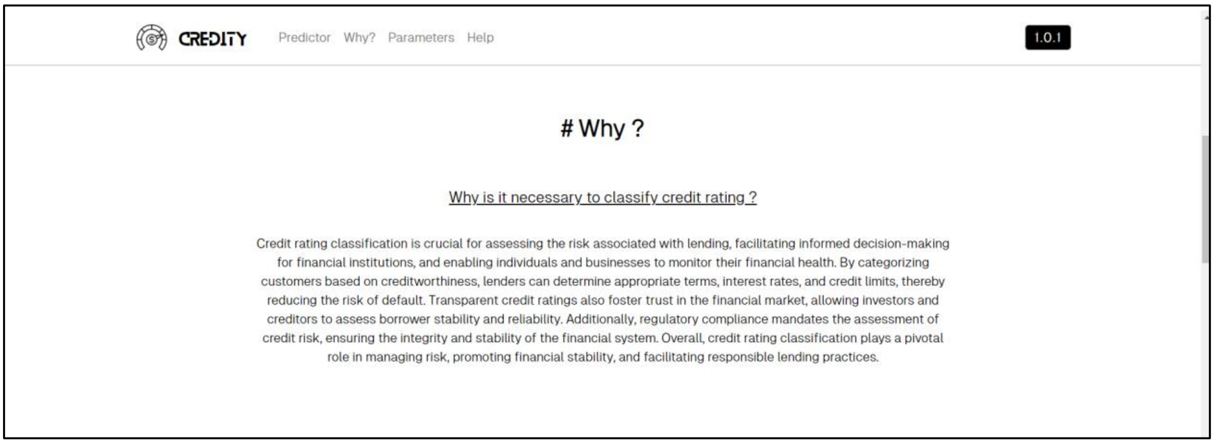
```
windows > venv\Scripts\activate

linux > source ./venv/bin/activate
```

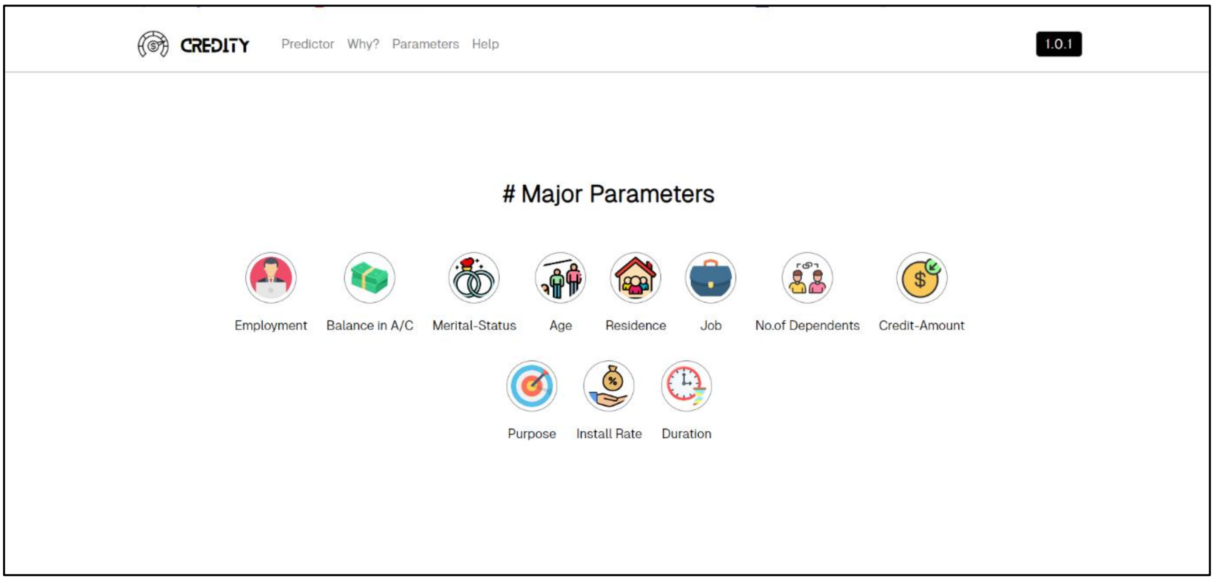
**Step 4 : After Activating Virtual Env Write Below Code**

