Overview:

The Loan Approval Prediction Project allows you to experiment with and compare the performance of various machine-learning models on a preprocessed loan dataset. The workflow includes:

1. Preprocessing the original dataset (`dataset.csv`) into a standardized format.
2. Running individual models, each of which is implemented in a separate Python file.

Setup:

To use the project, you can choose one of the following options:

Option 1: Clone the Repository (Preferable)

The code repository was made public after the project was submitted. Clone the repository from GitHub for the latest version of the code.

```
git clone https://github.com/kahf10/Loan-Approval-Prediction.git
cd Loan-Approval-Prediction
```

Option 2: Download the Code

Download the `zip` file uploaded to BBLearn. Extract it to your local machine and open the `Code` Folder.

Usage Instructions:

Step 1: Preprocessing

Requirements: Make sure `dataset.csv` is in the same directory as `Preprocessor.py` before running this step.

Run the `Preprocessor.py` file, which processes the original `dataset.csv` and generates a new processed file named `PreprocessedDataset.csv`:

To do the above, just run the file or type the following command in your terminal (depending on your version of Python):

```
python Preprocessor.py or python3 Preprocessor.py
```

Step 2: Running the Models

Requirements: Make sure `PreprocessedDataset.csv` is in the same directory as the model you are trying to run.

After preprocessing, you can run any of the following files in **any order**. Each file is standalone and can be executed directly.

a. **Logistic Regression:** Implements Logistic Regression with custom gradient descent for predictions.

```
python LogisticRegressionModel.py
```

b. **Linear SVM:** Implements Support Vector Machine with a linear kernel.

```
python LinearSVMModel.py
```

c. **Polynomial SVM**: You can change the degree of the kernel function by modifying the value on **line 77** of the file.

```
python PolynomialSVMModel.py
```

d. **Ensemble:** This file combines results from Logistic Regression and Linear SVM. Ensure both `LogisticRegressionModel.py` and `LinearSVMModel.py` are in the same directory before running.

```
python EnsembleModel.py
```

e. **Artificial Neural Network (ANN):** Implements an Artificial Neural Network with Dropout and Early Stopping.

```
python ANNModel.py
```

Requirements:

The project requires the following Python libraries:

- NumPy
- pandas
- matplotlib
- seaborn
- torch
- sklearn.metrics