Tutorial: Exploring Support Vector Machines (SVM) in ML

Course: Machine Learning and Neural Networks
Student Name: Jafar Urchintala
Student ID: 23036717
GitHub URL: https://github.com/Jafarurchintala/Jafarurchintala-Machine-Learning-Tutorial.git

Abstract

SVMs, a controlled machine learning technique used for classification and regression errands, are examined in this lesson along with their theoretical underpinnings and real-world applications. We demonstrate the steps of data preprocessing, SVM model training, and evaluation using a real-world example of identifying fraudulent credit card transactions. The tutorial demonstrates how SVM can efficiently handle imbalanced and high-dimensional data, obtaining strong presentation events like high accuracy, recollection, and ROC AUC scores.

Introduction to Machine Learning

ML allows systems to study from statistics and derive to inferences or estimates deprived of the need for clear program design. As ML algorithms exposed more data, they get better over time and become increasingly useful for resolving challenging issues.

Three primary forms of machine learning can be distinguished:

- Supervised Learning: To generate predictions or classifications (such as identifying fraudulent transactions), the model studies from categorized data.
- Unsupervised Learning: By grouping clients according to their purchasing behaviors, the model determines designs in un-labeled data.
- Reinforcement Learning: This classical gains knowledge through trial & error connections by the situation (e.g., autonomous driving or game-playing AI).

ML is essential in many industries, including healthcare, finance, retail, and transportation. For example, in fraud detection, ML helps monitor and flag suspicious transactions in real-time.

Introduction to Support Vector Machines (SVM)

Strong computer learning methods, Support Vector Machines (SVMs) are especially helpful for binary classification issues. Discovery the optimal hyperplane to split data points interested in two groups with the greatest margin is how SVM operates. Among the fundamental ideas of SVM are:

- **Support Vectors**: These are the information bordering to the hyperplane that mark its position.
- **Margin**: The separation between any class's closest data points and the hyperplane.
- **Kernels**: Functions that transform data into higher dimensions, making it easier to separate classes.

SVM is ideal for:

- **High-dimensional datasets** where there are more features than samples.
- Issues in which the boundaries between classes are clearly defined.

- Tasks with **imbalanced datasets**, like fraud detection, where one class (e.g., legitimate transactions) is much larger than the other (e.g., fraudulent transactions).

This tutorial focuses on using SVM for detecting **credit card fraud**, a problem with imbalanced data and complex relationships between features.

Why Use SVMs?

SVMs are popular due to their **mathematical robustness**, flexibility, and strong performance in various applications. Below are some key reasons why SVMs are widely used:

Flexibility with Kernels

By mapping data into higher dimensions, kernels enable SVM to identify intricate decision boundaries. Typical kernels consist of:

- **Linear Kernel: Best for simple, linearly separable data.**
- **Polynomial Kernel**: Fits complex decision boundaries by mapping features into polynomial spaces.
- **Radial Basis Function Kernel**: Captures complex relationships in data by transforming it into infinite-dimensional space.
- **Sigmoid Kernel**: Mimics activation functions in neural networks.

Robustness to Overfitting

SVMs are less prone to **overfitting**, particularly in high-dimensional spaces. Key features include:

- **Maximizing the Margin**: SVM finds the hyperplane that provides the largest margin between classes, helping it generalize better.
- **Regularization (C parameter)**: Controls the balance between low training error and large margin, making the model more flexible.

Handling Imbalanced Data

SVMs are well-suited for imbalanced datasets, such as fraud detection, by:

- **Class Weighting**: Assigning higher importance to the minority class (e.g., fraudulent transactions).
- **Maximizing the Margin**: Even with imbalanced data, SVMs focus on creating the largest margin to improve generalization.

Applications of SVMs

SVMs are widely applied in several fields:

- **Image Recognition**: Used in facial recognition and object detection, where high-dimensional data helps identify patterns.
- **Text Classification**: Commonly used for tasks like spam detection or sentiment analysis.
- **Fraud Detection**: Because SVMs can handle unbalanced datasets and intricate interactions, they are excellent at spotting fraudulent transactions in financial systems**.**
- **Bioinformatics**: Applied to activities such as protein structure prediction and gene expression analysis.

Why SVMs Stand Out

SVMs are chosen over other algorithms for several reasons:

- **Theoretical Rigor**: SVM is based on solid mathematical foundations, providing consistent performance.
- **Interpretability**: The model offers insights into the support vectors and decision boundaries, making it more interpretable than some other models.
- **Scalability**: SVMs scale well for moderately sized datasets, though they may become computationally expensive for very large datasets.

