Supervised Learning Assignment

Abtin Mahyar

Abstract

I did a classification task with different classifiers on 'TalkingData adtracking fraud detection' dataset which is provided on Kaggle. After that, classification report for each classifier is provided to select the best classifier for this task.

1. Overview

In this part, I preform different classification methods in order to predict whether an app will be downloaded after clicking on a mobile app advertisement using 'TalkingData adtracking fraud detection' dataset. At the end, classification report, ROC curve, and decision boundaries for each classifier will be reported.

2. Data Summary

Dataset mentioned above, has valuable information about user information such as their IP, and devices, and information about advertisement such as applications that users used. This dataset has around 185 million records which each record refers to the click action of a particular user in an exact time. The dataset has records for only four days from 6 November 2017 to 9 November 2017; As a result, features like 'year' and 'month' are not useful in our prediction (constant data) and should be dropped in preprocessing stage. Description of different features which were available in the dataset can be found on Kaggle. Since the original dataset is too large and needs lots of resources and time for its' calculations, I used a sampled dataset from the original one which is much smaller and has 100,000 records which is suitable for our purpose.

2. Metrics

- Accuracy is a common metric for binary classifiers; it takes into account both true positives and true negatives with equal weight.

$$Accuracy = \frac{True\ positives + True\ negatives}{dataset\ size}$$

- Confusion matrix is also a metric that can delivers valuable information from models' predictions which returns a matrix in a form of following.

$$\begin{pmatrix} True\ negatives & False\ negatives \\ False\ positives & True\ positives \end{pmatrix}$$

- Area Under Curve (AUC) is one of the most widely used metric for evaluation. It is used for binary classification problem. *AUC* of a classifier is equal to the probability that the classifier will rank a randomly chosen positive example higher than a randomly chosen negative example. AUC is the area under curve of plot False Positive Rate vs True Positive Rate.

$$TruePositiveRate = \frac{TP}{FN + TP}$$
$$TrueNegativeRate = \frac{TN}{TN + FP}$$
$$FalsePositiveRate = \frac{FP}{TN + FP}$$

- Precision is the number of correct positive results divided by the number of positive results predicted by the classifier.

$$Precision = \frac{TP}{TP + FP}$$

- Recall is the number of correct positive results divided by the number of all relevant samples (all samples that should have been identified as positive).

$$Recall = \frac{TP}{TP + FP}$$

- F1 Score is the harmonic mean between precision and recall. Mathematically, it can be expressed as follows:

$$F1 = \frac{1}{\frac{1}{precision} + \frac{1}{recall}}$$

4. Methodology

4.1. Data Preprocessing

The preprocessing done in the notebook consist of the following steps:

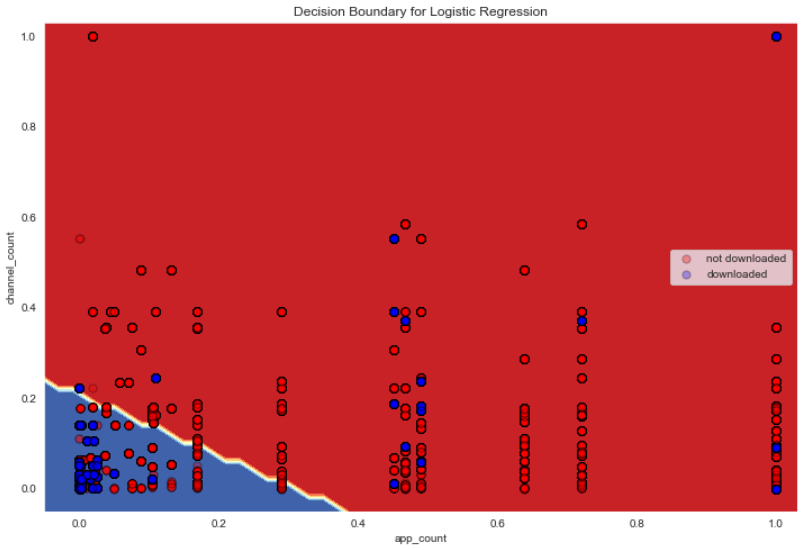
1. New features are extracted from the 'click_time' feature in the dataset such as 'day', 'hour', 'min', and 'sec'.
2. All other features in the dataset were categorical with high number of unique values as categories; therefore, they were encoded with count encoder and added to the dataset.

