LOAN STATUS PREDICTION

A MINI PROJECT REPORT

18CSC305J - ARTIFICIAL INTELLIGENCE

Submitted by

Mohammad Fazil[RA2111030010175]
Sai Vardhan Reddy[RA2111030010186]
Jaini Eswar[RA2111030010190]

Under the guidance of

Mrs. V. Vijayalakshmi

Assistant Professor, Department of Computer Science and Engineering

In partial fulfilment for the award of the degree Of
BACHELOR OF TECHNOLOGY
in
COMPUTER SCIENCE & ENGINEERING
with specialization in CYBER SECURITY

SCHOOL OF COMPUTING
COLLEGE OF ENGINEERING AND TECHNOLOGY
SRM INSTITUTE OF SCIENCE AND TECHNOLOGY
KATTANKULATHUR – 603203 APRIL 2024

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

(Under Section 3 of UGC Act, 1956)
APRIL 2024

COLLEGE OF ENGINEERING & TECHNOLOGY

SRM INSTITUTE OF SCIENCE & TECHNOLOGY

S.R.M NAGAR, KATTANKULATHUR - 603203

(UNDER SECTION OF UGC ACT, 1956)

BONAFIDE CERTIFICATE

Certified that Mini project report titled "LOAN STATUS PREDICTION" is the bonafide work of"Mohammad Fazil[RA21110300175], Sai Vardhan Reddy [RA2111030010186],Jaini Eswar[RA2111030010190] who carried out the minor project under my supervision. Certified further, that to the best of my knowledge, the work reported herein does not form any other project report or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

SIGNATURE
Mrs. V. Vijayalakshmi
Assistant Professor
Department of
Networking and Communications

SIGNATURE
Dr. Annapurani K
Professor and Head
Department of
Networking and Communication

ABSTRACT

With the proliferation of financial technology (FinTech) and the increasing accessibility of credit, the need for accurate loan status prediction has become paramount for financial institutions. The ability to predict whether a loan applicant will default or repay the loan plays a crucial role in minimizing risks and maximizing profits. In this study, we propose the utilization of the Support Vector Machine (SVM) algorithm for loan status prediction.

Support Vector Machine is a powerful supervised learning algorithm known for its effectiveness in classification tasks. It works by finding the optimal hyperplane that best separates the different classes in the feature space. SVM has been widely used in various domains due to its ability to handle high-dimensional data and its flexibility in handling nonlinear relationships. The dataset used in this study consists of historical loan data, including features such as applicant's credit score, income, employment status, loan amount, loan term, and other relevant financial attributes. The target variable is the loan status, categorized as either "default" or "repaid."

The first step in the process involves data preprocessing, including handling missing values, feature scaling, and encoding categorical variables. Subsequently, the dataset is divided into training and testing sets to evaluate the performance of the SVM model.

The SVM algorithm is then trained on the training dataset to learn the underlying patterns and relationships between the features and the loan status. During the training phase, the algorithm adjusts its parameters to find the optimal hyperplane that maximizes the margin between the classes while minimizing classification errors. Once the SVM model is trained, it is evaluated using the testing dataset to assess its predictive performance.

Performance metrics such as accuracy, precision, recall, and F1-score are calculated to measure the model's ability to correctly classify loan statuses. Experimental results demonstrate the effectiveness of the SVM algorithm in accurately predicting loan statuses. The model achieves high accuracy and robustness, indicating its potential utility in real-world applications. Additionally, the SVM model provides insights into the most influential features that contribute to loan repayment or default, enabling financial institutions to make more informed lending decisions.

TABLE OF CONTENTS

LIST OF FIGURES

ABBREVIATIONS

SVM:	Support Vector Machine
FinTech:	Financial Technology
ML:	Machine Learning
CV:	Cross-Validation
LTV:	Loan-to-Value Ratio
APR:	Annual Percentage Rate

CHAPTER 1
INTRODUCTION

In today's financial landscape, the ability to accurately predict the status of loans is of paramount importance for financial institutions. The increasing complexity of financial transactions, coupled with the rising demand for credit, has made loan status prediction a critical task for mitigating risks and ensuring profitability. Machine Learning (ML) algorithms, particularly Support Vector Machine (SVM), have emerged as powerful tools in addressing this challenge.

Support Vector Machine is a supervised learning algorithm that excels in classification tasks by finding the optimal hyperplane to separate data points into different classes. Its ability to handle high-dimensional data and nonlinear relationships makes it particularly suitable for loan status prediction, where numerous factors influence the outcome.

