École Polytechnique de Montréal

Département de génie informatique et génie logiciel

INF8245E

Machine Learning

Assignment #3

Submitted by

Sidney Gharib

Lab section #01

November 15th 2024

Model Training and Evaluation

b)

F1-Score per model

|  |  |  |
| --- | --- | --- |
| Model | Train set (F1-Score) | Test set (F1-Score) |
| Decision Tree | 100% | 0.45% |
| Random Forest | 0.99% | 0.49% |
| SVM | 0.28% | 0.26% |

Discussion:  
The F1-score indicates how well the model balances precision and recall, which is particularly important in imbalanced datasets. Here’s a breakdown:

The Decision Tree achieves an F1-score of 100% on the training set, which is expected due to the Decision Tree’s tendency to overfit, especially with high-depth trees. However, the test set F1-score of 0.45% highlights that this overfitting does not generalize well to new data, indicating poor performance in handling unseen examples.

The Random Forest has F1-score of 0.99% on the training set and 0.49% on the test set, the Random Forest shows better generalization compared to the Decision Tree. Random Forests can handle complex patterns while reducing overfitting by averaging predictions across multiple trees, which explains its relatively higher test F1-score.

The SVM model achieves a 0.28% F1-score on the training set and a 0.26% on the test set. This lower F1-score suggests that the SVM might be underfitting, likely due to a lack of tuning for the optimal kernel or regularization parameters.

Accuracy per model

|  |  |  |
| --- | --- | --- |
| Model | Accuracy Train set | Accuracy Test set |
| Decision Tree | 100% | 0.87% |
| Random Forest | 99% | 0.90% |
| SVM | 89% | 89% |

Discussion:

The Decision Tree has an accuracy of 100% on the training set, confirming overfitting. With an 87% accuracy on the test set, the model shows a significant drop, which reflects its inability to generalize well.

The Random Forest achieves 99% accuracy on the training set and 90% on the test set, indicating better generalization than the Decision Tree. Random Forest's ability to reduce variance helps maintain strong accuracy even with unseen data.

SVM: The SVM model has 89% accuracy on both training and test sets, showing consistent performance. However, the lower F1-score suggests it might struggle with minority classes, which is typical when accuracy alone is considered without focusing on the model's precision and recall balance.

d)

Model: Decision Tree

Training Set Classification Report

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Class | | Precision | Recall | F1-Score | | Support |
| 0 | | 1.00 | 1.00 | 1.00 | | 31942 |
| 1 | | 1.00 | 1.00 | 1.00 | | 4226 |
| Accuracy | 1.00 | | | | 36168 | |

Testing Set Classification Report

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Class | | Precision | Recall | F1-Score | | Support |
| 0 | | 0.93 | 0.92 | 0.92 | | 7980 |
| 1 | | 0.44 | 0.47 | 0.46 | | 1063 |
| Accuracy | 0.87 | | | | 9043 | |

Model: Random Forest

Training Set Classification Report

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Class | | Precision | Recall | F1-Score | | Support |
| 0 | | 1.00 | 1.00 | 1.00 | | 31942 |
| 1 | | 1.00 | 1.00 | 1.00 | | 4226 |
| Accuracy | 1.00 | | | | 36168 | |

Testing Set Classification Report

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Class | | Precision | Recall | F1-Score | | Support |
| 0 | | 0.93 | 0.96 | 0.94 | | 7980 |
| 1 | | 0.61 | 0.43 | 0.50 | | 1063 |
| Accuracy | 0.90 | | | | 9043 | |

Model: SVM

Training Set Classification Report

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Class | | Precision | Recall | F1-Score | | Support |
| 0 | | 0.90 | 0.99 | 0.94 | | 31942 |
| 1 | | 0.68 | 0.18 | 0.28 | | 4226 |
| Accuracy | 0.89 | | | | 36168 | |

Testing Set Classification Report

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Class | | Precision | Recall | F1-Score | | Support |
| 0 | | 0.90 | 0.99 | 0.94 | | 7980 |
| 1 | | 0.62 | 0.17 | 0.26 | | 1063 |
| Accuracy | 0.89 | | | | 9043 | |

A chart of a tree for training

Description automatically generated

A chart with a yellow and purple squares

Description automatically generated

A chart of a forest

Description automatically generated with medium confidence

A chart of a forest

Description automatically generated with medium confidence

A chart of a training set

Description automatically generated with medium confidence

A chart of a test set

Description automatically generated with medium confidence

Discussion:

Classification report:

The Decision Tree model performs perfectly on the training set, suggesting overfitting since it has likely memorized the data rather than learning general patterns. He precision, Recall, F1-score for both classes (0 and 1) are perfect at 1.00, as well as the accuracy. The Decision Tree's performance drops significantly on the test set, particularly for class 1 (subscribed), where both precision and recall are quite low (around 0.44 and 0.47, respectively). This further indicates that the model is overfitting, as it struggles to generalize well to new data. It leans towards classifying more samples as non-subscribed, resulting in many false negatives for class 1.

