Loan Eligibility Prediction

Check your loan eligibility using our predictive model.

LOAN PREDICTOR

Problem

- Loans are the core business of banks. The main profit comes directly from the loan’s interest. The loan companies grant a loan after an intensive process of verification and validation. However, they still don’t have assurance if the applicant is able to repay the loan with no difficulties.
- A survey conducted with 5000 respondents where 84% of the people said they were struggling to make the ends meet.
- Nearly half of all individuals in retail sector become a part of the DEBT TRAP

Married

Dependents

Education

Income

Loan Amount

Credit History

Features

Preprocessing

IS.NULL()

50 null values present.
As the value for credit history was missing, we removed the entire row with null values.

LABEL ENCODING

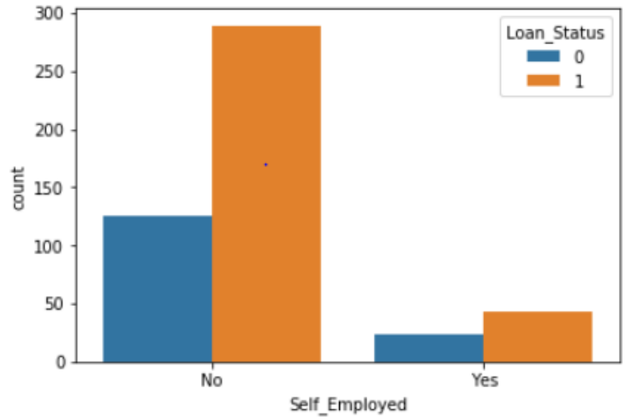
Categorical data i.e column GENDER and MARRIED were converted to numeric data ("YES " : 1), ("NO" : 0)

DEPENDENT VALUES 3+ TO 4

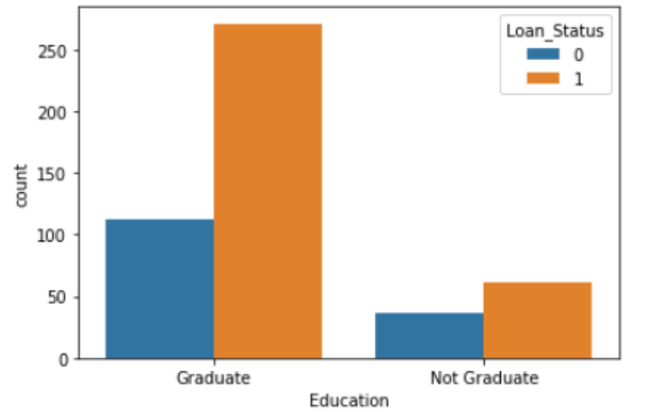
Data column ['DEPENDENT'] had values = 3+ which were replaced by 4.

Self Employed

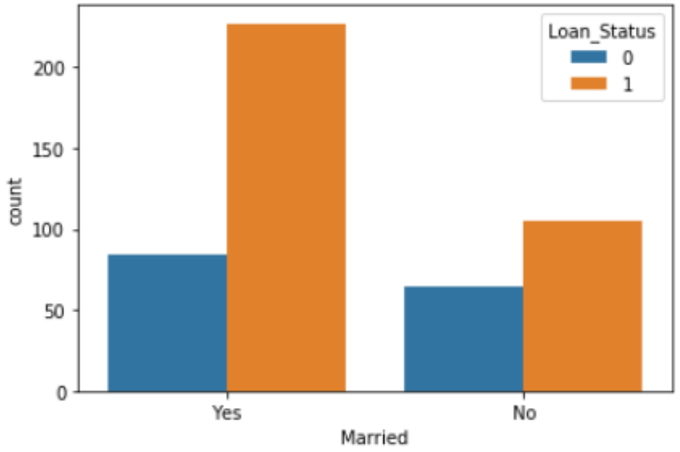
```
<matplotlib.axes._subplots.AxesSubplot at 0x1d50d21
```