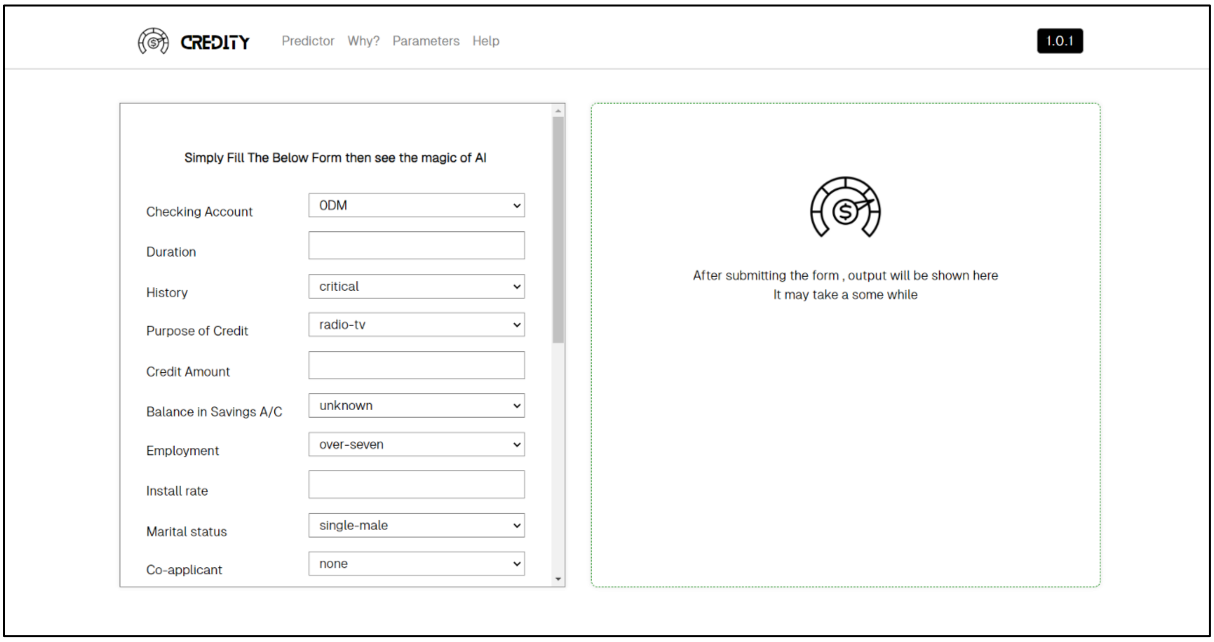
```
> python app.py
```

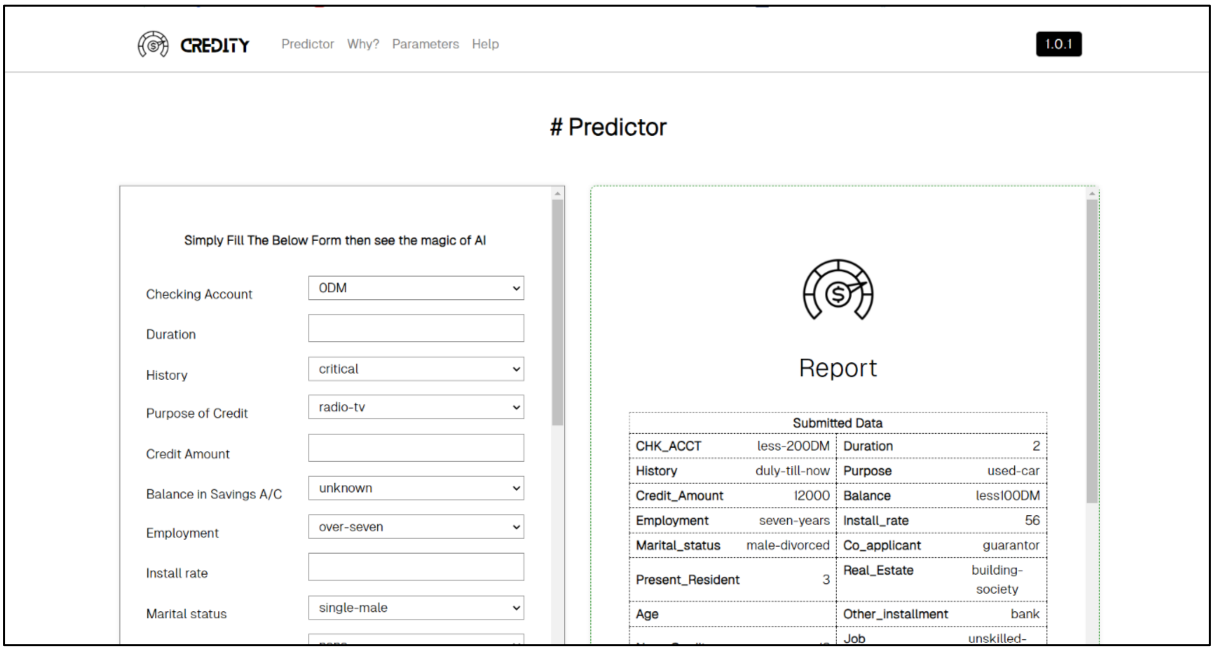
# WEB APPLICATION DEMO :