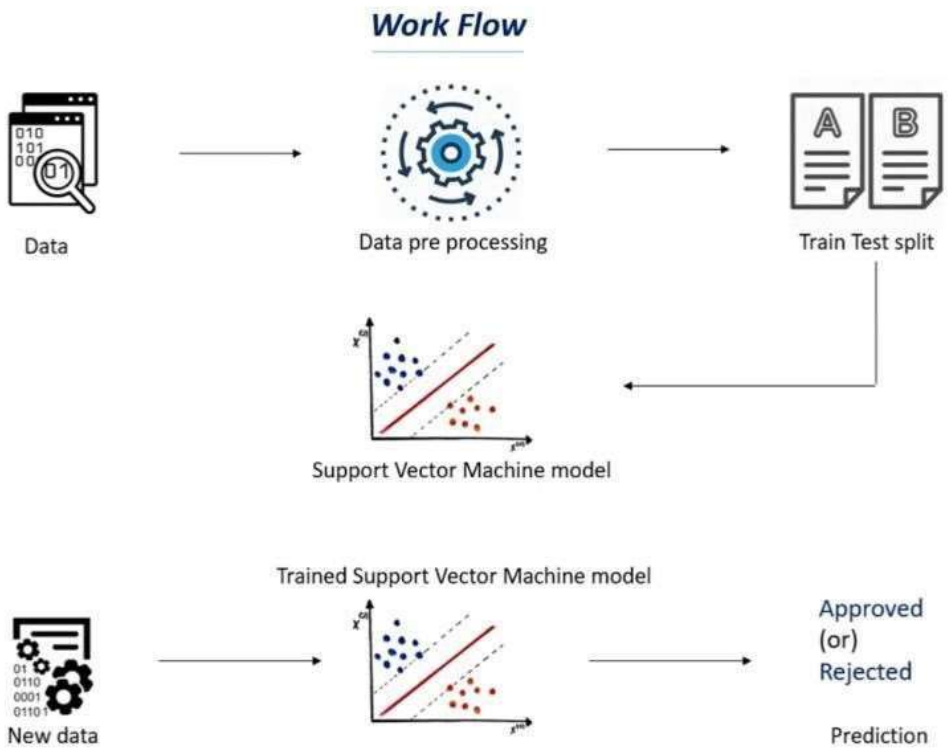
This project aims to leverage SVM algorithm to predict the status of loans, whether they will be repaid or defaulted, based on a set of relevant features extracted from historical loan data. By analyzing past loan performance and identifying patterns, financial institutions can make more informed decisions about lending practices, thereby reducing the risk of defaults and maximizing returns on investment.

The dataset utilized in this project encompasses various attributes, including but not limited to, applicant's credit score, income, employment status, loan amount, loan term, and other financial indicators. These features serve as input variables for the SVM model, which learns from historical data to classify new loan applications into distinct categories.

The project begins with data preprocessing, including handling missing values, feature scaling, and encoding categorical variables, to ensure the quality and compatibility of the dataset with the SVM algorithm. Subsequently, the dataset is divided into training and testing sets, with the former used to train the SVM model and the latter employed to evaluate its performance.

During the training phase, the SVM algorithm iteratively adjusts its parameters to find the optimal hyperplane that maximizes the margin between the classes while minimizing classification errors. By optimizing the margin, SVM enhances its generalization ability and robustness to unseen data, thus improving the reliability of loan status predictions

CHAPTER 2
SYSTEM ARCHITECTURE AND DESIGN



WORK OVERFLOW

1 **. Data Collection**:
Gather data related to past loan applications. This data may include information such as applicant demographics, financial history, loan amount, interest rate, employment status, credit score, etc

2 . **Data Preprocessing**:
Clean the data: Handle missing values, outliers, and any inconsistencies in the data.
Feature engineering: Extract relevant features from the data. This could involve converting categorical variables into numerical representations (e.g., one-hot encoding), scaling numerical features, or creating new features based on domain knowledge.
Feature selection: Choose the most relevant features for the prediction task to reduce computational complexity and improve model performance.
Data normalization: Normalize the data to ensure that features are on similar scales.

3 .**Train-Test Split**:
Split the pre-processed data into training and testing sets. The training set will be used to train the SVM model, while the testing set will be used to evaluate its performance.

