Title : SVM-Based Creditworthiness Classification Using Financial Risk Indicators

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.svm import SVC
from sklearn.metrics import classification_report
```

```
bank=pd.read_csv("/content/svm_bank_dataset.csv")
bank.columns = bank.columns.str.strip()
```

```
bank.head(5)
```

	IncomeToDebtRatio	CreditUtilization	\tCreditworthy
0	2.625225	2.070594	1
1	2.622298	3.043298	1
2	2.548860	2.540592	1
3	1.798008	1.837099	1
4	1.097395	4.249577	1

Next steps: Generate code with bank View recommended plots New interactive sheet

```
bank.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 300 entries, 0 to 299
Data columns (total 3 columns):
 #   Column             Non-Null Count  Dtype  
---  ------             --------------  -----  
 0   IncomeToDebtRatio  300 non-null    float64
 1   CreditUtilization  300 non-null    float64
 2   	Creditworthy      300 non-null    int64  
dtypes: float64(2), int64(1)
memory usage: 7.2 KB
```

```
bank.describe()
```

	IncomeToDebtRatio	CreditUtilization	\tCreditworthy
count	300.000000	300.000000	300.000000
mean	0.025356	0.109186	0.000000
std	2.701010	2.730316	1.001671
min	-5.203232	-5.421350	-1.000000
25%	-2.438856	-2.527057	-1.000000
50%	-0.191489	0.218337	0.000000
75%	2.532524	2.673652	1.000000
max	4.819330	5.101683	1.000000

```
bank.shape
```

```
(300, 3)
```

Dataset Description

Purpose of Dataset

This dataset is created to classify bank customers as creditworthy or not creditworthy, helping financial institutions automate loan approval and reduce credit risk using key financial indicators.

Structure of Dataset

The dataset consists of two input features (income_to_debt_ratio, credit_utilization) and one target label (creditworthy), suitable for binary classification using Support Vector Machines (SVM).

Column 1 – income_to_debt_ratio

Represents the ratio of a customer's income to their total debt obligations. Higher values indicate better repayment ability and lower financial stress, typically associated with creditworthy customers.

Column 2 – credit_utilization

Denotes the percentage of available credit that is currently being used. Lower values suggest more responsible credit usage, while high utilization can signal financial risk.

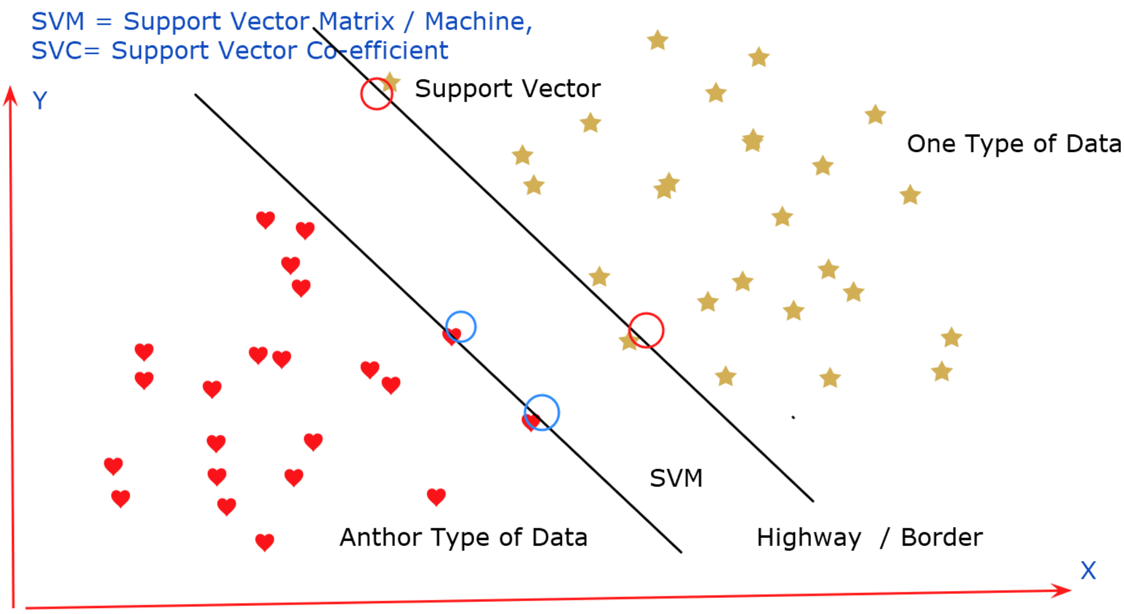
Column 3 – creditworthy (Target Label)

The binary output class used to train the SVM model:

1 → Creditworthy customer

-1 → Not creditworthy

This label helps the model learn to separate risky applicants from safe ones based on their financial behavior.




```
#Prepare features and labels
x=bank[['IncomeToDebtRatio','CreditUtilization']]
y=bank[['   Creditworthy']]
```

```
#Train test split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2,random_state=42)
```

Importance of SVM (Support Vector Machines)

Effective in High-Dimensional Spaces

SVM performs exceptionally well when the number of features exceeds the number of samples — ideal for complex domains like bioinformatics, text classification, or scanned luggage analysis.

Robust to Overfitting (Especially with Linear Models)

With appropriate kernel choice and regularization (like C parameter tuning), SVM offers strong generalization capabilities, even in noisy or overlapping datasets.

Maximizes Margin for Better Separation

SVM constructs the optimal hyperplane that maximizes the margin between classes — leading to more reliable and stable classification, especially in high-stakes fields like fraud detection or security screening.

Support Vectors Drive the Decision Boundary

Only a few critical data points (support vectors) are used to build the model — this makes SVM both memory-efficient and interpretability-friendly.

Powerful via the Kernel Trick

Through kernels (e.g., RBF, polynomial, sigmoid), SVM transforms data into higher dimensions to solve non-linear problems without explicitly computing the transformation — enabling SVM to adapt across industries from finance to healthcare.

