Analysis of Machine Learning Models on Binary and Multiclass Classification Datasets

Parsa Jafaripour and Robbie Bernstein
CSC 156
December 2025

Abstract

This paper explores the training and analysis of two datasets: a binary classification dataset, which have truth labels consisting of True or False labels, and a multiclass dataset, which classifies data into one of four classes. Each dataset is trained on three models, and the training and testing results of each are measured to decide which model is most suitable for each dataset and the overall training performance of each dataset.

Introduction

Machine learning allows humans to predict outcomes in ways that were not possible before. It has various applications in financial, medical, athletic, and political sectors. In this project, we analyze two datasets by training each on three models and determine the best model for each dataset. The first dataset is a binary classification dataset, the ISOT Fake News Detection dataset, which classifies news articles as either real or fake news. The models used to train this set are linear regression, multinomial Naive Bayes, and random forest. The second dataset is a multiclass classification dataset, the Customer Segmentation dataset, which predicts which group a particular customer fits into to predict purchasing habits and target marketing campaigns. Random forest, support vector machine, and artificial neural network are used to train this dataset. We will start by introducing these datasets in more detail, as well as the methods used for training and analysis. We will then discuss the hyperparameter tuning used on each model and the results of the optimal model given the best hyperparameters found during tuning. We will conclude with a discussion of the results, our takeaways from this project, and things we would improve during further analysis.

Datasets

Binary Dataset: ISOT Fake News Detection Dataset

The ISOT Fake News Detection Dataset is a binary dataset that attempts to predict whether a given news article is “real” news or “fake” news. It uses contents of the article and patterns related to headlines or statements to decide the validity of the information given in the article, then labels it as either “real” or “fake”.

The dataset contains a total of 44,919 instances, with 21,417 “real” articles and 23,502 “fake” articles, and four features that consist of information related to the article. The title is the headline of the article, and

the text contains the body. There are also the subject of the article and the date it was published, however these are likely to have minimal impact on whether an article is real or fake news. There is minimal missing data, with around 3% instances not having an article body, and there are also some mismatched dates in the fake news dataset.

Multiclass Dataset: The Customer Segmentation Dataset

The Customer Segmentation dataset is provided by an automobile company as part of an effort to optimize outreach by segmenting their customers. The primary task with this dataset is to classify each individual into one of four possible customer segments using a variety of personal and behavioral features. There are a total of 2,627 examples, each representing a potential or active customer.

The dataset's features include customer ID, gender, marital status, age, graduation status, profession, work experience in years, spending score, family size, and a categorical variable labeled Var_1. The gender feature is split approximately evenly between male (54%) and female (46%). Marital status is recorded as true for 58% of instances, false for 40%, with about 2% missing values. The age range spans from 18 to 89, with most individuals falling into the 25-55 year-old category. Graduation status is marked true for around 61% of examples, false for 38%, and 1% are missing. Profession is a categorical feature, with Artist and Healthcare being notable values. Work experience ranges from zero up to fourteen years, but the majority of customers report low levels of experience, typically around zero or one year. The spending score is divided into categories, with the lowest category making up approximately 62% of entries, the average group accounting for 24%, and all other categories representing 14%. Family size ranges from very small (one) to fairly large (nine), but most families are small to moderate in size.

Some features, such as marital status and graduation status, contain missing or null values. Categorical features like profession and Var_1 are imbalanced, meaning one category makes up the majority of cases. Appropriate preprocessing will be required to address missing data.

Methods

Fake News Dataset

For the binary classification dataset, the three models we will be testing are logistic regression, multinomial Naive Bayes, and random forest. The text samples will be vectorized into interpretable form using Scikit-learn's TF-IDF vectorizer, and is done beforehand and saved as a new dataset so that the vectorized data does not need to be recalculated for every training instance. Additionally, k-fold cross validation is used during training for each of these models. 5 folds was chosen, as it balances having a more complex training system and reducing training time.