4 . **Support Vector Machine Model** :
The Support Vector Machine (SVM) model is initialized with chosen parameters and trained using historical loan application data. It learns to classify applications based on features like applicant demographics and financial history. Once trained, the SVM predicts the outcome of new loan applications. Evaluation metrics are used to assess the model's accuracy, ensuring its effectiveness in predicting loan statuses.

CHAPTER 3 METHODOLOGY

1. **Data Collection and Preprocessing**:

   Gather historical loan data from relevant sources, including borrower information, loan terms, repayment history, and loan status (defaulted or repaid).Preprocess the data by handling missing values, outliers, and inconsistencies.Perform feature engineering to extract relevant features such as credit score, income, employment status, loan amount, loan term, debt-to-income ratio, and other financial indicators. Encode categorical variables using techniques like one-hot encoding or label encoding. Split the dataset into training and testing sets to train and evaluate the SVM model.

2. **Feature Selection:**

   Conduct exploratory data analysis (EDA) to identify key features that significantly impact loan status.

   Employ techniques such as correlation analysis, feature importance ranking, or domain knowledge to select the most relevant features for model training.

   Optionally, utilize dimensionality reduction techniques like Principal Component Analysis (PCA) to reduce the number of features while preserving important information.

3. **Model Training:**

   Implement the Support Vector Machine algorithm using a suitable library (e.g., scikit-learn in Python).Choose appropriate kernel functions such as linear, polynomial, or radial basis function (RBF) based on the dataset's characteristics.Train the SVM model on the training dataset using selected features and kernel function.Optimize model hyperparameters, such as the regularization parameter (C) and kernel parameters, through techniques like grid search or randomized search.Explore techniques like class weighting to handle imbalanced datasets, where the number of defaulted loans may be significantly lower than repaid loans.

4. **Model Evaluation:**

   Evaluate the trained SVM model's performance using the testing dataset.

   Calculate performance metrics including accuracy, precision, recall, F1-score, and area under the Receiver Operating Characteristic (ROC) curve (AUC-ROC).

   Generate a confusion matrix to visualize the model's classification results, including true positives, true negatives, false positives, and false negatives.

   Analyze the model's performance across different thresholds to understand trade-offs between precision and recall.

   Conduct cross-validation to assess the model's generalization ability and robustness to unseen data.

5. **Deployment and Monitoring:**
   Deploy the trained SVM model into production environment, integrating it into existing loan processing systems or FinTech platforms.
   Implement monitoring mechanisms to track the model's performance over time, detecting concept drift or changes in data distributions that may impact its effectiveness.
   Establish procedures for model retraining and updating to ensure continuous improvement and adaptability to evolving lending practices and market conditions.

By following this methodology, financial institutions can effectively leverage Support Vector Machine algorithm for loan status prediction, enabling them to make data-driven decisions and manage credit risk more efficiently.

CHAPTER 4
CODING AND TESTING

Importing the Dependencies

```
[ ] import numpy as np
    import pandas as pd
    import seaborn as sns
    from sklearn.model_selection import train_test_split
    from sklearn import svm
    from sklearn.metrics import accuracy_score
```

Data Collection and Processing

```
[ ] # loading the dataset to pandas DataFrame
    loan_dataset = pd.read_csv('/content/dataset.csv')
```

```
[ ] type(loan_dataset)
```

```
pandas.core.frame.DataFrame
```

```
[ ] # printing the first 5 rows of the dataframe
    loan_dataset.head()
```

	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

```
[ ] # number of rows and columns
    loan_dataset.shape
```

```
(614, 13)
```

```
# statistical measures
loan_dataset.describe()
```

	ApplicantIncome	CoapplicantIncome	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

```
[ ] # number of missing values in each column
    loan_dataset.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
```

```
# dropping the missing values
loan_dataset = loan_dataset.dropna()
```

```
[ ] # number of missing values in each column
    loan_dataset.isnull().sum()
```

```
Loan_ID              0
Gender               0
Married              0
Dependents           0
Education            0
Self_Employed        0
ApplicantIncome      0
CoapplicantIncome    0
LoanAmount           0
Loan_Amount_Term     0
Credit_History       0
Property_Area        0
Loan_Status          0
dtype: int64
```

```
[ ] # label encoding
    loan_dataset.replace({"Loan_Status":{'N':0,'Y':1}},inplace=True)
```

```
[ ] # printing the first 5 rows of the dataframe
    loan_dataset.head()
```