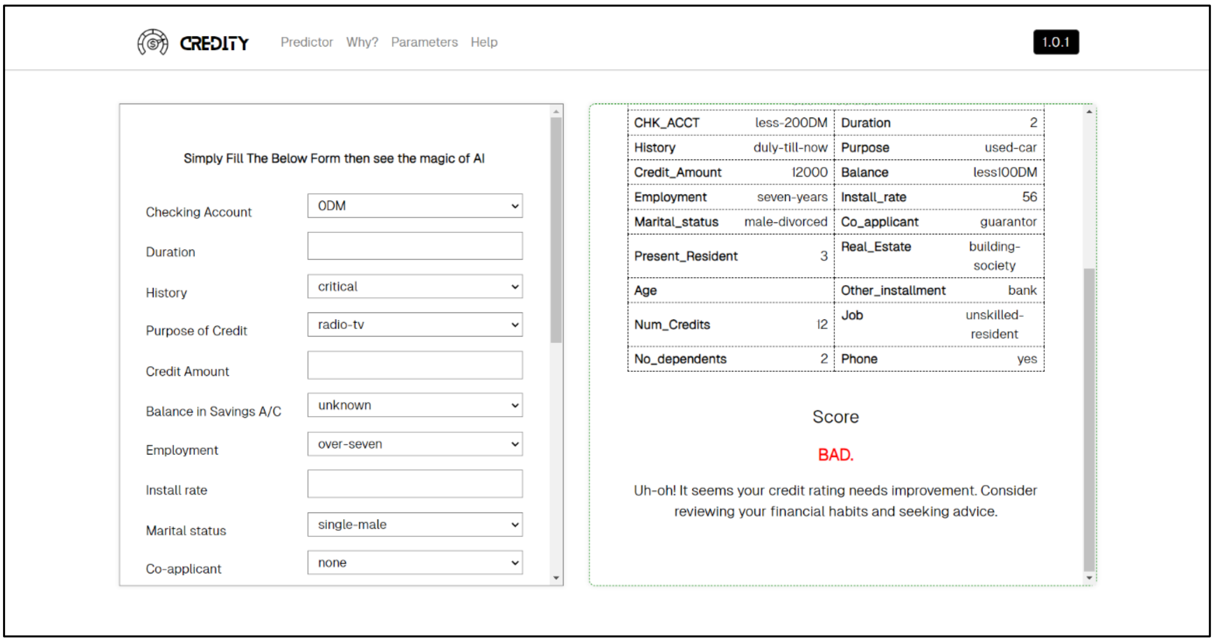


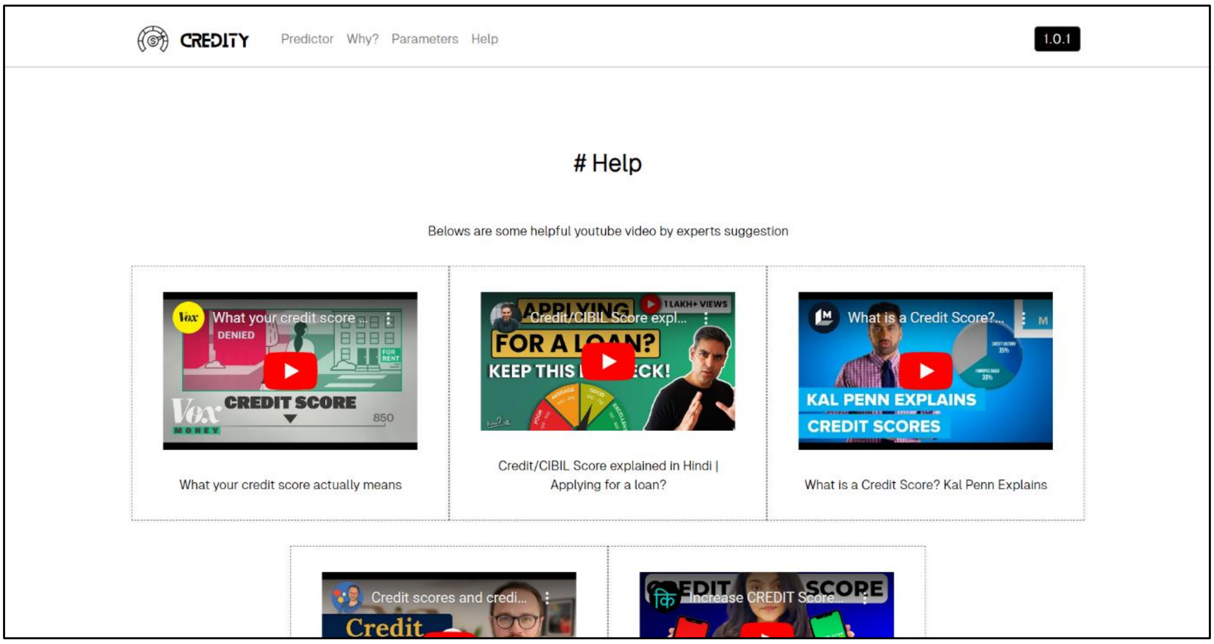



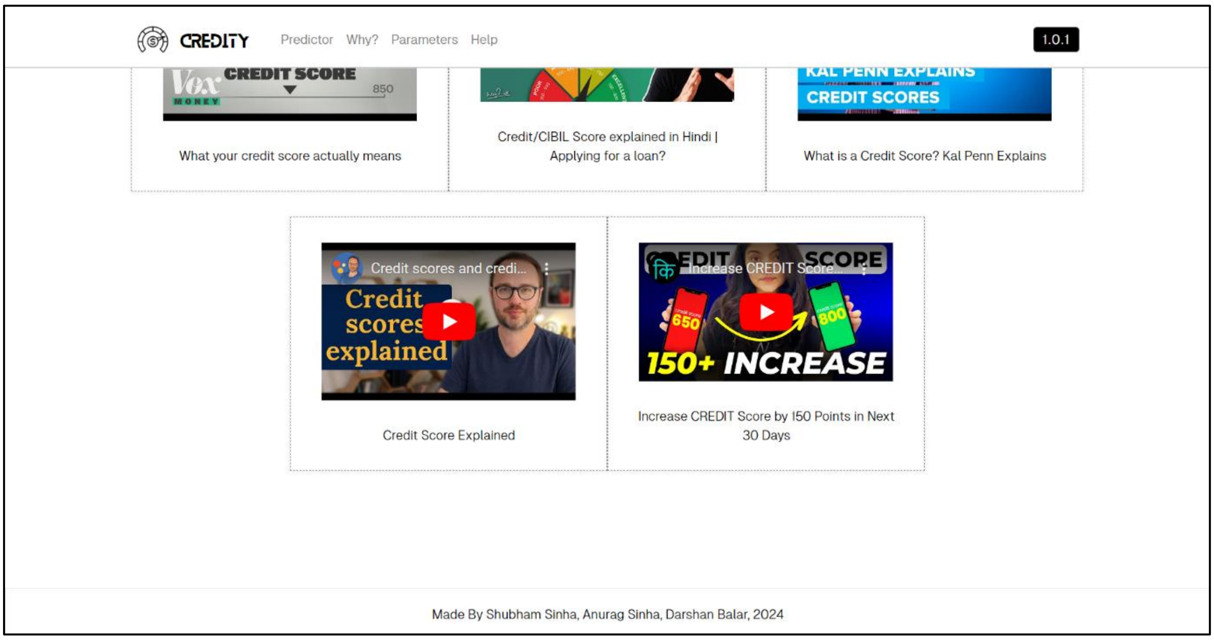

CREDITY
Predictor
Why?
Parameters
Help
1.0.1
# Major Parameters
Employment
Balance in A/C
Merital-Status
Age
Residence
Job
No.of Dependents
Credit-Amount
Purpose
Install Rate
Duration


CREDITY
Predictor
Why?
Parameters
Help
1.0.1
Simply Fill The Below Form then see the magic of AI
Checking Account
0DM
Duration
History
critical
Purpose of Credit
radio-tv
Credit Amount
Balance in Savings A/C
unknown
Employment
over-seven
Install rate
Marital status
single-male
Co-applicant
none
After submitting the form , output will be shown here
It may take a some while