3. Some of unrelated, unvaluable, and redundant columns such as ‘attributed_time’ dropped. resulted dataset has 9 columns.
4. Normalization performed on the dataset using a min-max scaler.
5. Since dataset was imbalanced (only 0.002% of target values were positive), different samples from original dataset were created using down-sampling, oversampling, and SMOTE methods to increase performance of trained models.
6. Dataset split to two train and test sets for evaluating and comparing different classifications methods. (80% of data for train phase and 20% for testing).

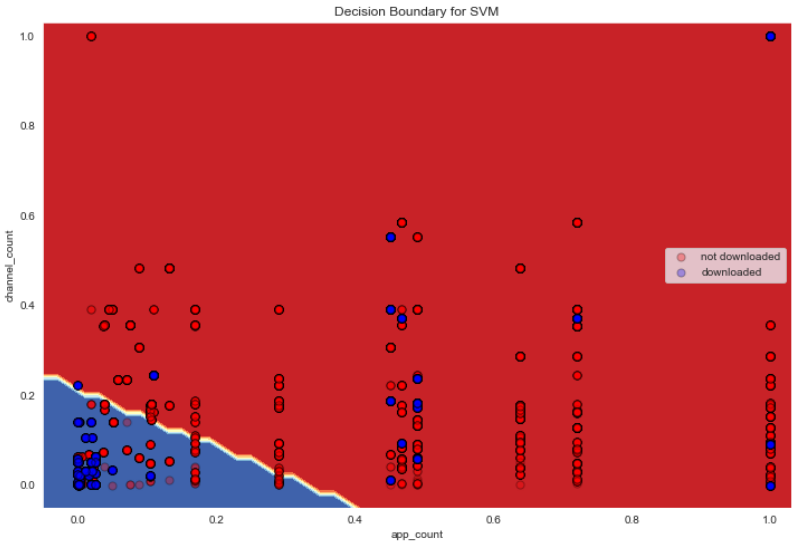
4.2. Algorithms and Techniques

1. **Logistic Regression**

   The classifier is a simple logistic regression model, which is used to model the probability of a certain target class. Also, Hyperparameters such as penalty and penalty strength for this model are tuned using grid search cross validation. Best estimator for this optimization was a logistic regression model with penalty = ‘L2’ and penalty strength = 100, and this model was trained with oversampled dataset. Following plot demonstrates decision boundary for this model.



* For visualizing decision boundaries, as we have more than two features in our dataset, just two most important features for classification were selected with respect to each model.