Like the Decision Tree, the Random Forest has essentially memorized the training data, likely due to the ensemble nature of the model capturing multiple features and patterns. However, Random Forests typically generalize better than single decision trees due to their averaging effect. The Random Forest model performs better on the test set than the Decision Tree, especially for class 1 (subscribed), with an increase in precision (0.61) but a relatively low recall (0.41). This suggests that while it’s better at correctly identifying true positives in class 1, it still misses a substantial portion (false negatives), indicating room for improvement in identifying subscribers.

The SVM struggles with class 1 on the training set, as indicated by a low recall (0.18). This implies that the SVM fails to capture important patterns in the subscribed class, potentially due to underfitting. Its focus on maximizing the margin between classes leads to a conservative approach that favors the majority class (class 0), causing many classes 1 samples to be misclassified as class 0. The SVM shows similar performance on the test set as on the training set, with a very low recall for class 1 (0.17). This consistent behavior across training and test sets suggests that the model may be underfitting class 1 and is biased toward class 0. This model struggles to generalize well for minority classes, and its overall performance for identifying subscribers is limited.

Confusion matrix:

The Decision Tree model perfectly classifies all training samples, with no false positives or false negatives. This can often indicate overfitting, especially with decision trees, as they tend to fit the training data very closely. On the test set, we see 629 false positives and 567 false negatives. This drop in accuracy compared to the training set indicates that the model did not generalize as well to unseen data, which is consistent with overfitting. The Decision Tree model may be too complex, capturing noise or specific patterns in the training data that do not generalize well to new data.

The Random Forest model achieves almost perfect classification on the training set, with only a single false negative. This high performance shows that the model is well-fitted to the training data but is less likely to be as overfit as the Decision Tree due to its use of ensemble methods and averaging. The Random Forest model performs better on the test set than the Decision Tree, with fewer false positives and a similar number of false negatives. This indicates better generalization to new data. Random Forest's use of multiple decision trees helps to reduce overfitting, which results in better performance on unseen data.

The SVM model struggles more on the training set compared to the Decision Tree and Random Forest models. It misclassifies 359 negative samples as positives and 3479 positives samples as negatives. This shows that SVM may be underfitting the training data, potentially due to a suboptimal choice of hyperparameters or model complexity.

Overall, we observe that the most common error is a false negative. This can be explained by the fact that the majority of the population is not subscribed, leading the model to classify elements as negative more frequently than positive. This explanation highlights that the model is biased toward the majority class (non-subscribers), resulting in a tendency to miss true positives (subscribers).

Model Tuning

1)

Random-Forest:

max\_depth: It is maximum depth of the tree. A shallow tree (low max\_depth) may underfit the data because it does not capture enough complexity. A deep tree (high max\_depth) may overfit, capturing noise and irrelevant patterns in the training data, leading to poor generalization on unseen data.

min\_samples\_split: The minimum number of samples required to split an internal node. A higher value for min\_samples\_split will make the tree more conservative by splitting nodes less frequently. This prevents the model from overfitting by limiting the depth and complexity of individual trees. A lower value allows more splits, making the tree grow deeper and potentially leading to overfitting.

bootstrap: A boolean parameter that determines whether bootstrap random sampling with replacement is used to generate training data for each tree. If set to False, all data points are used for training each tree (i.e., no bootstrapping). This typically results in a less diverse forest and can lead to overfitting

max\_features: The maximum number of features to consider when looking for the best split at each node. A lower value of max\_features introduces more randomness, resulting in more diverse trees and a better generalization ability of the random forest. A higher value may lead to more accurate individual trees but reduces the randomness, making the model more prone to overfitting.

max\_leaf\_nodes: The maximum number of leaf nodes in each tree. This can be used to limit the growth of trees. Limiting the number of leaf nodes can help prevent overfitting by restricting how complex each tree can get.

