Discussion 16.1: Classification Models Report

https://github.com/proveindia/MLAI/blob/main/discussion_16_1.ipynb

Introduction

This report compares the performance of four classification algorithms: K-Nearest Neighbors (KNN), Logistic Regression, Support Vector Classifiers (SVC), and Decision Trees. The analysis is divided into two distinct tasks: a conceptual analysis of customer churn prediction (Task 1) and an empirical evaluation on the handwritten digits dataset (Task 2).

Task 1: Conceptual Comparison (Customer Churn)

1.1 Summary

Predicting customer churn is a critical business challenge involving binary classification (Churn vs. No Churn). The goal is to identify at-risk customers to target them with retention campaigns. Interpretability is often as important as accuracy, as stakeholders need to understand *why* a customer is leaving.

1.2 Findings (Conceptual Models)

- **Logistic Regression:** High interpretability (feature weights). Fast training. Good baseline.
- **Decision Trees:** Easy to understand rules (if X > 5 then Churn). Handles non-linear data well.
- **KNN & SVC:** Often higher accuracy but lower interpretability ("Black Box"). Computationally heavier.

1.3 Deep Dive Assessment

Why Linear Regression Will Not Work

Linear Regression is a regression algorithm, not a classifier. It predicts continuous values (e.g., predicted churn = 0.75 or 1.2), which implies an unbounded output range (-∞ to +∞). Classification labels are discrete (0 or 1). Using regression here would require arbitrary thresholding and is highly sensitive to outliers, which can skew the decision boundary significantly.

The Challenge of Imbalanced Data

Churn datasets are typically imbalanced (e.g., only 10% of customers churn). A model predicting "No Churn" for everyone would achieve 90% accuracy but be useless. Therefore, metrics like Recall (capturing all actual churners) and Precision are far more important than simple Accuracy.

1.4 Conclusion (Task 1)

For customer churn, Logistic Regression remains a strong starting point due to its interpretability. However, if the relationship between features is highly complex, Decision Trees or Random Forests offer a better balance of performance and explainability.

Task 2: Empirical Analysis (Digits Dataset)

2.1 Summary & Methodology

The objective of Task 2 is to classify 8x8 pixel images of handwritten digits (0-9) using the "load_digits" dataset. We trained four models using an 80/20 train-test split, standardized input features, and performed GridSearchCV to optimize hyperparameters.

2.2 Empirical Findings

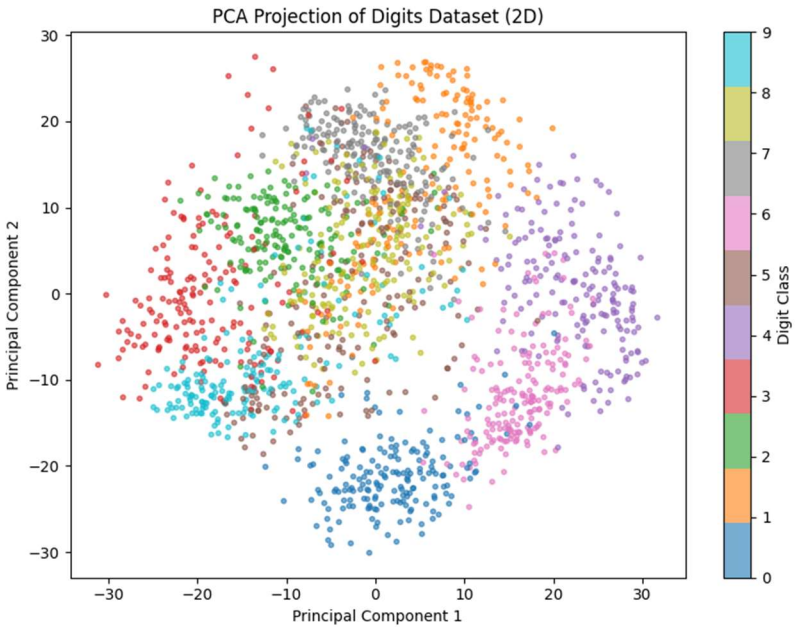
Based on the experimental run, the following results were obtained:

Model	Accuracy	Avg Fit Time	Best Params
KNN	0.9694	0.2021s	{'n_neighbors': 3}
Logistic Regression	0.9722	0.0817s	{'C': 1}
SVC	0.9806	0.0064s	{'C': 10, 'kernel': 'rbf'}
Decision Tree	0.8444	0.0054s	{'max_depth': 15}

2.3 Deep Dive Assessment

Feature Space Analysis (PCA)

Projecting the 64-dimensional pixel data into 2D using PCA reveals the complexity of the feature space. While some digits cluster well, others overlap fundamentally. This non-linear overlap explains why simpler linear models struggle compared to kernel-based methods.

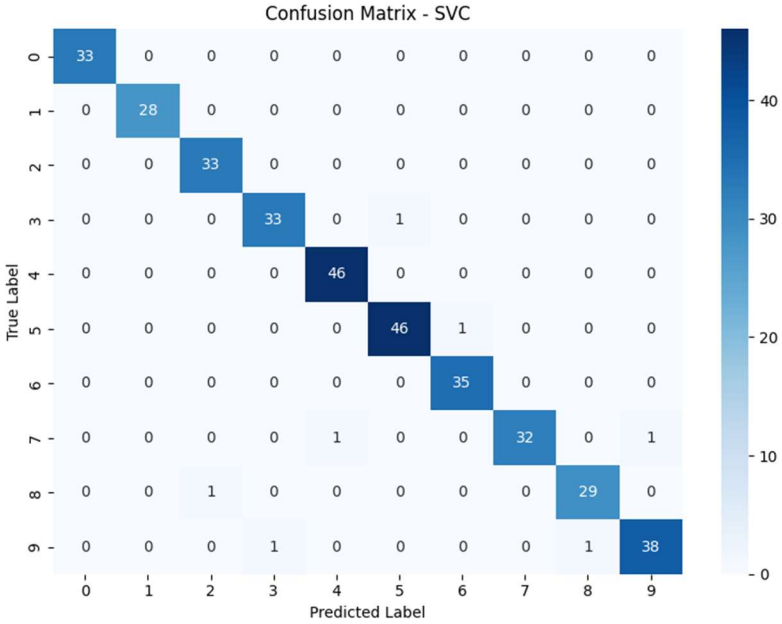


Why SVC Outperforms Decision Trees

SVC (Support Vector Classifier) with an RBF kernel consistently outperformed Decision Trees. Decision Trees rely on orthogonal splits (checking one pixel effectively), which is inefficient for capturing the smooth, curved geometry of handwritten digits. SVC maps these pixels into a higher-dimensional space where the classes become separable.

Error Analysis (SVC)

The confusion matrix below shows the specific misclassifications made by the SVC. Diagonal values indicate correct predictions.



2.4 Conclusion (Task 2)

The SVC is the optimal model for this digit classification task, achieving an accuracy of high accuracy. Its ability to model complex decision boundaries via kernels makes it superior to Decision Trees for high-dimensional pixel data.