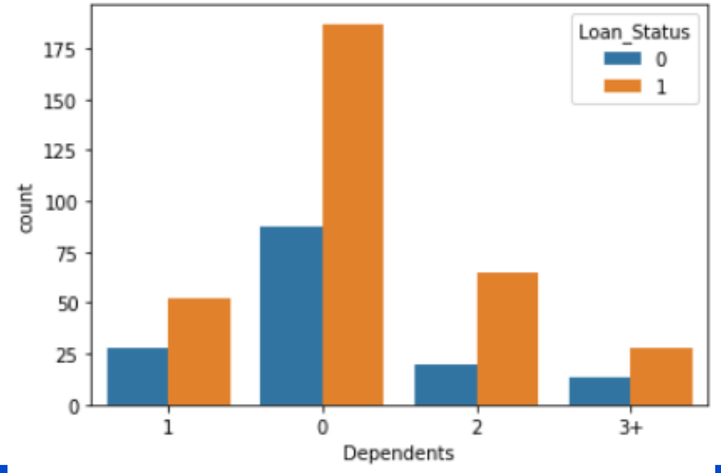
Education



Married



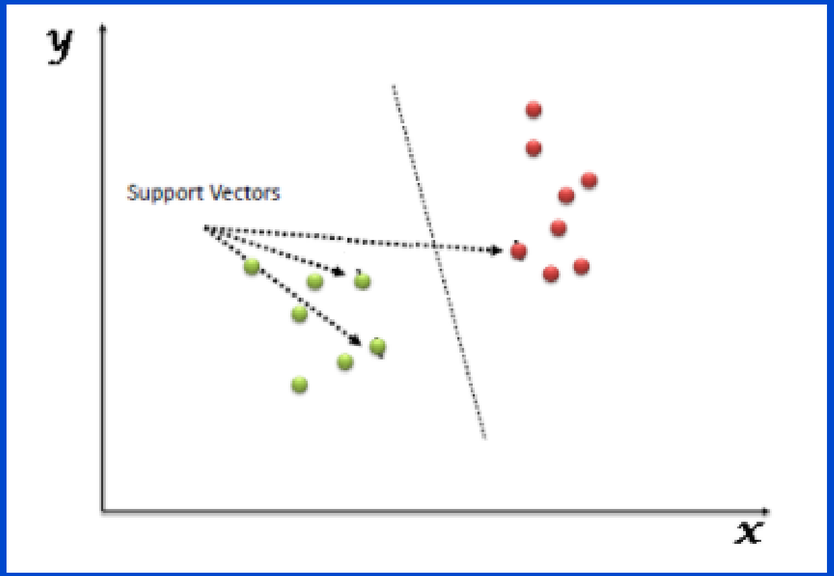
Dependents



SVM

- “Support Vector Machine” (SVM) is a supervised machine learning algorithm which can be used for both classification or regression challenges.

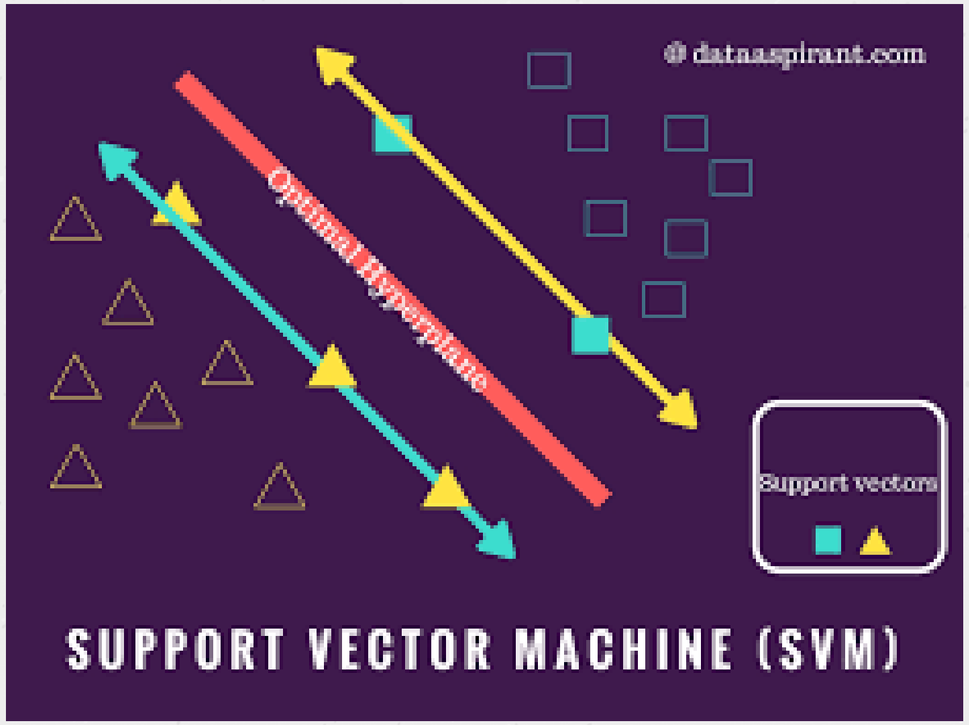
- However, it is mostly used in classification problems.