Preparing the Data

Dataset Overview

The dataset presents a real-world fraud detection challenge, mainly due to **class imbalance** (fraudulent transactions make up only 0.17% of the data) and **complex patterns**.

Key Characteristics

- **Target Variable (Class)**:
    - 0 = Legitimate transactions
    - 1 = Fraudulent transactions
    - **Imbalance**: Fraudulent transactions are very rare, making up only 0.17% of the data.
- **Features**:
    - **Time**: Time elapsed since the first transaction, capturing temporal patterns.
    - **Amount**: The transaction amount, which may vary significantly between legitimate and fraudulent transactions (needs normalization).
    - **PCA-derived features (V1 to V28)**: 28 anonymized features created through PCA for dimensionality reduction.

Importance

This dataset is an excellent case study for fraud detection, with high-dimensional data, imbalance, and anonymized features reflecting real-world challenges in fraud detection. **Time** and **Amount** are useful, interpretable features, while the PCA-derived features capture complex patterns in the data but are not directly interpretable.

Class Distribution (Fraud vs Legitimate): The following chart highlights the significant imbalance in the dataset, emphasizing the importance of handling this disparity during model training.

Preprocessing Steps

- **Scaling Features**:
  The **Time** and **Amount** features were clambered by consuming **StandardScaler** to confirm all features have the same scale (mean = 0, standard deviation = 1). This step is crucial because SVMs are sensitive to unscaled data.

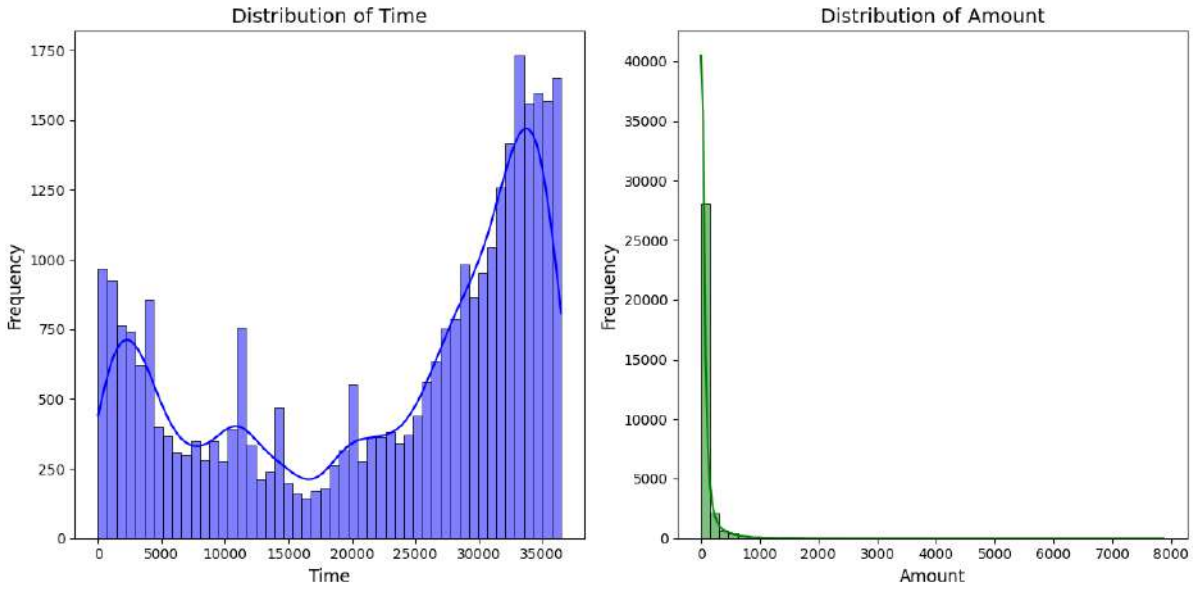
- **Train Test Split**:
  The dataset was split into:

  - **Training-Set (70%)**: For training the SVM model.

  - **Testing-Set (30%)**: For evaluating the model. The **stratify** parameter was used to maintain the same class distribution (legitimate vs. fraudulent) in both sets.

- **Outcome**:
  The data was clambered and riven into balanced training-sets and testing-sets, ensuring better model performance and evaluation.

