# Practical Application Assignment 17.1: Comparing Classifiers

Sudi Narasimhan

**I. Business Understanding** The core objective of this exercise is to best estimate whether a Portuguese bank's marketing campaign that was conducted through phone calls, would result in a client or customer subscribing to a term deposit. This type of AI/ML classification exercise enables enterprise resources to be better utilized such that the right type of customer is more likely to subscribe. Targeting the wrong customer wastes both time, effort, and investment. By employing such AI techniques, enhances corporate productivity and efficiency.

**II Situation Assessment** Resources employed for this exercise include a UC Irvine Machine Learning Repository dataset with 45211 instances, 16 features, and with no missing data. The goal is to compare the performance of the classifiers (k-nearest neighbors, logistic regression, decision trees, and support vector machines), and thereby assess which type of classification model best identifies whether a customer will subscribe to a term deposit. Metrics that will best identify include accuracy, Precision, Recall, F-1-score among others.

## K-Nearest Neighbors

Running the K-Nearest neighbors identified the following:

A screenshot of a computer screen

Description automatically generated

* This indicates that the model has a high percentage of identifying customers accurately. However, accuracy by itself as a datapoint, is insufficient. It has to be considered in context of the other metrics including
* Precision, Recall, and the F-1 score. With Precision, it identifies that when a customer is predicted to commit to a term deposit, 60% actually do. That could correlate to the fact that ~40% are false positives, which means that when the model predicts that a customer commits to a term deposit, ~40% do not.
* The Recall metric has identified 32.09% of True Positives, leaving ~67% as false negatives, indicating that when a customer is predicted to not commit, that they actually do. This is a substantial proportion of missed opportunity whereby the model is not capturing customers who actually commit, despite predicting that they wouldn’t. Marketing resources will not be effectively utilized to target the right customers as they’re not captured through the model.
* An F-1 score, a balance between Precision and Recall, underscores a need for improvement. The outlines a moderate to light ability to identifying true positives and minimizing false positives

A group of black text

Description automatically generated

* Identified the most optimal score of n\_neighbors to be 9 with highest value of .899.

A black text on a white background

Description automatically generated

* A subsequent model run with GridSearch and n\_neighbors revealed:

A screenshot of a computer

Description automatically generated

* Most of the values still hold with prior datapoints. However, ROC AUC indicates that the model is able to distinguish between classes and thereby able to distinguish if a customer is likely to commit or not. Higher ROC AUC values indicates that model performance is able to clearly demarcate different classes.

A yellow and purple squares with numbers

Description automatically generated

* 38% of the customers who were predicted to commit, do not do so.
* Only 30% of the customers who are predicted to commit, do so.
* Train time is barely negligible, and testing time is .29. This will be compared with other models.

## Logistic Regression

A graph with numbers and a number of percentages

Description automatically generated with medium confidence

* The initial data suggests that the model is able to capture 90.67% whether a customer will commit or not, leaning to higher level of performance.
* Precision of 67.74% indicates that when a customer does commit, 67.74% of the time it is correct.
* This does leave 32% of the time as false positives such that when the model predicted that a customer will commit, when in actuality, they don’t
* A Recall of 33.90% indicates that the model captures only 33% of all customers who commit. That leaves a large percentage of customers who are potentially false negatives whereby the model does not identify those customers who were predicted not to commit, but actually do.
* The F-1 score, a balance between precision and recall outlines the need for significant improvement.
* ROC AUC of ~88.87% indicates that the model does 90% accurately distinguish between classes.

A screenshot of a computer code

Description automatically generated

A screenshot of a computer code

Description automatically generated

* Identified that C with .1 was most optimal.

A screenshot of a computer

Description automatically generated

* Training time is .05 and testing time is 0.00
* In comparison to K-Nearest neighbors, the model is more accurate with less compute training time, and testing time

A yellow and purple squares with numbers

Description automatically generated

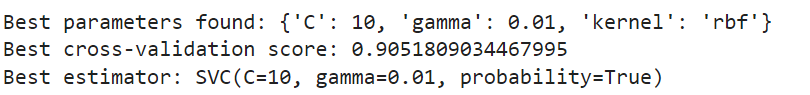
* 7161 customers are True Negatives, those who were predicted to not to commit and didn’t do so
* 142 as False Positives – Those customers who were thought to commit but didn’t do so
* 311 customers as True Positives – Those customers who were predicted to commit and did so
* 624 as False Negatives – Those customers who were not expected to commit but did so anyway. Essentially, the model failed to capture a class whereby they may have been incorrectly targeted

## Support Vector Machine

**A screenshot of a graph

Description automatically generated**

* The model captures ~91% of the time whether a customer will commit to a term deposit or not
* Precision of 68.39% indicates that the model is able to capture 68% of the time when a customer is predicted to commit, he or she actually does so
* Recall of 29.45 reveals that close to 80% of false negatives such that the model fails to capture a significant portion of actual positive cases whereby a customer who was not predicted to commit, actually does so. This leads to a large issue around missed opportunities
* F-1 score of the balance between Precision and Recall of 41.17% is comparably low, leaving significant room for improvement.
* ROC of AUC 84% is clearly able to distinguish the classes and good at ranking customers who are likelihood of committing.
* **Note**: Due to the high cost of computing and training with Support Vector Machine (SVM), a sample of the actual data had to be used. An attempt to train with the actual dataset took more than 12+ hours.