```
#Train SVM
svm_model=SVC(kernel='linear')
svm_model.fit(x_train,y_train.values.ravel())
```

```
SVC
SVC(kernel='linear')
```

```
#Predict and evaluate
y_pred=svm_model.predict(x_test)
print(classification_report(y_test,y_pred))
```

```
              precision    recall  f1-score   support

          -1       1.00      1.00      1.00        31
           1       1.00      1.00      1.00        29

    accuracy                           1.00        60
   macro avg       1.00      1.00      1.00        60
weighted avg       1.00      1.00      1.00        60
```

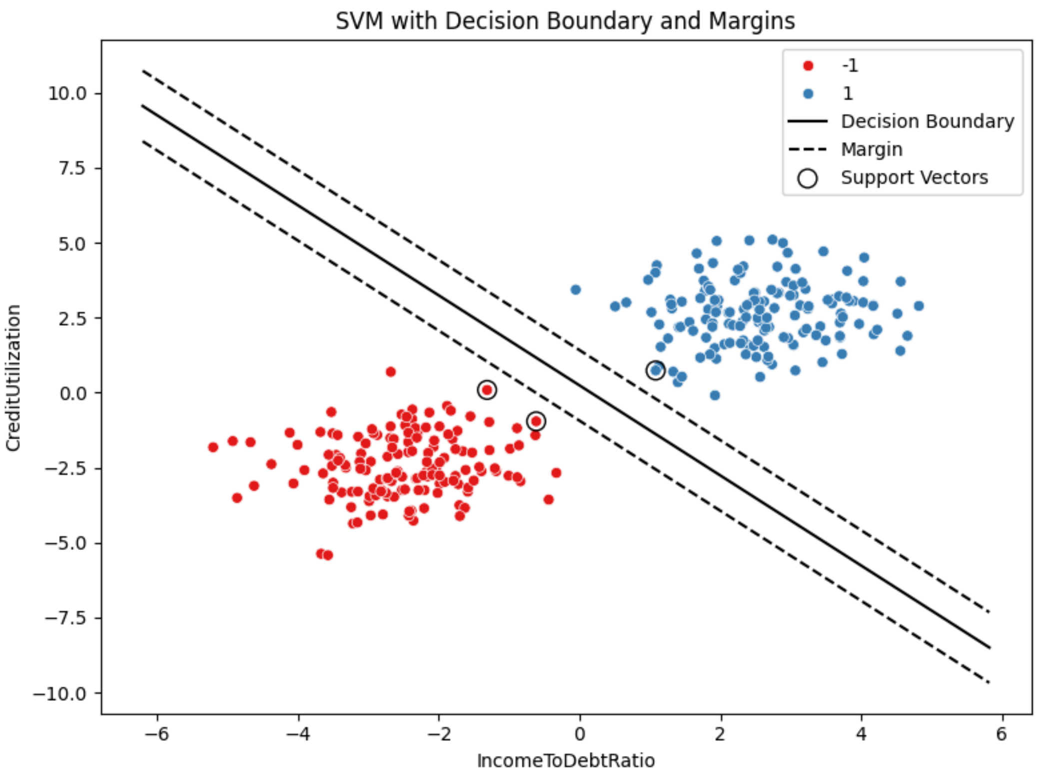
```
# Get the separating hyperplane
w = svm_model.coef_[0]
b = svm_model.intercept_[0]

# Create line space for decision boundary and margins
xx = np.linspace(x['IncomeToDebtRatio'].min() - 1, x['IncomeToDebtRatio'].max() + 1)
yy = -(w[0] * xx + b) / w[1]              # decision boundary
margin = 1 / np.linalg.norm(w)
yy_down = yy - margin                     # lower margin
yy_up = yy + margin                       # upper margin

# Plotting
plt.figure(figsize=(8, 6))
sns.scatterplot(data=bank, x='IncomeToDebtRatio', y='CreditUtilization', hue='Creditworthy', palette='Set1')
plt.plot(xx, yy, 'k-', label="Decision Boundary")
plt.plot(xx, yy_down, 'k--', label="Margin")
plt.plot(xx, yy_up, 'k--')

# Highlight support vectors
plt.scatter(svm_model.support_vectors_[:, 0], svm_model.support_vectors_[:, 1],
            s=100, facecolors='none', edgecolors='k', label="Support Vectors")

plt.title("SVM with Decision Boundary and Margins")
plt.xlabel("IncomeToDebtRatio")
plt.ylabel("CreditUtilization")
plt.legend()
plt.tight_layout()
plt.show()
```



Conclusion

The Support Vector Machine (SVM) classifier successfully separates creditworthy and non-creditworthy customers using two key financial indicators: IncomeToDebtRatio and CreditUtilization.

The decision boundary (solid line) clearly divides the two classes, maximizing the margin between them.

Support vectors (highlighted points) play a crucial role in determining the optimal separating hyperplane.

This indicates that SVM is highly effective in classifying financial risk using limited, interpretable features.

This outcome demonstrates the power of SVM in binary classification tasks, especially when decision boundaries are clear and the margin maximization principle is beneficial for generalization.

Disclaimer

This dataset is synthetically generated for academic purposes to demonstrate the functioning of Support Vector Machines (SVM) in a credit classification context. It does not represent actual customer data and should not be used for real-world financial decisions or policy modeling