Practical Assignment

Program: Erasmus Mundus Master COSI

Course: Data Science

Student: Dong Han

Problem description

In this practical assignment, our goal is to analyze Online Shoppers Purchasing Intention problem. In specific, the predictive model would be developed for the purchasing intention of an online store. The dataset consists of 10,000 instances belonging to two classes: False, if the user did not commit the buy; True, if the user finally bought something. There are 17 attributes for each instance. The class is the last one, "Revenue".

Dataset analyzation

There are 18 parameters in the dataset. There are both numerical data and categorical data. The standard deviations of the data can be visualized below:

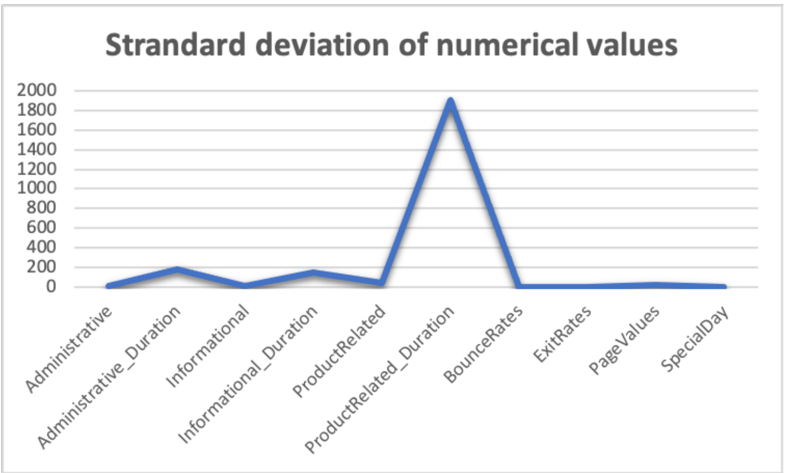


Figure 1. Standard deviations of numerical values

There are 8 categorical variables named operating system, browser, region, traffic type, visitor type, weekend, month, revenue. These describe some of the other details of the browsing history. The revenue shows if a customer bought a product. The number of categories in each of the fields can be found in Figure 2.

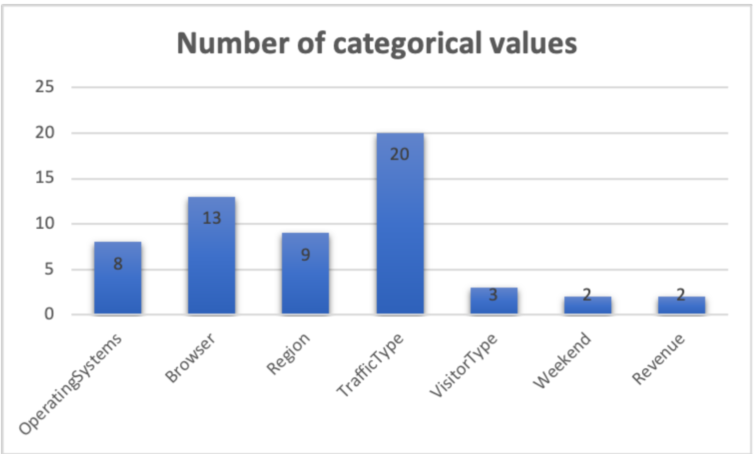


Figure 2. Categorical variables

Another thing is to check if the dataset is balanced. We are interested in the dataset distribution based on revenue class.

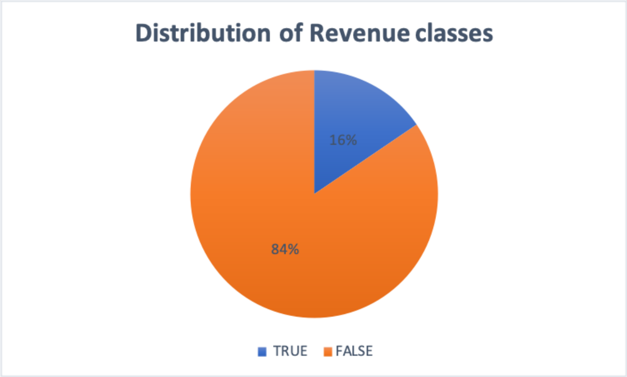


Figure 3. Distribution of revenue classes

It can be seen that only 16% of the positive samples in the dataset, and the remaining 84% belong to the negative category. The data imbalance is extremely serious. The imbalanced datasets may cause problems when creating classification models.

A binary classification model that always predicts the majority class may have a high associated accuracy value (correctly classified samples with respect to the total number of classification tests) while probably misclassifying most of the samples of the minority class. Moreover, in many situations, the minority class is the most important one. Therefore, we need to deal with this problem during the data preprocessing part.

Data preprocessing

Step 1: The class column is named "Revenue" and has two logical classes: False and True. In order to use the dataset for our prediction model, we converted this logical data type to factor. In the end, this created factor has two levels: True and False.

Step 2: In the categorical data, there are 3 parameters (Month, VisitorType, Weekend) with logical or factor data type. We processed the dataset by replacing all the categorical values of those 3 parameters with numerical values.

Step 3: We divided our dataset into two parts by random sampling, the 75% for training and the 25% for testing.