****

* Based on GridSearchCV Hyperparameter tuning the optimal parameters were identified.

**A screenshot of a computer

Description automatically generated**

* Training and Test time were higher for SVM then any other of the other models
* With the final results below:

**A screenshot of a graph

Description automatically generated**

* Accuracy indicates that the model is able to capture with 90.54% accuracy whether a customer will commit or not.
* Precision of 68% indicates that when a customer is predicted to commit, 68% of the time, the model is accurate
* Recall of 30% indicates a large number of false negatives, essentially identifying that 70% of the population who were not predicted to commit, actually did so indicating a lost opportunity.
* F1 Score of 41 establishes the balance between Precision and Recall with a low performance score
* ROC AUC of 84% indicates that the model is able to distinguish quite strongly of different classes and good at ranking customers by their proclivity to commit

A chart of a number of colored squares

Description automatically generated with medium confidence

* 10779 are True Negatives – Model correctly identified those who were predicted to not commit and didn’t do so.
* 189 are False Positives – Model identified 189 of the observations who were predicted to commit but didn’t do so
* 409 are Tue Positives – those who were predicted to commit and did so
* 980 are True Negatives - Those who were predicted not to commit but did do so, leaving missed opportunity as the model failed to accurately capture proportion who had the proclivity to commit to a term deposit

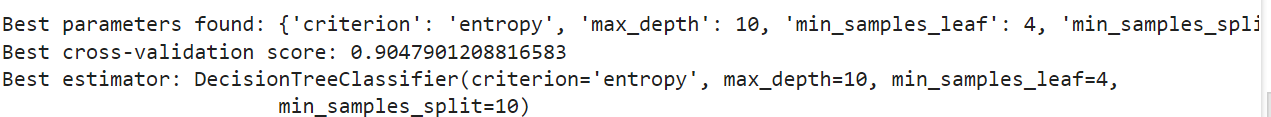
## Decision Trees

A screenshot of a computer screen

Description automatically generated

* Accuracy indicates that 87% of the time the model is able to capture whether a customer will commit or not 87% of the time.
* Precision of 45% indicates that when the model commits to predicting whether a customer will commit, it’s accurate 45% of the time, leaving about 55% are false positives, a population who were predicted to commit to a term deposit, but didn’t do so.
* Recall of 45% identifies those who correctly committed but leaves 55% as false negatives, a population who were predicted not to commit but did so anyway.
* F-1 Score of 45% indicates average model performance reflecting a balance between Precision and Recall
* Although not as strong as the other models, an ROC AUC of 69% indicates that it is not as strong as the other models in identifying and distinguishing classes and less effective in ranking customers who are likely to commit.

With GridSearchCV Hyperparameter tuning the results as below:



A screenshot of a computer

Description automatically generated

A screenshot of a computer screen

Description automatically generated

* The accuracy has improved as a result of the optimized hyperparameters through GridSearchCV.
* Precision has increased to 62.30% from 44.58% indicating that when a model predicts that a customer will commit, it’s correct 62% of the time. The false positives whereby a customer is predicted to commit but doesn’t do so is reduced to 38% of the time.
* Recall has reduced to 37% but that leaves a substantial number of false negatives whereby approximately 63% of the time when the model predicts that they will not commit, they still do so, leaving a gap where the model does not capture actual commits.
* F-1 Score has improved to 46% with a better balance between Precision and Recall.
* ROC-AUC has improved to 88.17% indicating a strong ability for the model to distinguish between classes, ranking customers more effectively in their likelihood to commit.
* Training time is .31 but the testing time is negligible.
* A chart of a comparison of a number and a label

  Description automatically generated with medium confidence
* 10657 are True Negatives whereby customers correctly predicted as not committing to a term deposit and did not
* 311 are False Positives such that model predicted that they would commit when they did not
* 514 are True Positives such that the model predicted that they would commit and did do so
* 875 are False Negatives such that the model predicted that they would not commit but did do so discounting actual commits.

## Model Comparison

* Logistic Regression does better (90.67%) than the Decision Tree of 90.40% and SVM of 90.54% in terms of model accuracy
* With Precision, of SVM 68.39% and Logistic Regression 67.74% have the highest precision, meaning that they are more reliable in predicting committers correctly.
* Decision Tree (37.01%) and Logistic Regression (33.90%) show better recall compared to SVM (29.45%) and k-NN (32.09%). Recall indicates how well the model captures actual committers.
* Logistic Regression (88.87%) and Decision Tree (88.17%) have the highest ROC AUC, indicating strong performance in distinguishing those who commit and those who don’t.
* With Training Time under consideration Logistic Regression is likely the best performance with the balance of precision, recall and F-1 score.