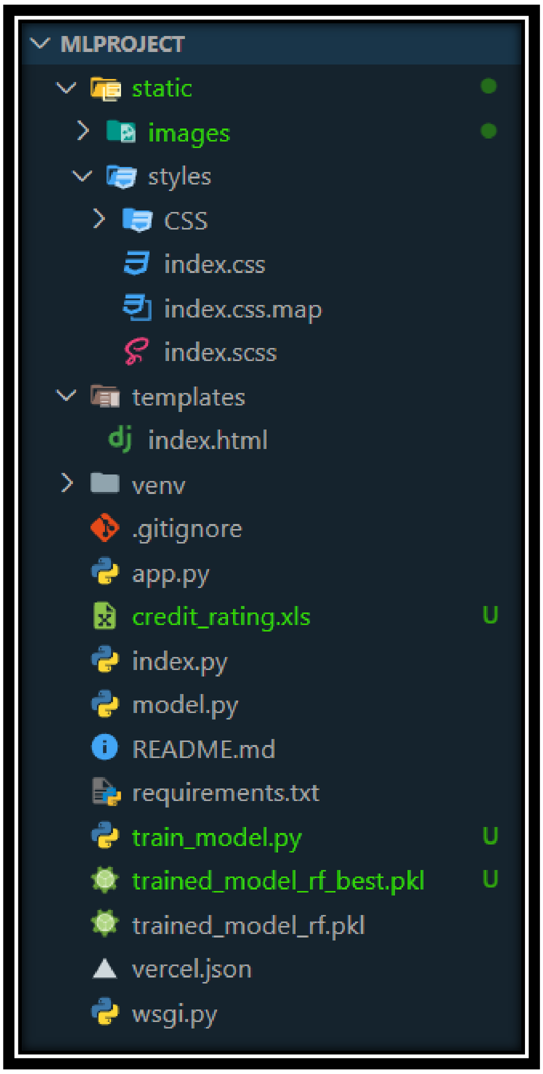

CREDITY
Predictor
Why?
Parameters
Help
1.0.1
# Predictor
Simply Fill The Below Form then see the magic of AI
Checking Account
0DM
Duration
History
critical
Purpose of Credit
radio-tv
Credit Amount
Balance in Savings A/C
unknown
Employment
over-seven
Install rate
Marital status
single-male
Report
Submitted Data
CHK_ACCT
less-200DM
Duration
2
History
duly-till-now
Purpose
used-car
Credit_Amount
12000
Balance
less100DM
Employment
seven-years
Install_rate
56
Marital_status
male-divorced
Co_applicant
guarantor
Present_Resident
3
Real_Estate
building-society
Age
Other_installment
bank
Job
unskilled-


CREDITY
Predictor
Why?
Parameters
Help
1.0.1
Simply Fill The Below Form then see the magic of AI
Checking Account
0DM
Duration
History
critical
Purpose of Credit
radio-tv
Credit Amount
Balance in Savings A/C
unknown
Employment
over-seven
Install rate
Marital status
single-male
Co-applicant
none
CHK_ACCT
less-200DM
Duration
2
History
duly-till-now
Purpose
used-car
Credit_Amount
12000
Balance
less100DM
Employment
seven-years
Install_rate
56
Marital_status
male-divorced
Co_applicant
guarantor
Present_Resident
3
Real_Estate
building-society
Age
Other_installment
bank
Num_Credits
12
Job
unskilled-resident
No_dependents
2
Phone
yes
Score
BAD.
Uh-oh! It seems your credit rating needs improvement. Consider reviewing your financial habits and seeking advice.


CREDITY
Predictor
Why?
Parameters
Help
1.0.1
# Help
Belows are some helpful youtube video by experts suggestion
What your credit score actually means
Credit/CIBIL Score explained in Hindi | Applying for a loan?
What is a Credit Score? Kal Penn Explains
KAL PENN EXPLAINS
CREDIT SCORES
Credit scores and credi...
Increase CREDIT Score...


CREDITY
Predictor
Why?
Parameters
Help
1.0.1
What your credit score actually means
Credit/CIBIL Score explained in Hindi | Applying for a loan?
What is a Credit Score? Kal Penn Explains
Credit scores and credi...
Credit scores explained
Credit Score Explained
Increase CREDIT Score...
150+ INCREASE
Increase CREDIT Score by 150 Points in Next 30 Days
Made By Shubham Sinha, Anurag Sinha, Darshan Balar, 2024

# FOLDER STRUCTURE



## train_model.py

```
import pandas as pd
from sklearn.model_selection import train_test_split, GridSearchCV
from sklearn.preprocessing import MinMaxScaler, OneHotEncoder
from sklearn.compose import ColumnTransformer
from sklearn.pipeline import Pipeline
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, confusion_matrix,
f1_score
import joblib
import matplotlib.pyplot as plt
import seaborn as sns

# Load the dataset
dataset = pd.read_excel("credit_rating.xls")

# Separate features (X) and target variable (y)
X = dataset.drop(columns=["Credit classification"])