Logistic regression is fast to train and predict, and it is easy to regularize, which will be beneficial because of the large size of the dataset. It performs well on highly dimensional data such as news articles. The main hyperparameters we will be tuning are regularization strength C, which helps reduce overfitting,

and the type of logistic regression solver used, as Scikit-learn provides several types of solvers and one may perform better on this dataset than another.

The second model we will test is multinomial Naive Bayes. It outputs probabilities, which will be helpful in determining the strength of the model, and it is computationally efficient, which will be beneficial for a large dataset such as this one. The hyperparameters we will tune are the smoothing factor, which may have some effect on the probabilities, and whether or not the model learns class priors. Multinomial Naive Bayes was chosen over models such as Bernoulli or Gaussian because of its superior performance on text analysis.

The final model we will train is random forest. Random forest combines several high-variance decision trees and averages their predictions, which improves upon using one decision tree and provides a very strong model. The hyperparameters to tune are maximum tree depth, which controls at what depth the model stops adding nodes, and the number of trees in the random forest. Each of these hyperparameters control the complexity of the model.

The dataset is split into two CSV files, one containing “real” news articles and the other containing “fake” news articles. We will take each of these files and split them into 80% training and 20% testing, then combine each training and testing dataset with the appropriate labels so that we have one training set and one testing set.

The two main metrics we will use to measure the strength of these models are accuracy and recall. Since the dataset is balanced, we do not need to worry about the downsides that come with using accuracy. We will also be using recall over precision because we would prefer false positives over false negatives. In other words, it is better to label a real news article as fake rather than label a fake article as real.

Customer Segmentation Dataset

The three models we will train on our multiclass classification dataset, the Customer Segmentation Dataset, will be random forest, support vector machine, and artificial neural network. Each of these models are able to learn non-linear decision boundaries and can handle multiple classes.

The first model we will train is random forest. It naturally supports multiclass datasets, and as explained for the binary dataset, provides a stronger model than simply using a decision tree. The same hyperparameters tuned for the binary dataset will be tuned for this dataset, the maximum tree depth and the number of trees.

The second model to train is support vector machines with an RBF kernel. We chose the RBF kernel because it works well with higher dimensional data and provides more complex models than what linear or polynomial kernels provide. The hyperparameters that will be tuned are the regularization strength and the gamma value, which impacts the transformation calculated by the kernel.

The final model we will be training is artificial neural network. Neural networks provide a more complex and sophisticated model compared to random forest or SVM. The hyperparameters we will tune are the number and size of hidden layers, which will affect the complexity of the model and may keep the model from being unable to learn the data, and the learning rate, which will help to tune the model to be able to learn the optimal model.

The dataset comes pre-split into train and test sets, so no additional data processing is necessary.

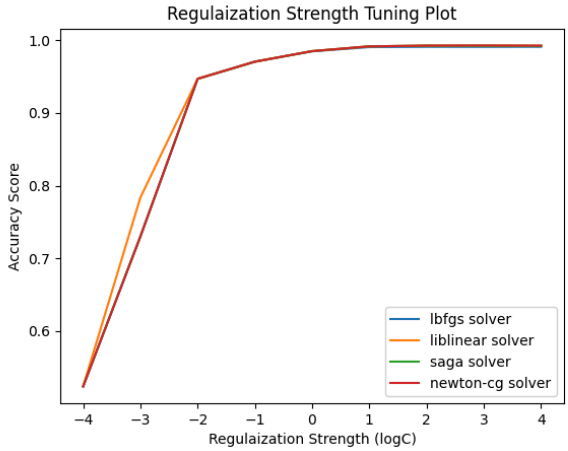
The two metrics we will use to measure each model are accuracy and F1 score. Accuracy provides a quick and easy way to measure the strength of a model, but may be misleading if the dataset is unbalanced. F1 score provides a more reliable score for datasets that are unbalanced, and since we do not prefer either false positives or negatives, we can use F1 to balance precision and recall which will give us a more general idea of the strength of each model.