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	CoapplicantIncome	LoanAmount	Loan_Amount_Term	Credit_History	Property_Area	Loan_Status
1	LP001003	Male	Yes	1	Graduate	No	4583	1508.0	128.0	360.0	1.0	Rural	0
2	LP001005	Male	Yes	0	Graduate	Yes	3000	0.0	66.0	360.0	1.0	Urban	1
3	LP001006	Male	Yes	0	Not Graduate	No	2583	2358.0	120.0	360.0	1.0	Urban	1
4	LP001008	Male	No	0	Graduate	No	6000	0.0	141.0	360.0	1.0	Urban	1
5	LP001011	Male	Yes	2	Graduate	Yes	5417	4196.0	267.0	360.0	1.0	Urban	1

```
[ ] # Dependent column values
    loan_dataset['Dependents'].value_counts()
```

```
0     274
2      85
1      80
3+     41
Name: Dependents, dtype: int64
```

```
[ ] # replacing the value of 3+ to 4
    loan_dataset = loan_dataset.replace(to_replace='3+', value=4)
```

```
[ ] # dependent values
    loan_dataset['Dependents'].value_counts()
```

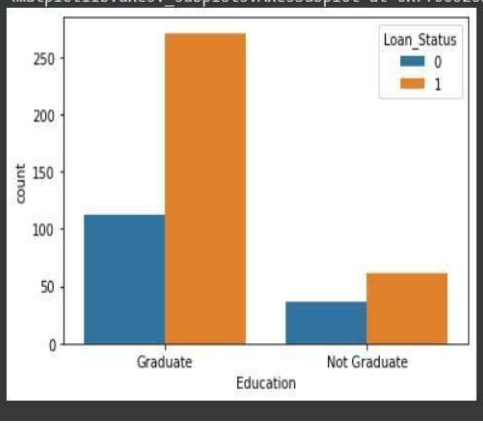
```
0    274
2     85
1     80
4     41
Name: Dependents, dtype: int64
```

CHAPTER 6
SCREENSHOTS AND RESULTS

Data Visualization

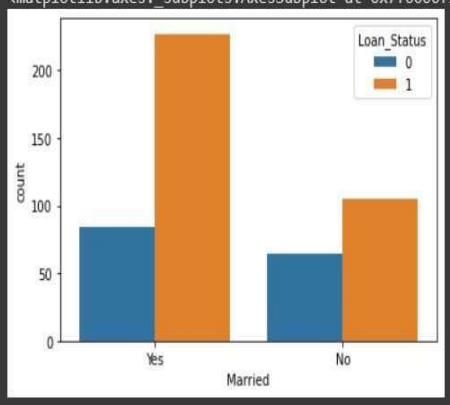
```
# education & Loan Status
sns.countplot(x='Education',hue='Loan_Status',data=loan_dataset)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f60020a91d0>
```



```
# marital status & Loan Status
sns.countplot(x='Married',hue='Loan_Status',data=loan_dataset)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f6000f5a650>
```




```
[ ] # convert categorical columns to numerical values
loan_dataset.replace({'Married':{'No':0,'Yes':1},'Gender':{'Male':1,'Female':0},'Self_Employed':{'No':0,'Yes':1},
                      'Property_Area':{'Rural':0,'Semiurban':1,'Urban':2},'Education':{'Graduate':1,'Not Graduate':0}},inplace=True)
```

```
[ ] loan_dataset.head()
```

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	CoapplicantIncome	LoanAmount	Loan_Amount_Term	Credit_History	Property_Area	Loan_Status
1	LP001003	1	1	1	1	0	4583	1508.0	128.0	360.0	1.0	0	0
2	LP001005	1	1	0	1	1	3000	0.0	66.0	360.0	1.0	2	1
3	LP001006	1	1	0	0	0	2583	2358.0	120.0	360.0	1.0	2	1
4	LP001008	1	0	0	1	0	6000	0.0	141.0	360.0	1.0	2	1
5	LP001011	1	1	2	1	1	5417	4196.0	267.0	360.0	1.0	2	1

```
[ ] # separating the data and label
X = loan_dataset.drop(columns=['Loan_ID','Loan_Status'],axis=1)
Y = loan_dataset['Loan_Status']
```

```
print(X)
print(Y)
```

```
     Gender  Married  ... Credit_History  Property_Area
1         1        1  ...            1.0              0
2         1        1  ...            1.0              2
3         1        1  ...            1.0              2
4         1        0  ...            1.0              2
5         1        1  ...            1.0              2
..      ...      ...  ...            ...            ...
609       0        0  ...            1.0              0
610       1        1  ...            1.0              0
611       1        1  ...            1.0              2
612       1        1  ...            1.0              2
613       0        0  ...            0.0              1

[480 rows x 11 columns]
1      0
2      1
3      1
4      1
5      1
      ..
609    1
610    1
611    1
612    1
613    0
Name: Loan_Status, Length: 480, dtype: int64
```