y = dataset["Credit classification"]

# Identify categorical features
categorical_features =
X.select_dtypes(include=['object']).columns.tolist()

# Define preprocessing steps for categorical and numerical features
categorical_transformer = OneHotEncoder(handle_unknown='ignore')
numerical_transformer = MinMaxScaler()

# Get column indices for categorical features
categorical_indices = [X.columns.get_loc(col) for col in
categorical_features]

# Get column indices for numerical features
numerical_indices = [i for i in range(len(X.columns)) if i not in
categorical_indices]

# Combine preprocessing steps
preprocessor = ColumnTransformer(
    transformers=[
        ('cat', categorical_transformer, categorical_indices),
        ('num', numerical_transformer, numerical_indices)
    ])
```

```python
# Split the dataset into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.2, random_state=42)

# Define the classification model using Random Forest Classifier
classification_model = Pipeline(steps=[('preprocessor',
preprocessor),
                                       ('classifier',
RandomForestClassifier())])

# Define the hyperparameters grid
param_grid = {
    'classifier__n_estimators': [100, 200, 300], # Number of trees in
the forest
    'classifier__max_depth': [None, 10, 20], # Maximum depth of the
trees
    'classifier__min_samples_split': [2, 5, 10] # Minimum number of
samples required to split a node
}

# Perform grid search with 5-fold cross-validation
grid_search = GridSearchCV(classification_model, param_grid, cv=5,
scoring='accuracy', n_jobs=-1)

# Fit the grid search to find the best hyperparameters
grid_search.fit(X_train, y_train)

# Get the best hyperparameters and the best estimator
best_params = grid_search.best_params_
best_estimator = grid_search.best_estimator_

# Save the best estimator (model)
joblib.dump(best_estimator, "trained_model_rf_best.pkl")

# Print the best hyperparameters
print("Best Hyperparameters:", best_params)

# Predict the target variable on the test set using the best
estimator
y_pred = best_estimator.predict(X_test)

# Calculate accuracy
accuracy = accuracy_score(y_test, y_pred)

# Print the accuracy
print("Accuracy:", accuracy)
```

```python
# Plot confusion matrix
def plot_confusion_matrix(y_true, y_pred):
    cm = confusion_matrix(y_true, y_pred)
    plt.figure(figsize=(8, 6))
    sns.heatmap(cm, annot=True, fmt='g', cmap='Blues', cbar=False)
    plt.xlabel('Predicted')
    plt.ylabel('Actual')
    plt.title('Confusion Matrix')
    plt.show()

# Calculate F1 score
f1 = f1_score(y_test, y_pred, average='weighted')

# Print F1 score
print("F1 Score:", f1)

# Plot confusion matrix
plot_confusion_matrix(y_test, y_pred)
```