Hyperparameter Tuning

Fake News Dataset

Logistic Regression

The two hyperparameters that we tuned for the logistic regression model are regularization strength (C) and the type of model used. Regularization strength reduces overfitting by penalizing the weights of the parameters. This value is varied from 10^{-4} to 10^{4}. The other hyperparameter is the model type. We tested four types of Scikit-learn's provided solvers: lbfgs, liblinear, saga, and newton-cg. Below is the hyperparameter tuning curve, with a curve for each solver, regularization strength plotted on the x-axis and scaled by log, and accuracy score plotted on the y-axis.

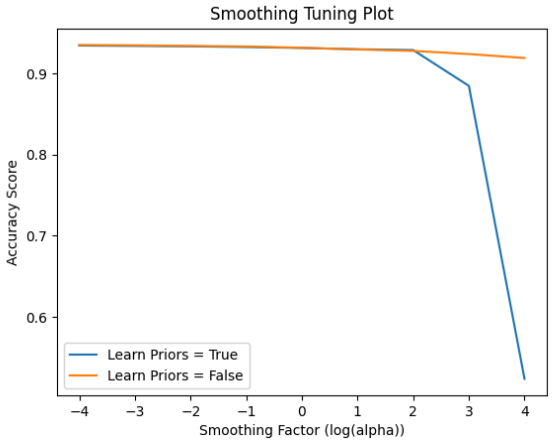


As regularization is increased, meaning regularization strength decreases, the highest cross validation fold accuracy increases. With low values of C, the model becomes underfit, which results in low accuracies. As you increase C, this accuracy increases as underfitting occurs, however you will eventually overfit as

C gets high. In this case, the C value plateaus off at a C value of around 10 and peaks at C = 1000. For the solvers, there isn't much of a difference between each as they mostly overlap on the plot above. The optimal combination of C and solver is the newton-cg solver with C = 1000, providing an accuracy of 0.9930.

Naive Bayes

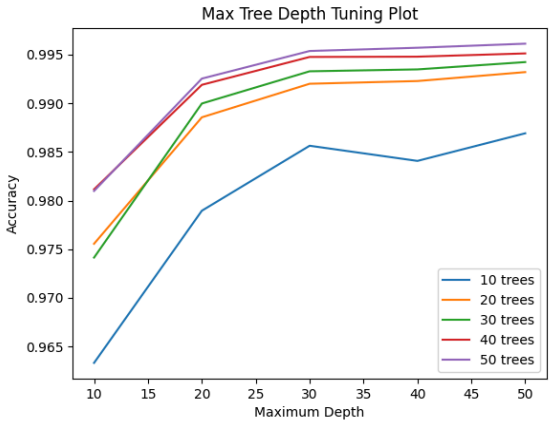
The multinomial Naive Bayes model includes two hyperaparameters to tune. The most important hyperparameter is the smoothing factor. This keeps the class probabilities from being calculated as 0, and is varied from 10^{-4} to 10^{4}. The other hyperparameter is whether or not the model learns prior probabilities. These are plotted below as separate curves, with the blue curve representing True, as in priors are learned, and the orange curve representing False, as in priors are not learned. The smoothing factors (alpha) are plotted on the x-axis and scaled by log, and the accuracies are plotted on the y-axis.



Similar to the logistic regression solver tuning plot, there is a small if any difference between learning priors or not, up until alpha is around 100. As alpha increases, the accuracy score decreases, albeit slowly. The exception is in the True curve when the accuracy decreases sharply after alpha = 10. The optimal hyperparameter combination is alpha = 10^{-4} and learn priors = False, with an accuracy of 0.9352.

Random Forest

The two hyperparameters to tune for random forest are tree depth and number of trees. Each of these hyperparameters increase the complexity of the model. The number of trees is varied from 10 to 50, and max depth is varied from 10 to unconstrained, but plotted up to 50. Each of the values for the number of trees is plotted as a separate curve on the plot below. Maximum depth is plotted on the x-axis, and accuracy is plotted on the y-axis.