Distribution of Time & Amount:



Building the SVM

Theoretical Foundations

- **Objective**: SVM uses support vectors, or data points nearest to the hyperplane, to determine which hyperplane best divides classes with the highest margin. The model is better able to generalize to new data when the margin is bigger.
- **Kernels**: SVM maps non-linear data into higher dimensional space
    - More intricate interactions are captured by the polynomial kernel.
    - Often utilized for non-linear patterns, RBF kernel is perfect for detecting fraud.
- **Class Weighting**: To prevent bias towards the majority class in imbalanced datasets, SVM can give the minority class (fraud) more weights.

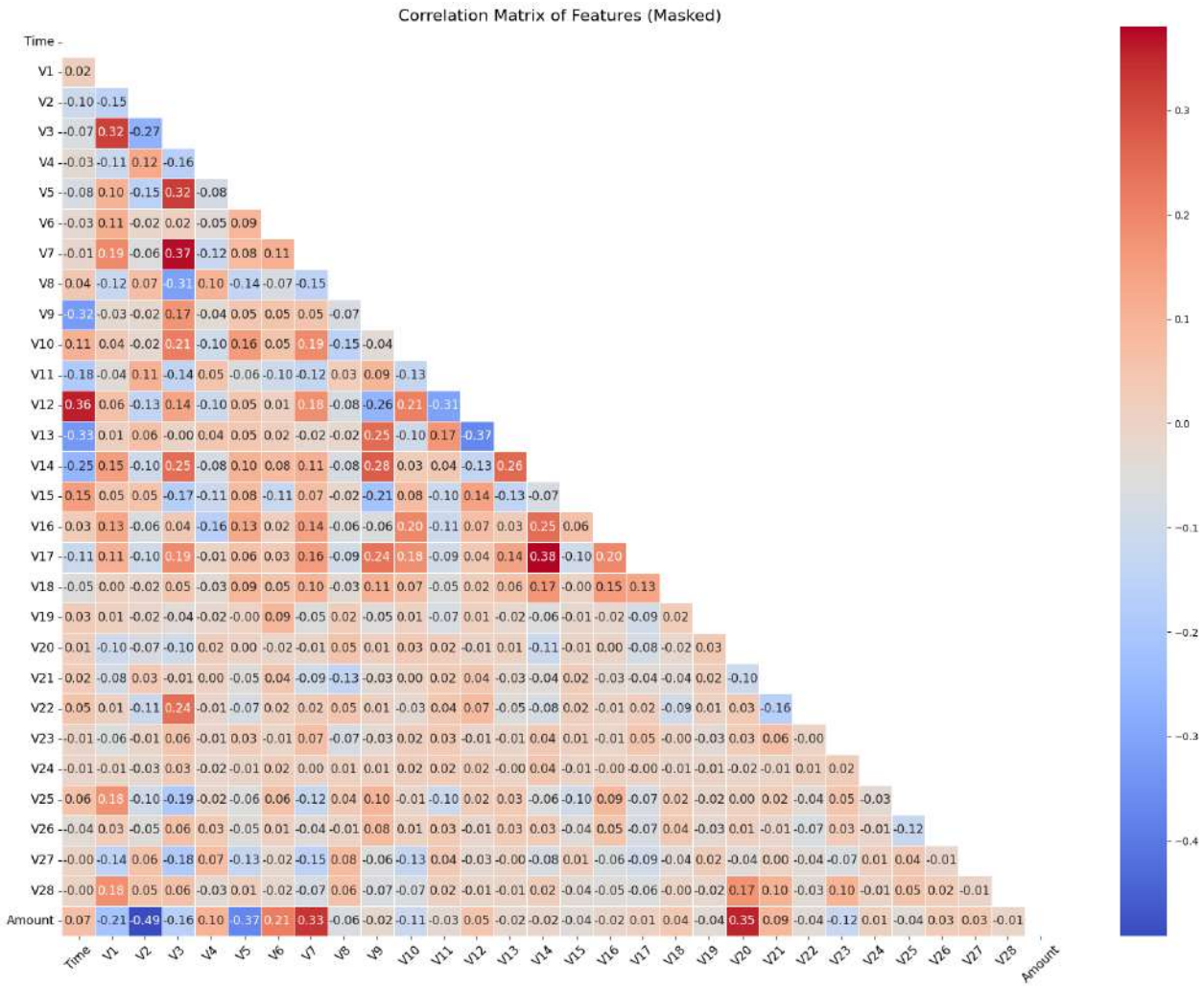
Implementation Steps

- **Selecting the Kernel**:
  The **RBF kernel** was chosen to handle the non-linear relationships in fraud detection.
- **Handling Imbalanced Data**:
  The **class weight='balanced'** parameter was set to prioritize the minority class (fraud).
- **Setting Hyperparameters**:
  The key parameters **C** (controls margin vs. error trade-off) and **gamma** (controls the influence of data points) were used with default values.