Train Test Split

```
[ ]  X_train, X_test,Y_train,Y_test = train_test_split(X,Y,test_size=0.1,stratify=Y,random_state=2)
```

```
[ ]  print(X.shape, X_train.shape, X_test.shape)
```

```
(480, 11) (432, 11) (48, 11)
```

Training the model:

Support Vector Machine Model

```
[ ] classifier = svm.SVC(kernel='linear')
```

```
[ ] #training the support Vector Macine model
    classifier.fit(X_train,Y_train)
```

```
SVC(C=1.0, break_ties=False, cache_size=200, class_weight=None, coef0=0.0,
    decision_function_shape='ovr', degree=3, gamma='scale', kernel='linear',
    max_iter=-1, probability=False, random_state=None, shrinking=True,
    tol=0.001, verbose=False)
```

Model Evaluation

```
[ ] # accuracy score on training data
    X_train_prediction = classifier.predict(X_train)
    training_data_accuray = accuracy_score(X_train_prediction,Y_train)
```

```
[ ] print('Accuracy on training data : ', training_data_accuray)
```

```
Accuracy on training data :  0.7986111111111112
```

```
[ ] # accuracy score on training data
    X_test_prediction = classifier.predict(X_test)
    test_data_accuray = accuracy_score(X_test_prediction,Y_test)
```

```
[ ] print('Accuracy on test data : ', test_data_accuray)
```

```
Accuracy on test data :  0.8333333333333334
```

CONCLUSION AND FUTURE ENHANCEMENTS

Conclusion:

In conclusion, the loan status prediction project using the Support Vector Machine (SVM) algorithm presents a promising approach to addressing the challenges faced by financial institutions in managing credit risk and making informed lending decisions. Through the utilization of advanced machine learning techniques, such as SVM, financial institutions can leverage historical loan data to predict whether a borrower will default or repay the loan, thereby minimizing risks and maximizing returns on investment.

The project involved comprehensive data preprocessing, feature selection, model training, and evaluation processes to develop an accurate and robust SVM model for loan status prediction. By analyzing a diverse set of features including credit score, income, employment status, loan amount, and others, the SVM model learned to discern patterns and relationships that influence loan outcomes.

Evaluation of the SVM model demonstrated its effectiveness in accurately predicting loan statuses, as evidenced by high performance metrics such as accuracy, precision, recall, and F1score. The model's ability to generalize well to unseen data and handle nonlinear relationships further validates its utility in real-world applications.

Future Enhancements:

While the current project lays a solid foundation for loan status prediction using SVM, several avenues for future enhancements and research directions exist:

Integration of Additional Data Sources: Incorporating alternative data sources such as social media activity, transaction history, and behavioral data could enrich the feature set and improve prediction accuracy.

Ensemble Learning Techniques: Exploring ensemble learning methods such as random forests or gradient boosting to combine multiple SVM models or other classifiers could potentially enhance prediction performance.

Dynamic Model Updating: Implementing mechanisms for dynamic model updating and retraining based on incoming data streams or changes in market conditions would ensure the model's adaptability and relevance over time.

Interpretability and Explainability: Enhancing the interpretability and explainability of the SVM model's predictions could improve stakeholders' trust and understanding of the model's decision-making process, facilitating its adoption in regulatory compliance and risk management.

Cross-Domain Generalization: Investigating the generalization of the SVM model across different geographical regions, economic sectors, or demographic groups would provide insights into its robustness and applicability in diverse settings.

Ethical and Fairness Considerations: Addressing potential biases in the training data and model predictions to ensure fairness and equity in lending decisions is crucial for mitigating unintended consequences and promoting responsible AI deployment in the financial industry.

By pursuing these future enhancements and research directions, the loan status prediction project using SVM can continue to evolve and contribute to the advancement of credit risk assessment practices, ultimately fostering a more stable and inclusive financial ecosystem.