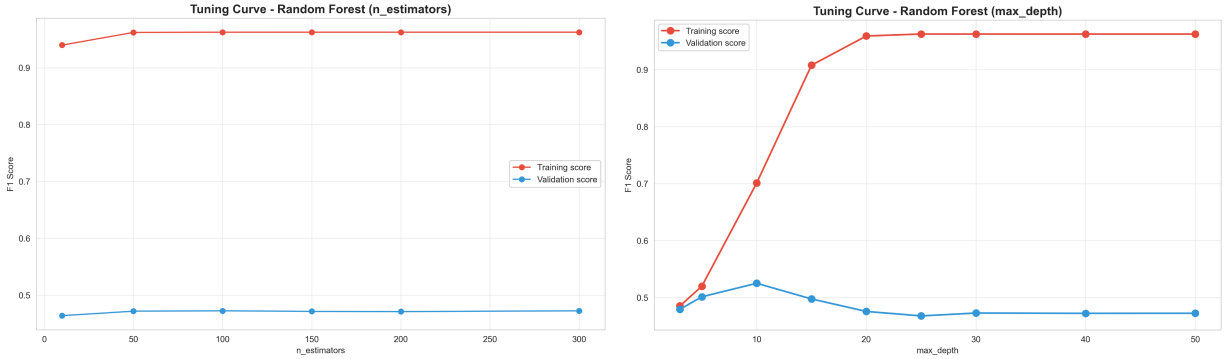


The general trend for these hyperparameters is as they increase, accuracy increases as well. The optimal model consists of 50 trees and an unconstrained tree depth with an accuracy of 0.9964. As noted above, increasing the amount of trees would likely give a better model, but even small increases in the tuning range greatly increase training time, as this model took longer to tune than the previous two.

Customer Segmentation Dataset

Random Forest

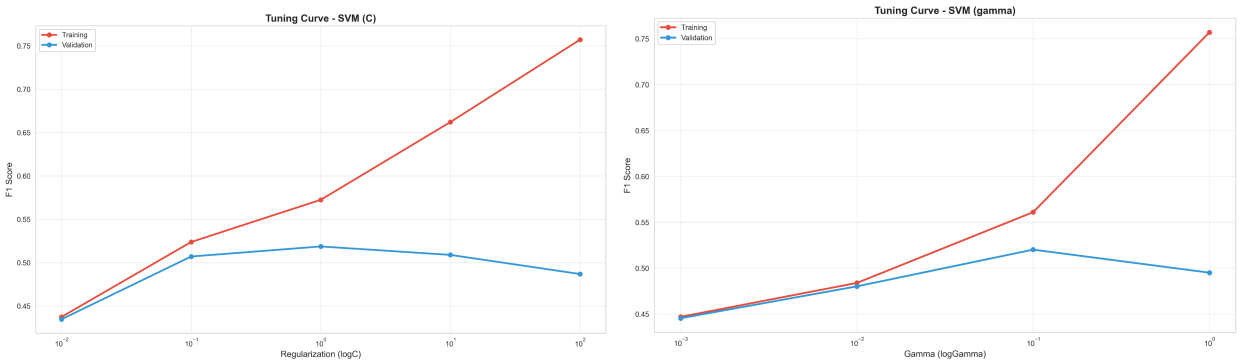
The hyperparameters tuned for random forest are maximum tree depth and the number of trees in the random forest. These hyperparameters increase or decrease the complexity of the tree. The number of trees is varied from 10 to 300, and maximum depth is varied from 3 to 50. Below are two plots, one for depth tuning, and the other for number of trees (n_estimators) tuning, with the hyperparameters on the x-axis and weighted F1 score on the y-axis.



Both of these curves show signs of overfitting. In both cases, the training scores are close to 1, while the validation scores remain below 0.5. There is also little variation in F1 scores as the magnitude of the hyperparameters increase, although greater values of number of trees and max depth correspond with slightly higher F1 scores. This is evidence that even the optimal model, which had the hyperparameters max depth = 10, number of trees = 100, and an accuracy of 0.6828, is not a strong model for this dataset.