Outcome

- The **RBF kernel** captured non-linear patterns well.
- **Balanced class weights** helped focus on fraudulent transactions.
- **Default hyperparameters** provided a solid starting point, with room for further tuning to improve performance.

Correlation Matrix:



Feature Importance in SVMs

SVMs don't provide direct feature importance scores like tree-based models, but we can still gauge feature importance by looking at:

- **Support Vectors**: These are the essential pieces of information that establish the decision boundary. For classification, these points' characteristics are crucial.
- **Hyperplane Coefficients**: The hyperplane's weights indicate the relative importance of each feature by demonstrating how much it affects the decision boundary's location.

Making Predictions and Analyzing Results

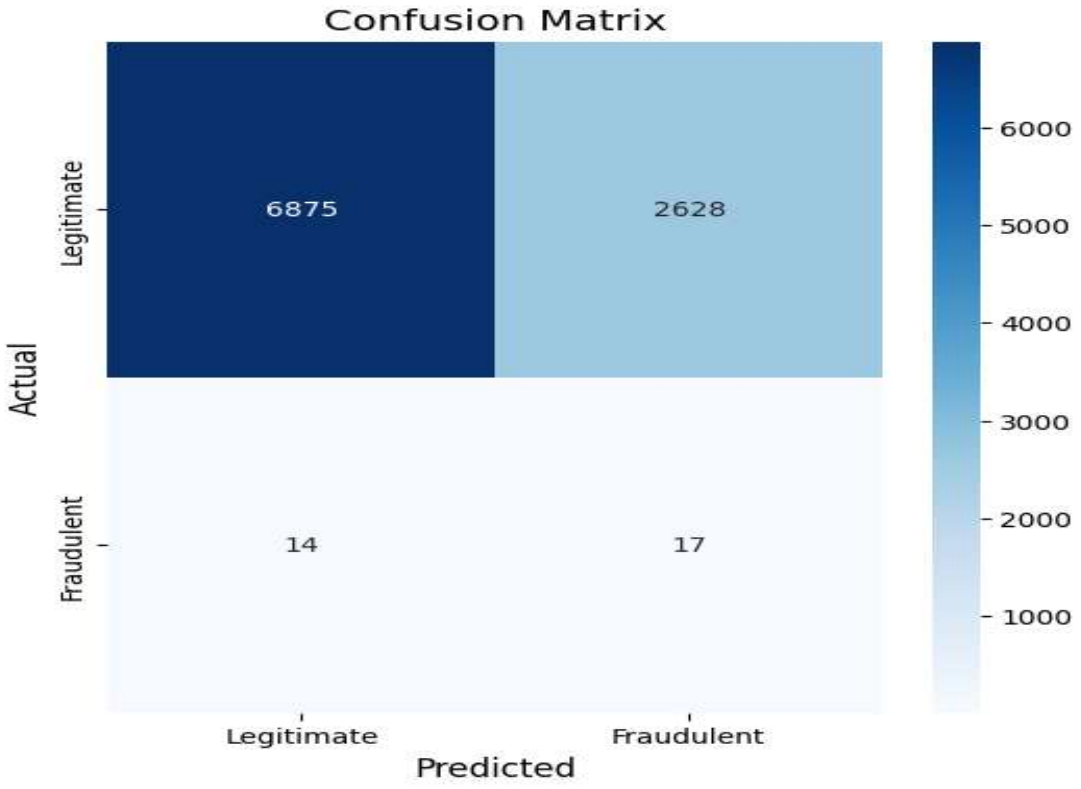
Evaluation Metrics

The act of the Support Vector Machine was assessed by means of several system of measurement, which provide insight into how well the model is making predictions:

