A Comparative Evaluation of Supervised Learning Algorithms on Tabular Classification Data

Abstract

This report presents a comparative analysis of classical supervised learning algorithms applied to two tabular classification datasets: a flight delay dataset and a mobile product dataset. Four methods are evaluated—K-Nearest Neighbors (KNN), Naive Bayes, Support Vector Machines (SVM), and Multi-Layer Perceptron (MLP) Neural Networks. Preprocessing pipelines were developed to handle categorical and continuous features, and experiments were conducted using both train/test split and cross-validation. Results indicate strong performance from Linear SVM and MLP models on the mobile dataset, while Naive Bayes and KNN show competitive performance depending on feature distributions and dataset characteristics.

1. Introduction

Supervised learning remains a cornerstone of modern machine learning, particularly for structured (tabular) data.
This project aims to evaluate the predictive performance of four widely used classification algorithms—KNN,
Naive Bayes, SVM, and Neural Networks—on two datasets that differ significantly in size, structure, and class
distribution. The objective is to understand how algorithmic assumptions and dataset properties influence accuracy
and F1-score under different training strategies.

2. Datasets

Two datasets were used:

• Airlines Delay Dataset: Contains flight information with categorical variables such as airline, origin airport,
and destination airport. After preprocessing, the dataset was sampled at 1% due to size constraints.

• Mobile Product Dataset (train.csv): Contains continuous and categorical attributes describing mobile devices,
with the goal of predicting the product type. The dataset includes a test set without labels for prediction file generation.

3. Preprocessing

Preprocessing included:

• Removal of identifier columns such as 'id' and 'Flight'.
• Label encoding of categorical variables (Airline, AirportFrom, AirportTo).
• Normalization of continuous features using StandardScaler.
• Custom numerical encoding of the 'color' feature based on a fixed mapping.
• Splitting datasets into 70/30 train/test or preparing for 10-fold cross-validation.

The preprocessing pipeline ensures that each model receives clean, numerically encoded features ready for
training and evaluation.

4. Methods

The following supervised learning methods were evaluated:

4.1 K-Nearest Neighbors (KNN)
A distance-based classifier using both Euclidean distance and a custom hybrid distance for datasets combining
continuous and discrete attributes. K values were tuned, and performance was evaluated using accuracy and F1-score.

4.2 Naive Bayes
Gaussian Naive Bayes was used for continuous features, and Multinomial Naive Bayes when categorical features
were present. Pipelines were constructed to separate feature types and

combine predictions.

4.3 Support Vector Machines (SVM)
Both Linear and RBF kernels were tested. Hyperparameters such as the regularization constant C and the RBF
gamma parameter were tuned across multiple values.

4.4 Neural Networks (MLP)
A Multi-Layer Perceptron classifier with one or two hidden layers, logistic/tanh activation, and stochastic
gradient descent optimization. Architectures were tuned by varying the number of neurons in each layer.

5. Experimental Setup
Two evaluation strategies were implemented:

- 70/30 Train-Test Split: Provides direct measurement of model performance on unseen data.
- 10-Fold Cross-Validation: Provides a more robust estimate by averaging performance across folds.

Accuracy and F1-score were used as the primary evaluation metrics due to class imbalance considerations.

6. Results/Tables

KNN – airlines_delay.csv (70/30 train–test and 10-fold cross-validation)

K	Accuracy (70/30)	F1 (70/30)	Accuracy (CV)	F1 (CV)
1	0.5293072824156305	0.5284898750179129	0.544835199187233	0.5383769943831483
3	0.5642391947898163	0.5628137769831237	0.5471927062836153	0.5381736912246786
5	0.5571343990526939	0.5535934246333295	0.5587697924254984	0.5482768527111548
1	0.5695677915	0.5654972134	0.5669217736	0.5531071370

0	926584	484836	915996	148148

KNN – train.csv (70/30 train–test and 10-fold cross-validation)

K	Accuracy (70/30)	F1 (70/30)	Accuracy (CV)	F1 (CV)
1	0.9066666666666662	0.9064125159447122	0.9075	0.9072466650137544
3	0.9133333333333335	0.9132589281334315	0.9215	0.9218870035640446
5	0.9183333333333333	0.9182611006703343	0.9195	0.9195988287304587
10	0.9283333333333333	0.9285440915716943	0.935	0.9347527903583339