SVM:

C: Regularization parameter. The strength of the regularization is inversely proportional to C. A small value for C makes the model simpler (underfitting), while a large value makes the model more complex and fitting to the training data (risk of overfitting)

Kernel: The kernel function defines the decision boundary in the transformed feature space. Here are some values: linear, polynomial, RBF: for non-linear data separability, sigmoid

Degree: This parameter is used only with the **Polynomial** kernel and defines the degree of the polynomial function. A higher degree allows the model to capture more complex patterns.

Gamma: It defines how far the influence of a single training sample reaches. A low gamma means a broader range of influence, while a high gamma means a smaller influence, which might result in overfitting if too high.

Shrinking: A boolean parameter that, if set to True, enables the shrinking heuristic. This heuristic helps to speed up the optimization process by reducing the search space of possible solutions, making the model training faster.

2)

Discussion:

My best performance by accuracy are the following combinations with these parameters

**Decision Tree:**

Criterion: Gini, Max\_depth : 10, Max\_leaf\_nodes: 10, Min\_samples\_leaf: 2

**RandomForest:**

Bootsrap: : True, Max\_depth: 20, n\_estimators' 300

**SVC:**

C: 10, kernel': rbf, Shrinking: True

Picked from:

param\_grid\_decision\_tree = {

'criterion': ['gini','entropy'],

'max\_depth': [10, 20, 50],

'min\_samples\_leaf': [2, 3],

'max\_leaf\_nodes': [5, 10]

}

param\_grid\_random\_forest = {

'n\_estimators': [10, 50, 300],

'max\_depth': [20, 30],

'bootstrap': [True, False],

}

param\_grid\_svm = {

'kernel':['rbf', 'sigmoid'],

'shrinking': [True, False],

'C': [1, 10],

}

3) Cross-validation is essential in machine learning because it provides a more reliable estimate of a model's performance by testing it on multiple subsets of the data. This helps evaluate how well the model generalizes to unseen data, preventing overfitting or underfitting based on a single train-test split. A model that performs well across different folds is less likely to be overfitting. It also reduces bias. Using a single train-test split might give biased estimates of model performance, as the split could be unrepresentative of the overall data. Cross-validation is especially useful during hyperparameter tuning, as it helps assess how different hyperparameter choices affect the model’s performance across multiple data subsets, allowing for more reliable decisions about parameter settings.

Stratified cross-validation is a specific approach in cross-validation where the data is split into folds while maintaining the same distribution of target class labels in each fold. This is particularly important in classification tasks where the classes might be imbalanced. Without stratification, some folds may end up with very few or no samples from the minority class, leading to biased performance estimates. Stratified cross-validation ensures that each fold maintains a similar proportion of class labels as the original dataset, providing more accurate and reliable metrics like precision, recall, and F1-score, especially for minority classes. It also ensures that the model is trained and tested on data that reflects the true distribution of classes in the population.

Report

A screenshot of a test

Description automatically generated

A graph of a line

Description automatically generated with medium confidence

Increasing the value of max\_depth generally leads to higher accuracy because it allows the model to capture more complex patterns in the data. By permitting deeper trees, the model can make finer distinctions and better fit the training data. However, if max\_depth is set too high, the model may overfit the training set, which reduces its ability to generalize to new data.A graph with a line

Description automatically generated

Increasing the n\_estimators (the number of trees in the Random Forest) generally increases accuracy because it enhances the model’s ability to make robust predictions by leveraging ensemble learning principles.

A graph of a line

Description automatically generatedThe choice of kernel affects the model's ability to capture non-linear relationships in the data. The linear kernel assumes a linear relationship, which may oversimplify the data's complexity, resulting in lower accuracy. The polynomial and RBF kernels, however, allow for non-linear decision boundaries, which can better capture complex patterns and improve accuracy.

2)

A graph with a line drawn on it

Description automatically generated

3)

The best model in this comparison is the Random Forest, with a test accuracy of approximately 89.8% Random Forest typically outperforms individual Decision Trees and even Support Vector Machines (SVM) in many cases. Random Forest can capture complex patterns in the data without overfitting as easily as a single Decision Tree might. This ability to generalize well to unseen data often leads to higher test accuracy. SVM can handle non-linear patterns by using kernels, but it still finds a single decision boundary that best separates the classes. This can be limiting if the data is highly complex or if there are many intricate patterns. In contrast, Random Forest combines many decision trees, each capturing different parts of the data. This ensemble approach allows Random Forest to adapt better to non-linear and complex patterns without relying on a single boundary.