## App.py

```python
from flask import Flask,render_template,request
from model import load_model, predict_credit_classification

app = Flask(__name__, static_url_path='/static')

model = load_model()

@app.route('/',methods=['GET', 'POST'])
def home_page():

    chak=""
    duration=""
    history=""
    purpose=""
    credit_amount=""
    balance=""
    employment=""
    install_rate=""
    marital_status=""
    co_applicant=""
    present_residence=""
```

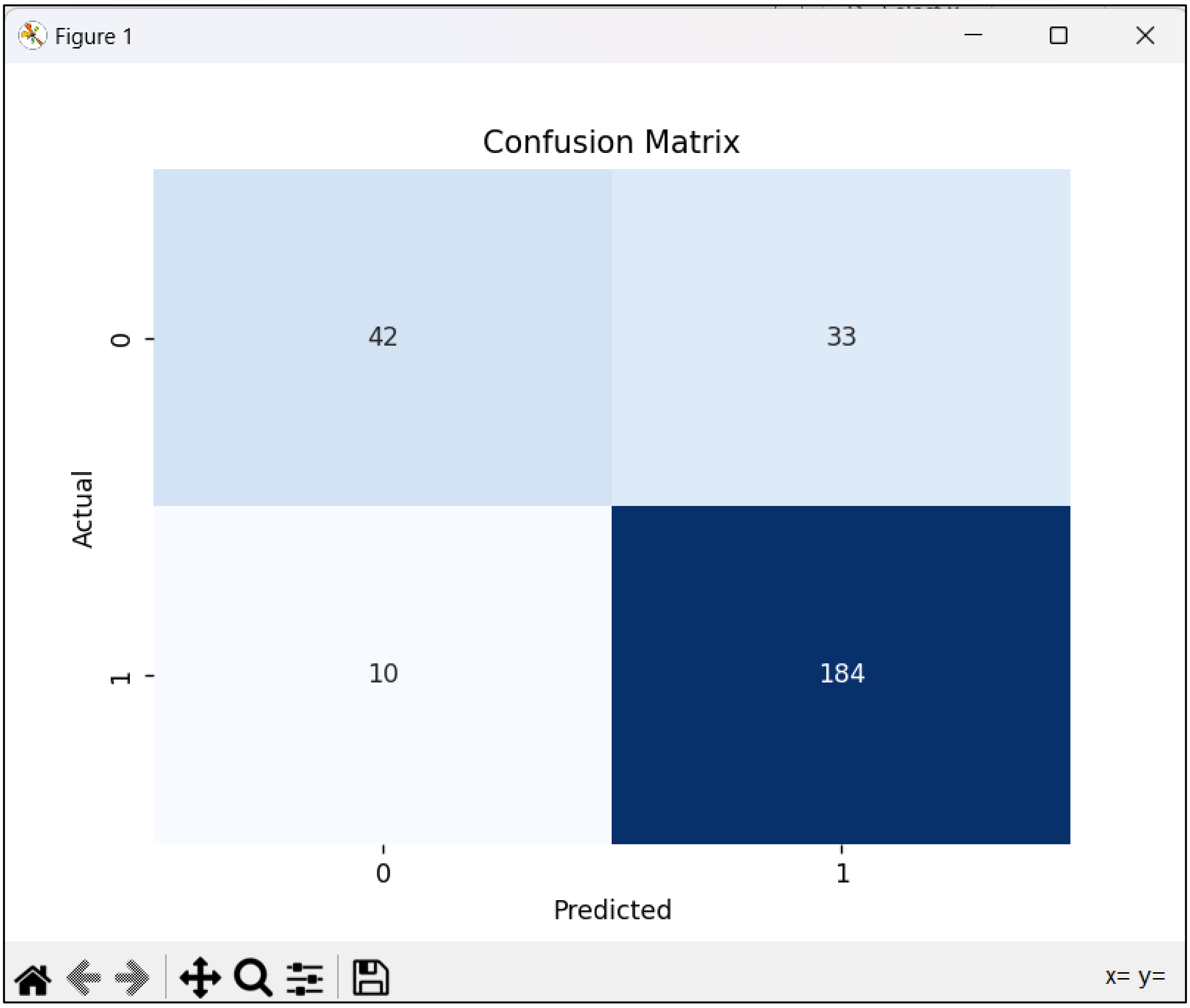
```
    real_estate=""
    age=""
    other_installment=""
    residence=""
    no_of_credits=""
    job=""
    no_of_dependents=""
    phone=""
    foreign=""

    if request.method == 'POST':

        chak = request.form['CHK_ACCT']
        duration = request.form['Duration']
        history = request.form['History']
        purpose = request.form['Purpose_of_credit']
        credit_amount = request.form['Credit_Amount']
        balance = request.form['Balance_in_Savings_A_C']
        employment = request.form['Employment']
        install_rate = request.form['Install_rate']
        marital_status = request.form['Marital_status']
        co_applicant = request.form['Co_applicant']
        present_residence = request.form['Present_Resident']
        real_estate = request.form['Real_Estate']
        age = request.form['Age']
        other_installment = request.form['Other_installment']
        residence = request.form['Residence']
        no_of_credits = request.form['Num_Credits']
        job = request.form['Job']
        no_of_dependents = request.form['No_dependents']
        phone = request.form['Phone']
        foreign = request.form['Foreign']

        data = {
            'CHK_ACCT': chak,
            'Duration': duration,
            'History': history,
            'Purpose_of_credit': purpose,
            'Credit_Amount': credit_amount,
            'Balance_in_Savings_A_C': balance,
            'Employment': employment,
            'Install_rate': install_rate,
            'Marital_status': marital_status,
            'Co_applicant': co_applicant,
            'Present_Resident': present_residence,
            'Real_Estate': real_estate,
            'Age': age,
            'Other_installment': other_installment,
```

```
            'Residence': residence,
            'Num_Credits': no_of_credits,
            'Job': job,
            'No_dependents': no_of_dependents,
            'Phone': phone,
            'Foreign': foreign
        }

        prediction = predict_credit_classification(model,
[list(data.values())])


        return render_template('index.html',
prediction=prediction,chak=chak,duration=duration,history=history,cre
dit_amount=credit_amount,balance=balance,employment=employment,instal
l_rate=install_rate,marital_status=marital_status,co_applicant=co_app
licant,present_residence=present_residence,real_estate=real_estate,ot
her_installment=other_installment,no_of_credits=no_of_credits,job=job
,no_of_dependents=no_of_dependents,phone=phone,purpose=purpose)


    return render_template('index.html')

if __name__ == '__main__':
    app.run(debug=False)
```