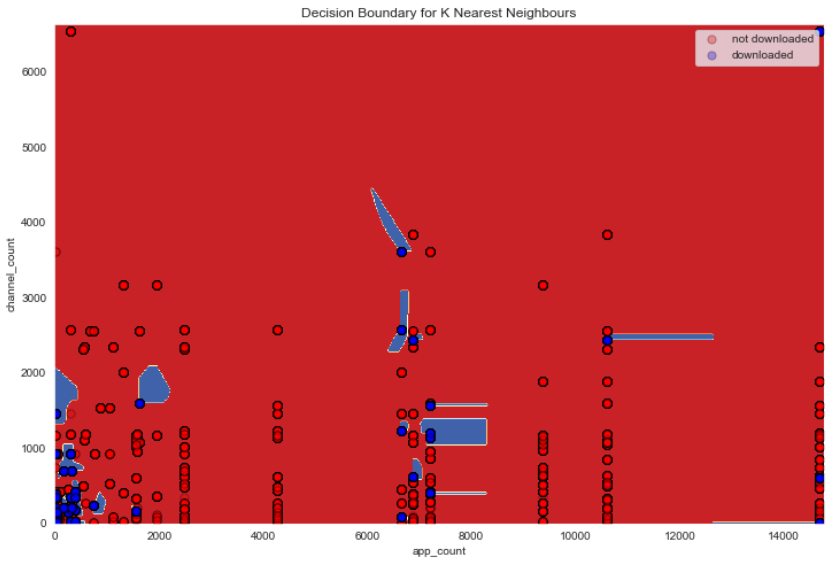


2. **Support Vector Machine**

This classifier maps training examples to points in space so as to maximize the width of the gap between two categories which makes it a non-probabilistic binary linear classifier. I set a linear kernel for this model, and it was trained with sampled dataset using SMOTETomek technique; However, since SVM is a slow supervised learning model, I made up a sample with 10% data from that dataset in order to save time and resources. Following plot demonstrates decision boundary for this model.

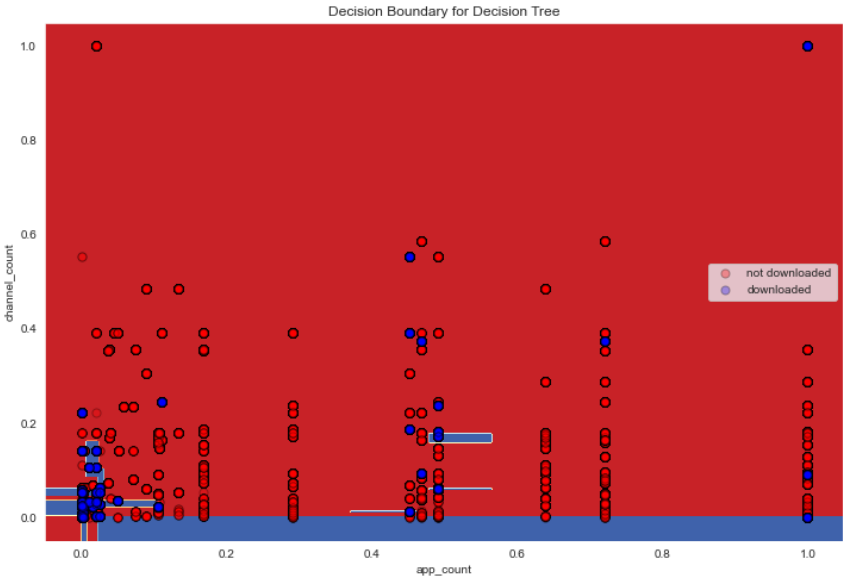
3. **K-Nearest Neighbor**

This classifier is a non-parametric model which assign a label to an unlabeled vector (a query or test point) with respect to labels form k nearest neighbors to that vector. Also, Hyperparameters such as number of neighbors, weights, and metric for this model are tuned using grid search cross validation. Best estimator for this optimization was a model with weights = 'distance', number of neighbors = 3, and metric = 'euclidean', and this model was trained with sampled dataset using SMOTETomek technique. Following plot demonstrates decision boundary for this model.