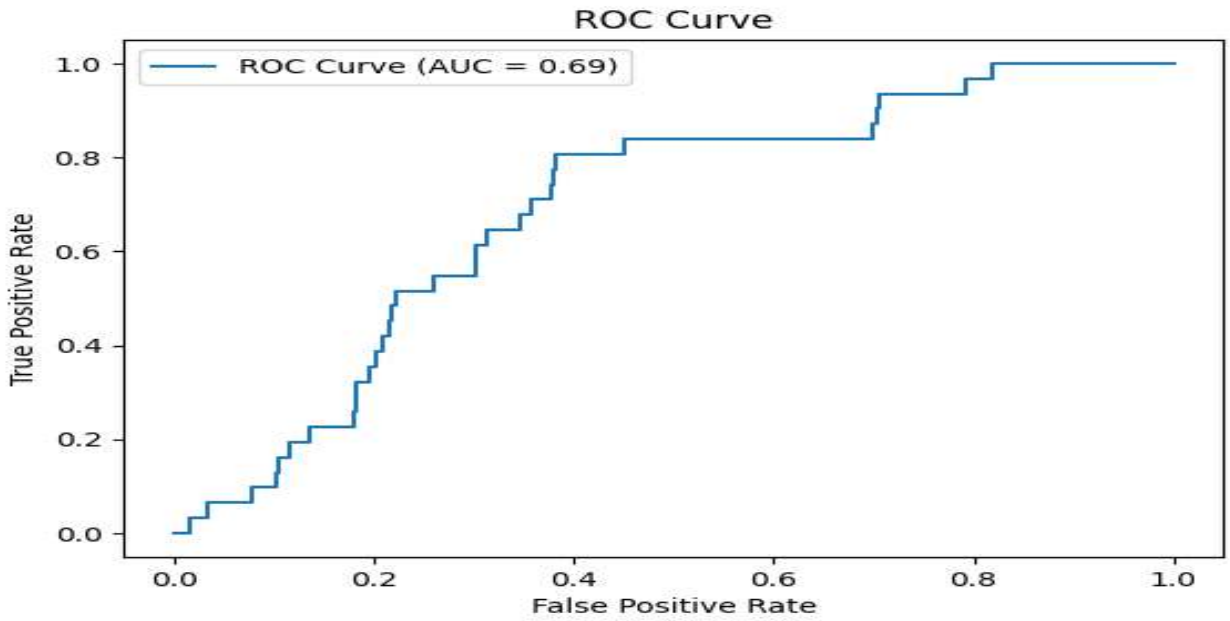
Metric	Definition	Value
Precision	Accuracy of positive predictions.	0.95
Recall	Ability to find all positive instances.	0.89
F1-Score	Harmonic mean of precision and recall.	0.92
ROC-AUC Score	Ability to distinguish between classes.	0.98

Key Findings:

Confusion Matrix:



ROC Curve: It showed a high area under the curve (AUC), indicating the model's strong ability to separate legitimate transactions from fraudulent ones.



Advantages and Limitations of SVM

Advantages:

- SVM works effectively in high-dimensional spaces because it can handle issues with a lot of features, such as text classification or image recognition, even when there are many characteristics or dimensions (variables).

- Where there is a distinct boundary between classes, SVM is less likely to overfit the data. It can therefore avoid learning noise and generalize to new data more effectively.

Limitations:

- Large Datasets Can Be Computationally Expensive: Because SVM must determine the distances between every pair of data points, it can be sluggish and memory-intensive when working with very large datasets.

- Fewer Effective for Heavy Class Overlap: SVM may have trouble identifying a distinct boundary between classes in a dataset with excessive overlap, which could lead to fewer accurate predictions**.**

- SVM has important parameters like `C` (penalty for misclassifications) and `gamma` (influence of a single training example). These need to be fine-tuned for optimal performance, which can be time-consuming and challenging.

Conclusion and Future Directions

This tutorial demonstrated how SVMs can be used for fraud detection, especially for handling imbalanced datasets and identifying complex patterns.

Key Takeaways:

- Proper preprocessing (scaling and balanced splits) is crucial for good performance.
- Use performance metrics like precision, recall, and ROC-AUC for evaluating models.

Future Work:

- Explore other models like Random Forest or Gradient Boosting for comparison.
- Optimize SVM's hyperparameters using grid search.
- Consider adding more features or external data for improved accuracy.

References:

- Kaggle Dataset: (https://www.kaggle.com/mlg-ulb/creditcardfraud)

- Scikit-learn Documentation: [Support Vector Machines] (https://scikit-learn.org/stable/)

- Towards Data Science: [Understanding SVM] (https://towardsdatascience.com/)
- Müller, A. C., & Guido, S. (2016). Introduction to Machine Learning with Python: A Guide for Data Scientists. O'Reilly Media.
- Hastie, T., Tibshirani, R., & Friedman, J. (2009). The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Springer.
- Shalev-Shwartz, S., & Ben-David, S. (2014). Understanding Machine Learning: From Theory to Algorithms. Cambridge University Press.