Support Vector Machine

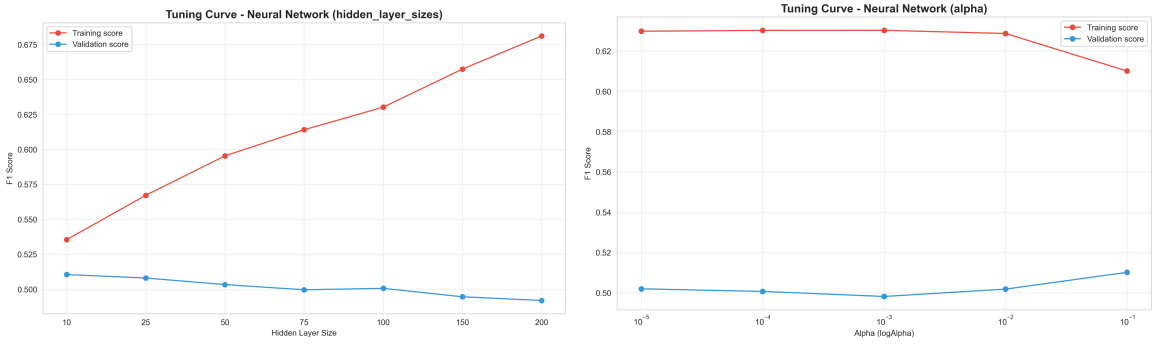
The hyperparameters tuned for SVM are the regularization strength (C) and gamma. C helps balance underfitting and overfitting of models, while gamma impacts that calculation of the RBF kernel. C is varied from 0.01 to 100, and gamma is varied from 10^{-3} to 1. These are plotted on the x-axis, with F1 score plotted on the y-axis.

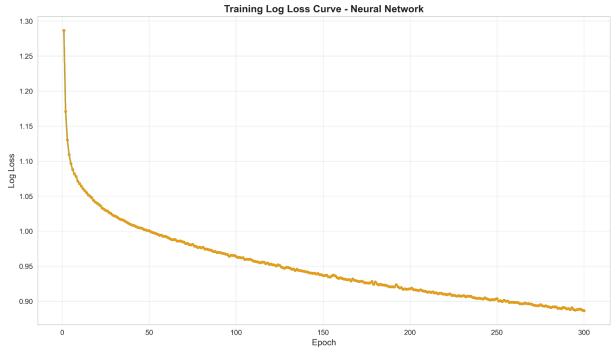


Similarly to random forest, this model did not perform well on this dataset. The overfitting concerns are not as evident here, as the validation F1 scores peak before overfitting occurs. For both C and gamma, as the values are increased, the training and validation F1 scores increase. The validation scores peak, then decrease, while the training scores continue to increase towards 1. The optimal model had the hyperparameters C = 1, gamma = 0.1, with an accuracy of 0.5648.

Artificial Neural Network

The hyperparameters we tuned for ANN are the number of hidden layers and the learning rate strength. Hidden layers increase the complexity of the model, while learning rate (alpha) balances the steps taken during backpropagation to ensure the optimal model is not overstepped, or never reached. These are plotted below on the x-axis, with F1 scores plotted on the y-axis. The log loss curve is also shown below, which displays the decrease in loss over time.





Similar to the previous two models, the neural network did not perform well on the dataset. Both training and validation scores remained below 0.7. During hidden layer hyperaparameter tuning, increasing the number of hidden layers increased the training accuracy, but decreased the validation accuracy. This shows that overfitting occurred with more hidden layers, which is consistent with a more complex model. Learning rate tuning saw little variation in training and validation scores over time and a large difference between the two, suggesting overfitting. The optimal model had alpha = 0.001, hidden layers = 50, and an accuracy of 0.5905. The log loss curve shows the decrease in loss over time, however, its plateau is not as clear, suggesting that the model may be stopping too early.

Results

For each of the datasets, the optimal hyperparameters were taken and used to train each optimal model. These models were used to evaluate the performance of each model type on the training and testing datasets.