4. **Decision Tree**

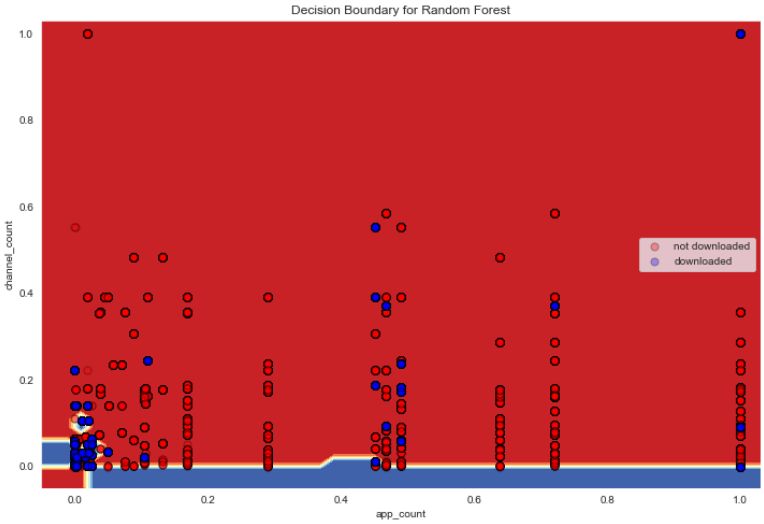
This classifier is a non-parametric tree-like model of decision of possible consequences. Also, Hyperparameters such as split function (criterion), maximum depth of the tree, minimum number of samples required to split, and minimum number of samples required to be at a leaf node for this model are tuned using grid search cross validation. Best estimator for this optimization was a decision tree model with criterion = 'entropy', max depth = 9, min sample split = 9, and min sample in a leaf node = 1, and this model was trained with sampled dataset using SMOTETomek technique. Following plot demonstrates



decision boundary for this model.

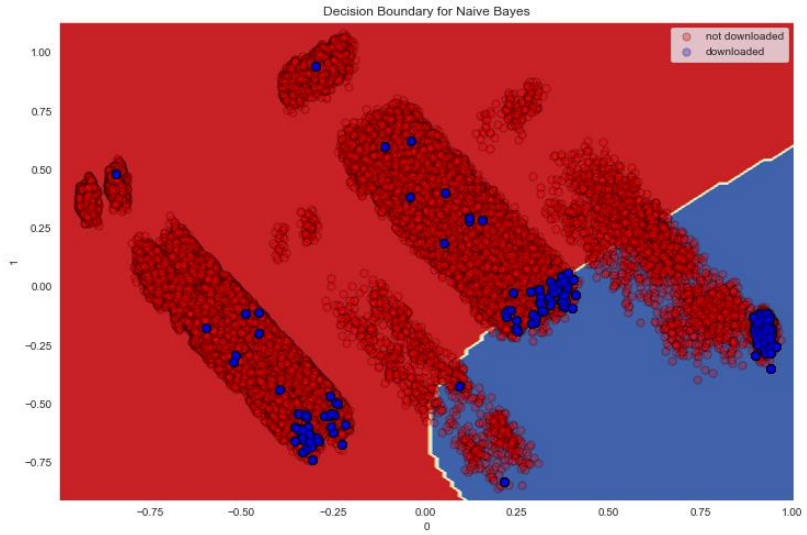
5. **Random Forest**

This classifier is an ensemble learning method which operates by constructing a multitude of decision trees at training time. Also, Hyperparameters such as number of estimators, and maximum number of features to consider for this model are tuned using grid search cross validation. Best estimator for this optimization was a random forest model with criterion = 'entropy', number of estimators = 10, maximum feature method = 'sqrt', and this model was trained with sampled dataset using SMOTETomek technique. Following plot demonstrates decision boundary for this model.



6. **Naive Bayes**

This model is a probabilistic classifier based on applying Bayes' theorem with strong independence assumption between features. Gaussian naïve bayes has the best performance compare to the other models, and this model was trained with oversampled dataset. Following plot demonstrates decision boundary for this model.