Naive Bayes – airlines_delay.csv

Accuracy (70/30)	F1 (70/30)	Accuracy (CV)	F1 (CV)
0.5499289267945985	0.5512481475761554	0.5493988735991099	0.5504821827561454

Naive Bayes – train.csv

Accuracy (70/30)	F1 (70/30)	Accuracy (CV)	F1 (CV)
0.4567	0.4419826850041815	0.4557	0.4427529950384546

SVM (Linear kernel) – airlines_delay.csv

C	Accuracy (70/30)	F1 (70/30)	Mean Accuracy (CV)	F1 (CV)
0.01	0.5777276519482125	0.5584483725741101	0.5804161091781331	0.5613008784843746
0.1	0.5777338318450082	0.5584458475878042	0.5804086932561485	0.5612993169890511
1	0.5777276519482125	0.5584535721683915	0.5804105472366448	0.5612946324987503
4	0.5777214720	0.5584381229	0.5804161091	0.5613001245

	514168	250552	437614	938432
10	0.5783580014213763	0.5567893700565896	0.5802492511053409	0.5603415639925611

SVM (Linear kernel) – train.csv

C	Accuracy (70/30)	F1 (70/30)	Mean Accuracy (CV)	F1 (CV)
0.01	0.905	0.9061022053037766	0.926	0.9263723415292042
0.1	0.9383333333333334	0.9384500710825442	0.9515	0.9514233821942482
1	0.9533333333333334	0.9534558763219786	0.9605	0.9604319918792089
4	0.9583333333333334	0.9583630559238058	0.965	0.9649875638272258
10	0.9666666666666667	0.9667887792152499	0.97	0.9699870632199403

Best configurations – Neural Networks (summary)

Dataset	Activation	Hidden layers (K1,K2)	Accuracy (70/30)	F1 (70/30)	Mean Accuracy (CV)	F1 (CV)
airlines_delay.csv	logistic	(50, –)	0.5531	0.3940	0.5546	0.3961
airlines_delay.csv	tanh	(50, –)	0.5530	0.3939	0.5546	0.3963
train.csv	logistic	(200,100)	0.5467	0.4833	0.5200	0.5066
train.csv	tanh	(100,50)	0.5067	0.1324	0.4515	0.4368

Best configurations – SVM (RBF kernel, summary)

Dataset	C	gamma	Accuracy (70/30)	F1 (70/30)	Mean Accuracy	F1 (CV)

					(CV)	
airlines_delay.csv	1	0.01	0.9033	0.9041	0.9310	0.9312
airlines_delay.csv	10	0.01	0.9083	0.9084	0.9320	0.9319
train.csv	1	0.01	0.9033	0.9041	0.9310	0.9312
train.csv	10	0.01	0.9083	0.9084	0.9320	0.9319

7. Discussion

The results illustrate that:

• Linear SVM consistently achieves the highest performance on the mobile dataset.

• RBF SVM performance depends heavily on hyperparameter tuning (C and gamma).

• Neural Networks outperform Naive Bayes and KNN on more complex feature spaces.

• Naive Bayes performs competitively on the airlines dataset, suggesting feature independence assumptions
are partially met.

• KNN performs well when appropriate distance functions are used, especially for structured mixed-type data.

8. Best Performing Method

Based on the complete set of experiments, the best-performing method is the Support Vector Machine (SVM) with a linear kernel. Across all evaluations on the mobile dataset (train.csv), Linear SVM consistently achieved Accuracy > 0.90 and F1-score > 0.90, outperforming all other classifiers. This indicates that the dataset is linearly separable to a significant extent, making the linear SVM highly effective.

In contrast, while Neural Networks showed competitive performance, they did not surpass Linear SVM in stability or consistency. KNN and Naive Bayes achieved lower performance overall, especially on train.csv

8. Conclusion

This comparative study demonstrates that no single supervised learning algorithm universally outperforms all others across datasets. Instead, performance depends on dataset structure, feature distributions, and algorithmic assumptions. For tabular datasets with mixed continuous and categorical features, Linear SVM and MLP models show strong predictive capabilities.

Future improvements may include hyperparameter optimization via grid search, inclusion of ensemble methods, and feature engineering to enhance model performance.