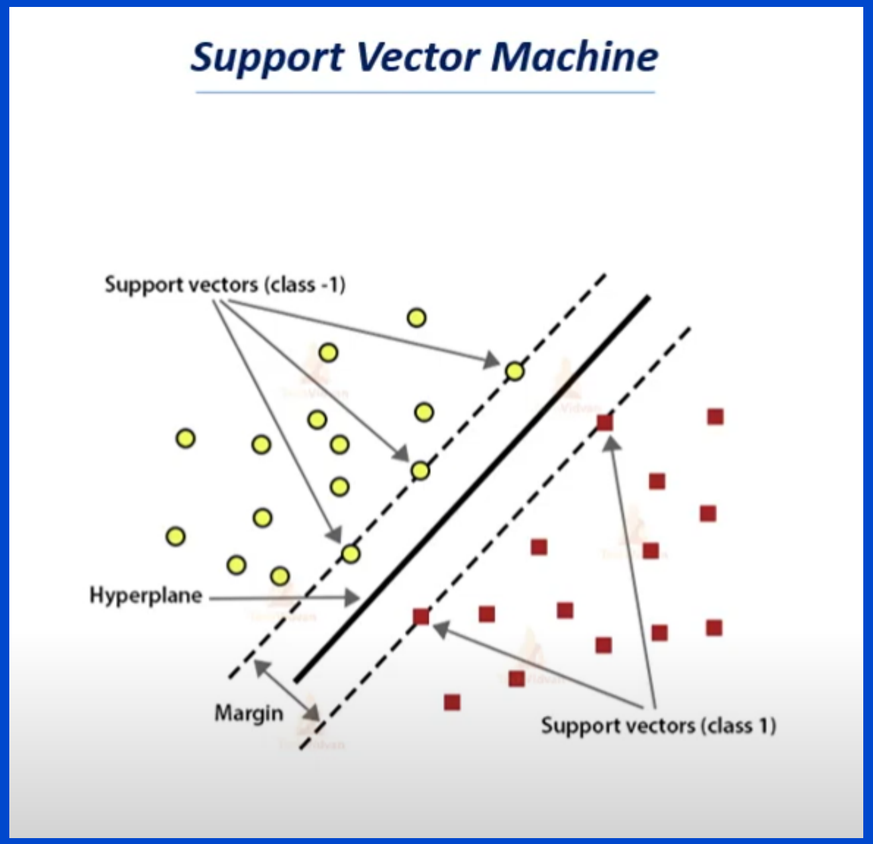
SVM's

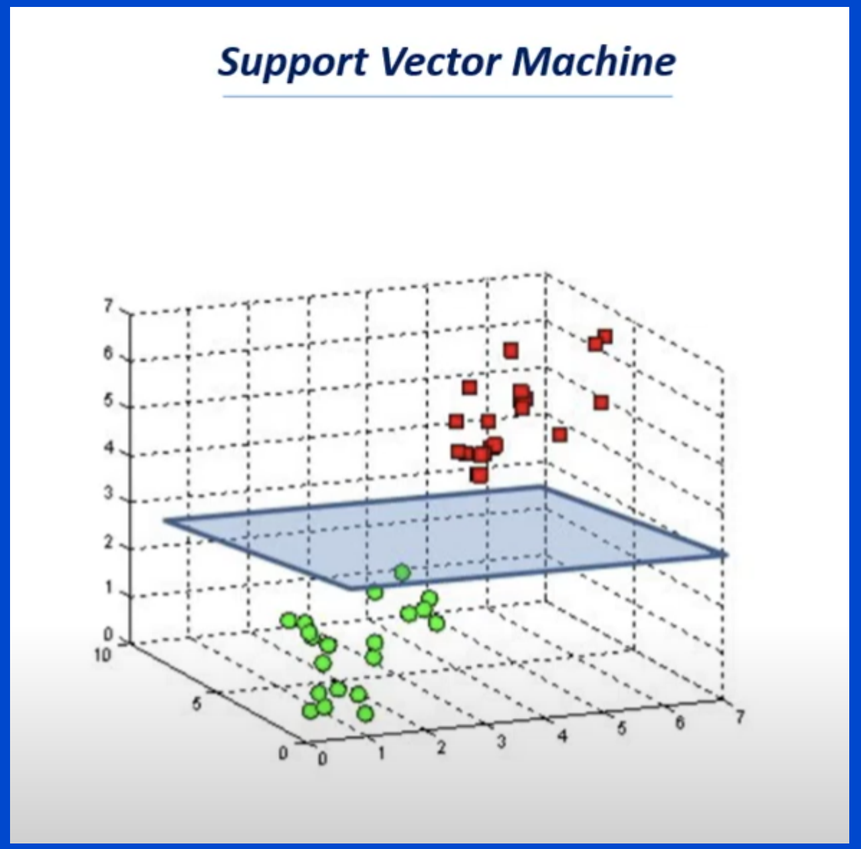
Scikit-learn

- Face Detection
- Intrusion Detection
- Classification of e-mails, news articles, and web pages
- Handwriting recognition



SVM Visualisation





Train Test Split

```
#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)
```

Train 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)
```

Accuracy on Training data

```
# 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 on Test

```
# 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
```