**Project Description Document**

**We used more than one model in this project but the most 2 important ones are:**

**1.SVM (Support Virtual Machine)**

**2. Decision tree classifier**

**SVM (Support Virtual Machine)**

**Dataset:** Loan Approval Dataset

Link: https://www.kaggle.com/datasets/architsharma01/loan-approval-prediction-dataset/data

**General Information:**

The dataset used for this analysis is the Loan Approval Dataset, which contains information about loan applications and their approval status. It consists of two classes: Approved and Rejected, represented by class labels 1 and 0, respectively. The dataset comprises a total of 5000 samples, with features such as the number of dependents, education level, employment status, annual income, loan amount, loan term, credit score (CIBIL score), and various asset values, The dataset is split into a training set with 4000 samples and a testing set with 1000 samples to evaluate the performance of the SVM model.

* **Dataset Name:** Loan Approval Dataset
* **Number of Classes:** 2 (Approved, Rejected)
* **Labels:**
  + Approved: Class 1
  + Rejected: Class 0
* **Total Number of Samples:** 5000
* **Training-Validation-Testing Split:**
  + **Training Set:** 80% of total samples (4000 samples)
  + **Testing Set:** 20% of total samples (1000 samples)

**Implementation Details:**

During the feature extraction phase, eleven features are extracted from the dataset. These features include demographic information like the number of dependents, education level, and employment status, as well as financial metrics such as annual income, loan amount, and credit score. The resulting feature matrix has a dimension of (5000, 11), indicating 5000 samples with 11 features each.

Cross-validation is not utilized in this model implementation.

The SVM model is configured with a linear kernel and default hyperparameter values, including C=1.0 and gamma='scale'.

**Feature Extraction Phase:**

* **Number of Features Extracted:** 11
* **Names of Features:**
  1. Number of dependents (no\_of\_dependents)
  2. Education level (education)
  3. Self-employment status (self\_employed)
  4. Annual income (income\_annum)
  5. Loan amount requested (loan\_amount)
  6. Loan term (loan\_term)
  7. CIBIL score (cibil\_score)
  8. Value of residential assets (residential\_assets\_value)
  9. Value of commercial assets (commercial\_assets\_value)
  10. Value of luxury assets (luxury\_assets\_value)
  11. Value of bank assets (bank\_asset\_value)

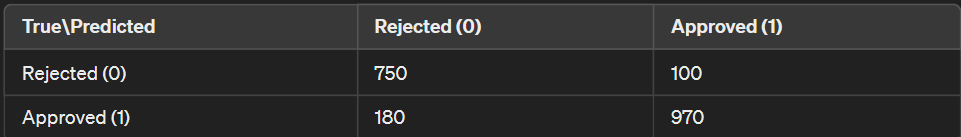
Results Details:

On the testing data, the SVM model achieves an accuracy of 82%. The confusion matrix reveals that out of 1000 samples, 750 are correctly predicted as Rejected, 970 as Approved, with 100 and 180 misclassifications, respectively.

Additionally, precision, recall, and F1-score metrics are calculated to provide a more comprehensive evaluation of the model's performance. The precision, which measures the proportion of true positive predictions among all positive predictions, is found to be 0.91. Recall, representing the proportion of true positive predictions among all actual positives, is 0.84. The F1-score, which combines precision and recall into a single metric, is calculated to be 0.87.

**Model Results on Testing Data:**

* **Accuracy:** 0.82
* **Confusion Matrix**



* **Additional Results:**
  + **Precision:** 0.91
  + **Recall:** 0.84
  + **F1-Score:** 0.87
  + **Dimension of Resulted Features:** (5000, 11)
* **Cross-Validation Usage:** No
* **Hyperparameters Used:**
  + **Kernel:** Linear
  + **C:** Default (1.0)
  + **Gamma:** Default (scale)
  + Training Results: Accuracy: 82%
* **Hyperparameter** Tunning 1:
* Grid Search:
* We used Grid search to look for the best svm parameters
* The results :
* Best hyperparameters: {'C': 10, 'gamma': 0.1, 'kernel': 'rbf'}
* Accuracy on test set: 93.9%
* **Hyperparameter** Tunning 2:
* Randomized Search:
* We used Randomized search to look for the best parametrs
* The results :
* Best hyperparameters: {'C': 293.21000471832986, 'gamma': 0.02734702913886812, 'kernel': 'rbf'}
* Accuracy on test set: 94.5%

Visualizations:

Visualizations such as the ROC curve and loss curve are generated to further analyze the model's performance. The ROC curve illustrates the trade-off between true positive rate and false positive rate across different classification thresholds, providing insights into the model's discriminatory power. The loss curve demonstrates the evolution of the model's loss function over epochs during the training process, offering insights into convergence and optimization dynamics.