Fake News Dataset

Model	Hyperparameters	Training Accuracy	Testing Accuracy	Training Recall	Testing Recall
Logistic Regression	C = 1000 solver = newton-cg	1.0000	0.9931	0.9999	0.9926
Naive Bayes	alpha = 0.0001 fit priors = False	0.9374	0.9323	0.9308	0.9260
Random Forest	# trees = 50 max depth = None	1.0000	0.9960	0.9999	0.9963

Customer Segmentation Dataset

Model	Hyperparameters	Training Accuracy	Testing Accuracy	Training F1 Score	Testing F1 Score
Random Forest	# trees = 100 max depth = 10	0.6828	0.5434	0.6778	0.5338
Support Vector Machine (RBF Kernel)	C = 1 gamma = 0.1	0.5648	0.5366	0.5582	0.5299
Artificial Neural Network	alpha = 0.001 hidden layers = 50	0.5905	0.5211	0.5872	0.5186

Discussion

Fake News Dataset

Overall, all three models performed exceptionally on the dataset. Logistic regression and random forest, in particular, were the two best models with margin of error differences in scores. Both models had near 100% training accuracies and recalls (training accuracies were <1.000, but rounded to 4 decimal places). Normally, this would be a sign that the model has overfit to the dataset, however the testing accuracy and recall are also both close to 1, as each is just greater than 0.99. Naive Bayes also performed well, with ~ 0.93 accuracy and recall scores for both training and testing splits. While not listed above, it should be noted that logistic regression and random forest both took over one minute to tune hyperparameters, while Naive Bayes took no more than 5 seconds. This aligns with the results above because logistic regression and random forest have more complex structures and therefore provide better models compared to Naive Bayes. We can also conclude that the Fake News dataset is highly learnable. It is likely that there is little to no noise in the distribution with clear separation between each class.

Customer Segmentation Dataset

Contrary to the Fake News dataset, none of the three models performed well on this dataset. The highest scores across the board were from random forest, which had training accuracy = 0.68, testing accuracy = 0.54, training F1 = 0.67, and testing F1 = 0.53. These results are concerning, as the models are barely getting half of the testing samples correct. The training splits performed slightly better for each model, which is expected given possible overfitting, and the testing scores are all barely above 0.5. This is still better than random guessing, however none of the models are “good” models. The main reason why these results have occurred are likely because this dataset has a significant amount of noise or overlap between classes, or that the dataset is not large enough to allow the models to properly learn it. It is also possible

that the models we chose are not well suited for this dataset, as there may be another model we didn't test that would perform well.

Conclusion

This project displayed two extremes in training machine learning models on datasets. We saw an example of a highly learnable dataset, which returned high accuracy and recall scores for all models, and a dataset which is difficult to learn, which produced lower accuracy and F1 scores. This shows that machine learning is not easy and does not produce absolute truths. It has the ability to recognize patterns that will produce repeatable and accurate results that give us insights into particular trends, while its limitations can make it difficult or possible to discern patterns in other areas. It is why the field of machine learning has grown rapidly as people and companies are chasing better models that give more accurate and trustworthy results.

There are certainly areas to improve if more training and analysis were to occur. In the Fake News dataset, the large size caused limitations in hyperparameter tuning. It was unnecessary to spend extra time tuning hyperparameters when the models already performed well, however more thorough tuning may be useful. It would also be useful to test more models on the Customer Segmentation dataset to see if there is a model or models that could learn the dataset with higher accuracy.

References

[1] C. Bisaillon, "Fake and Real News Dataset," Kaggle, 2020. [Online]. Available: https://www.kaggle.com/datasets/clmentbisaillon/fake-and-real-news-dataset. [Accessed: Dec. 13, 2025].

[2] A. Sudarshan, "Customer Segmentation Dataset," Kaggle. [Online]. Available: https://www.kaggle.com/datasets/abisheksudarshan/customer-segmentation. [Accessed: Dec. 13, 2025].