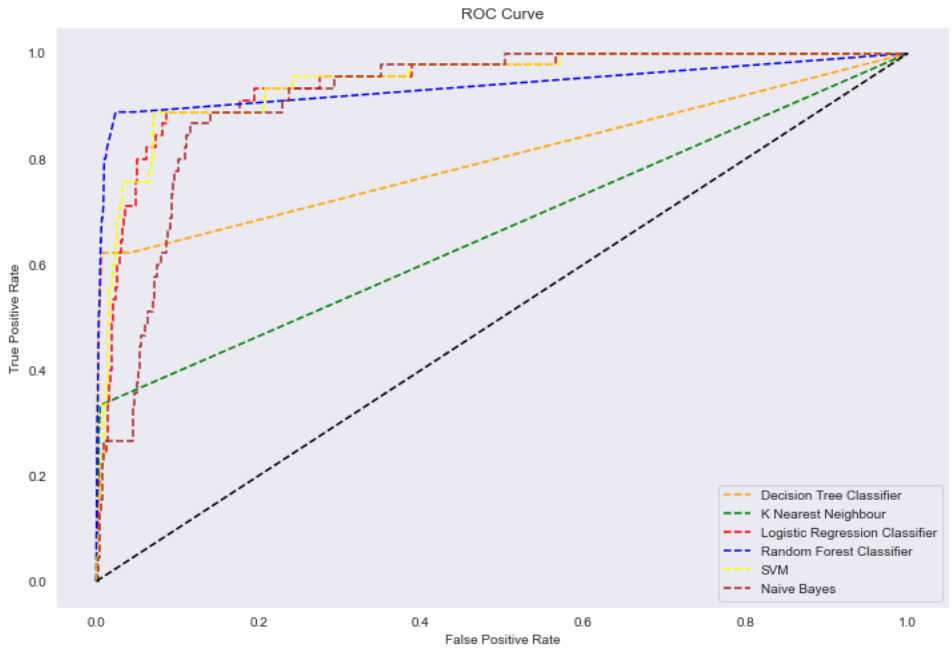
* Since naïve bayes classifier do not offer an intrinsic method to evaluate feature importance, I perform PCA on dataset in order to plot decision boundary for this model.

4.3. Models Evaluation and Comparison

During development, a test set was used to evaluate the model. Full description of models' performances which trained with the dataset is listed as follows.

Test dataset has 19,955 records labeled to class '0', and 45 records labeled to class '1'. Confusion matrix is in form of $\begin{pmatrix} TN & FP \\ FN & TP \end{pmatrix}$. Since dataset is imbalanced, for calculating precision, recall, F1-score, and accuracy, macro average method is used.

Model	Precision	Recall	F1-Score	Accuracy	Confusion Matrix	AUC
Logistic Regression	0.51	0.88	0.5	0.92	183637 159238	0.9425
SVM	0.51	0.9	0.5	0.916	182965 165940	0.9448
KNN	0.58	0.64	0.6	0.995	198903 2 65 13	0.6644
Decision Tree	0.53	0.8	0.55	0.977	19528817 42728	0.8019
Random Forest	0.7	0.62	0.65	0.997	199393 4 16 11	0.8627
Naïve Bayes	0.51	0.87	0.48	0.87	173766 257939	0.9146



5. References

1. https://towardsdatascience.com/metrics-to-evaluate-your-machine-learning-algorithm-f10ba6e38234
2. https://scikit-learn.org/stable/modules/classes.html#module-sklearn.metrics
3. https://machinelearningmastery.com/hyperparameters-for-classification-machine-learning-algorithms/
4. https://www.youtube.com/watch?v=pDw_JHHvj-0
5. https://scikit-learn.org/stable/modules/classes.html#module-sklearn.model_selection
6. https://scikit-learn.org/stable/modules/grid_search.html
7. https://en.wikipedia.org/wiki/Receiver_operating_characteristic
8. https://inblog.in/Feature-Importance-in-Naive-Bayes-Classifiers-5qob5d5sFW
9. https://scikit-learn.org/stable/modules/naive_bayes.html
10. https://en.wikipedia.org/wiki/Hyperparameter_optimization
11. https://scipython.com/blog/plotting-the-decision-boundary-of-a-logistic-regression-model/
12. https://machinelearningmastery.com/feature-selection-with-real-and-categorical-data/
13. https://hackernoon.com/how-to-plot-a-decision-boundary-for-machine-learning-algorithms-in-python-3o1n3w07
14. https://stackoverflow.com/questions/42916137/why-does-my-roc-curve-look-like-a-v
15. https://www.datacamp.com/community/tutorials/svm-classification-scikit-learn-python