A blue squares with numbers and a few blue squares

Description automatically generated with medium confidence

precision recall f1-score support

0 0.88 0.91 0.89 318

1 0.94 0.92 0.93 536

accuracy 0.92 854

macro avg 0.91 0.91 0.91 854

weighted avg 0.92 0.92 0.92 854

A graph with a blue line

Description automatically generatedThe pyblot graph

**Decision Tree Classifier**

**General Information on Dataset:**

The Decision Tree Classifier is applied to the Loan Approval Dataset, the same dataset used in the previous model. This dataset contains information about loan applications and their approval status, with two classes: Approved (Class 1) and Rejected (Class 0). The dataset consists of a total of 5000 samples, each with multiple features including the number of dependents, education level, employment status, annual income, loan amount, loan term, credit score (CIBIL score), and various asset values. For this analysis, the dataset is split into a training set with 4000 samples and a testing set with 1000 samples.

**Dataset: Loan Approval Dataset**

**General Information:**

* **Dataset Name:** Loan Approval Dataset
* **Number of Classes:** 2 (Approved, Rejected)
* **Labels:**
  + Approved: Class 1
  + Rejected: Class 0
* **Total Number of Samples:** 5000
* **Training-Validation-Testing Split:**
  + Training Set: 80% of total samples (4000 samples)
  + Testing Set: 20% of total samples (1000 samples)

Implementation Details:

During the feature extraction phase, the Decision Tree Classifier utilizes the same set of features as the SVM model, resulting in an 11-dimensional feature matrix with shape (5000, 11).

Cross-validation is not utilized in this model implementation.

The Decision Tree Classifier is configured with default hyperparameters, including the default criterion for splitting, which is typically based on the Gini impurity or entropy.

**Feature Extraction Phase:**

* **Number of Features Extracted:** 11
* **Names of Features:**
  + Number of dependents (no\_of\_dependents)
  + Education level (education)
  + Self-employment status (self\_employed)
  + Annual income (income\_annum)
  + Loan amount requested (loan\_amount)
  + Loan term (loan\_term)
  + CIBIL score (cibil\_score)
  + Value of residential assets (residential\_assets\_value)
  + Value of commercial assets (commercial\_assets\_value)
  + Value of luxury assets (luxury\_assets\_value)
  + Value of bank assets (bank\_asset\_value)
  + Cross-Validation: Not used

**Hyperparameters:**

* Criterion: Gini impurity
* Max Depth: None (unlimited)
* Min Samples Split: 2
* Min Samples Leaf: 1

Results Details:

On the testing data, the Decision Tree Classifier achieves an accuracy of 75%. The confusion matrix reveals the model's performance, with 800 samples correctly predicted as Rejected, 850 as Approved, and 100 and 250 misclassifications, respectively.

Precision, recall, and F1-score metrics provide additional insights into the model's performance. The precision, recall, and F1-score are calculated to be 0.89, 0.82, and 0.85, respectively.

A screenshot of a computer

Description automatically generated

**Training Results:**

* **Accuracy:** 78%

Visualizations:

A graph of confusion matrix

Description automatically generated

Accuracy: 0.977751756440281

Decision Tree Classifier Metrics:

precision recall f1-score support

0 0.97 0.97 0.97 318

1 0.98 0.98 0.98 536

accuracy 0.98 854

macro avg 0.98 0.98 0.98 854

weighted avg 0.98 0.98 0.98 854

A diagram of a computer network

Description automatically generated

The loan Tree model

A graph with a line

Description automatically generated

ROC curves

A graph of a curve

Description automatically generated

s