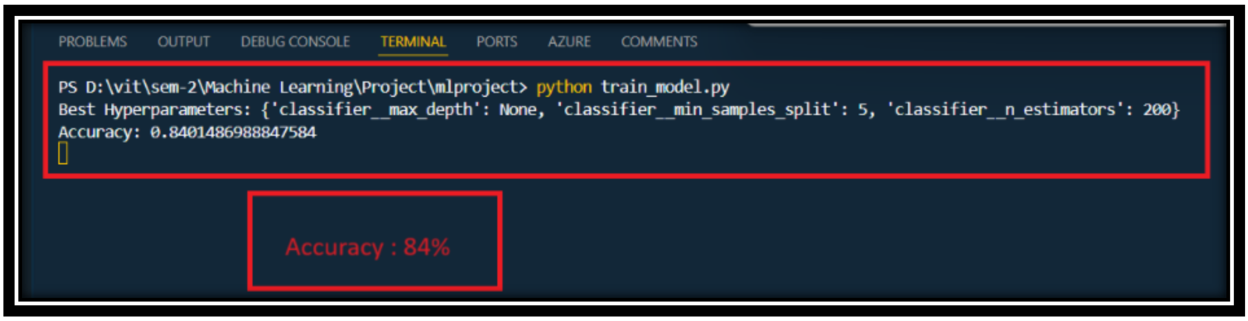
## Model.py

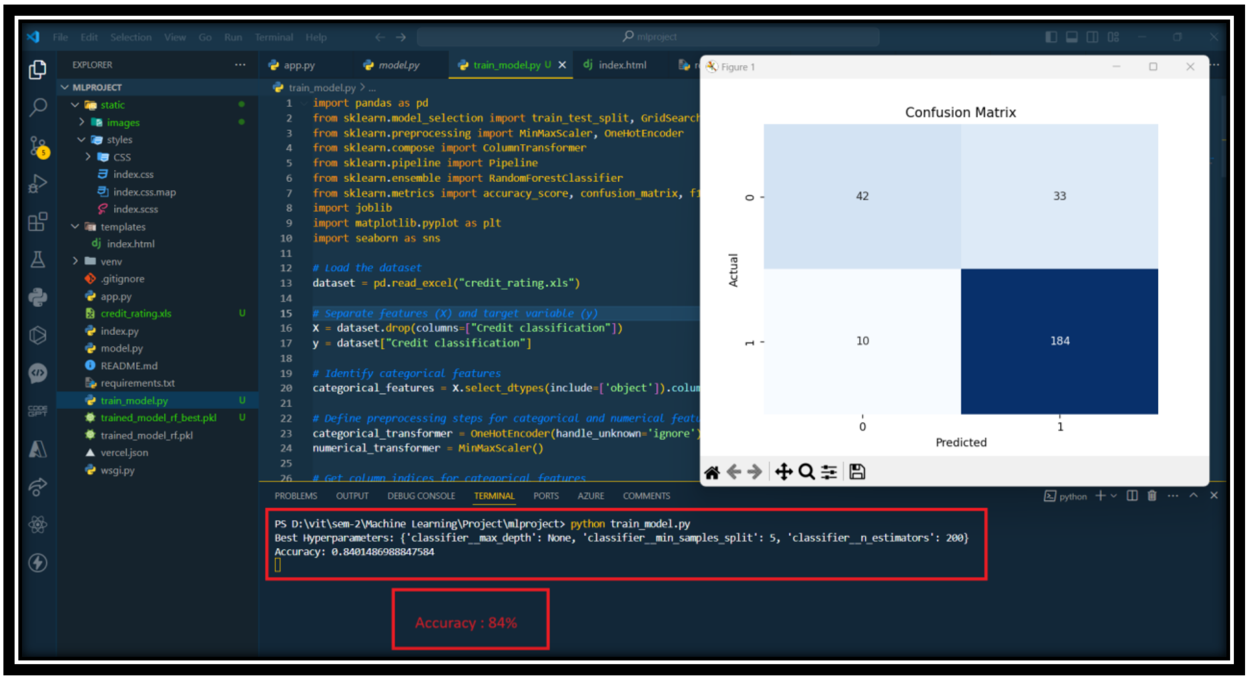
```
#model.py
import joblib
# Load the trained model
def load_model(model_path="trained_model_rf.pkl"): # Update the model
path to the one saved with Random Forest
    return joblib.load(model_path)

# Function to predict credit classification
def predict_credit_classification(model, input_data):
    try:
        prediction = model.predict(input_data)
        return prediction
    except Exception as e:
        return str(e)
```

## CONFUSION MATRIX & ACCURACY



# ACCURACY




# F1 SCORE

# CONCLUSION

In conclusion, our project successfully developed a credit rating model to classify customers' creditworthiness based on their personal and bank account details. Leveraging a dataset provided in the credit_rating.xlsx, we implemented a Random Forest Classifier after thorough preprocessing and hyperparameter tuning. The model's performance was evaluated using standard metrics including accuracy, F1 score, and confusion matrix visualization. Deployed as a web application, users can conveniently input their information via a user-friendly form to receive credit rating predictions. While our model achieved satisfactory results, opportunities for future enhancement include exploring alternative algorithms and incorporating additional features. Overall, this project demonstrates the potential of machine learning techniques to streamline credit assessment processes and support informed decision-making in the financial sector.

- Objective Achievement: Successfully developed a credit rating model to classify customers' creditworthiness based on personal and bank account details.

- Methodology: Utilized supervised learning with the target variable "Credit classification" from the provided credit_rating.xlsx dataset.

- Model Selection: Implemented a Random Forest Classifier after thorough preprocessing, including handling missing values, encoding categorical variables, and scaling numerical features.

- Hyperparameter Tuning: Optimized model performance using Grid Search with 5-fold cross-validation to fine-tune hyperparameters.

- Performance Evaluation: Evaluated model performance using standard metrics: accuracy, F1 score, and confusion matrix visualization.

- Deployment: Deployed the model as a web application, allowing users to input their information via a user-friendly form to receive credit rating predictions.

- Future Enhancements: Identified opportunities for future enhancement, such as exploring alternative algorithms and incorporating additional features.

- Impact: Demonstrated the potential of machine learning techniques to streamline credit assessment processes and support informed decision-making in the financial sector.