Step 4: Sampling dataset to make it less imbalanced. There is one thing that should be kept in mind, the sampling should always be done on a training dataset. Therefore, for the training dataset, we performed over-sampling with the replacement on the minority samples and under-sampling without replacement for majority samples.

This is general data preprocessing for all the classification models used later.

Methodology

Notice: for all predictive models we keep the predictive formula the same (Revenue ~.) to compare different model's performance.

1. KNN (mknn)

The core idea of KNN is to find the k points closest to the point to be classified in the feature space. If most of these k points belong to a certain category, the sample also belongs to this category.

2. Linear Discriminant Analysis (mlda)

The purpose of LDA is to perform dimensionality reduction processing, so as to retain as much classification information as possible, so we need to find the best projection direction to separate the data points as much as possible. The criterion for separation is that data of the same kind should be as close as possible and data of different kinds should be as far away as possible.

In mlda model, we simply using the lda() function.

3. Logistic Regression (mlr)

The essence of logistic regression is to assume that the data obey this distribution, and then use maximum likelihood estimation to estimate the parameters. We manually define the continuous predicted value, with one side of the boundary defined as 1, and the other side as 0. In this way, we convert the regression problem into a classification problem.

In mlr model, we specific family parameter is binomial in glm() function. In predict() function, we set type as a response since this case, our prediction is in form of probability not log-odds. For output of predict() function, we set all the values larger than 0.5 as “TRUE” and the remained part is set to “FALSE”

4. Classification trees (mtree)

The classification tree splits the response variable into mainly two classes *Yes* or *No*, also can be numerically categorized as 1 or 0. The classification decision trees are built with unordered values with dependent variables. The observation is assigned to the most commonly occurring class.

In the mtree model, we use the normal tree() function and also prune.tree() function to prune the tree. The parameter best allows us to select a subtree. In this case, we test different configurations with respect to number of subtrees. We get the best performance when the best is equal to 7. It gives us the same result as tree() function. Therefore, for less complexity, we can leave out prune.tree() only using tree().

5. Random Forests (mrf)

Random forest is composed of many decision trees, and there is no correlation between different decision trees. When we perform classification tasks, new input samples are entered, and each decision tree in the forest is judged and classified separately. Each decision tree will get its classification result. The most classified result of the decision tree is picked as the final result of random forest.

6. Artificial Neural Networks (mann)

Artificial Neural Network (ANN) is a network of groups of small processing units that are modeled based on the behavior of human neural networks. It follows the non-linear path and processes information in parallel throughout the nodes.

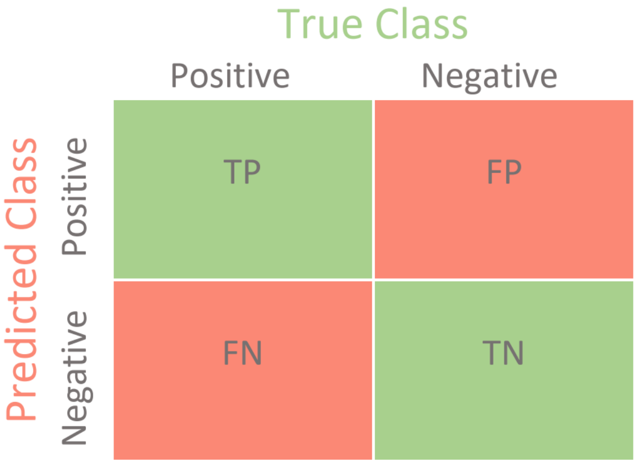
7. Support Vector Machines (msvm)

The basic model of Support Vector Machine (SVM) is to find the best separation hyperplane in the feature space to maximize the interval between positive and negative samples on the training set. We used tune() to compute best model

Result analysis

There are four metrics based on confuse matrix are used here to evaluate the classification results from different algorithms.

A confusion matrix is a NxN matrix that is usually used to describe the performance of a classification model.



		True Class	
		Positive	Negative
Predicted Class	Positive	TP	FP
	Negative	FN	TN

Figure 4. Confusion matrix

Accuracy: Measures the percentage of correctly classified observations.

Accuracy = (TP + TN)/(TP+TN+FP+FN)

Precision: A measure of the accuracy in the classification for positive samples, the proportion of observations marked as positive that are correctly classified.

$$\text{Precision} = TP / (TP + FP)$$

Recall: The rate at which all actual positive samples are correctly classified. Also known as Sensitivity.

$$\text{Recall} = TP / (TP + FN)$$

F-score: Combine accuracy and recall as a measure of classification effectiveness. The specific formula is as follows (ß often takes 1):

$$\text{F measure} = ((1 + \beta)^2 \times \text{Recall} \times \text{Precision}) / (\beta^2 \times \text{Recall} + \text{Precision})$$

We also plotted the ROC curve and calculated Area under the curve (AUC) values. ROC is a probability curve. AUC is a probability that the classifier rank a randomly chosen positive instance higher than a randomly chosen negative one. It indicates the ability of model to distinguish between classes. Generally, an AUC of 0.5 suggests no discrimination, 0.7 to 0.8 is considered acceptable, 0.8 to 0.9 is considered excellent, and more than 0.9 is considered outstanding.

All the results show below are according to the testing part.

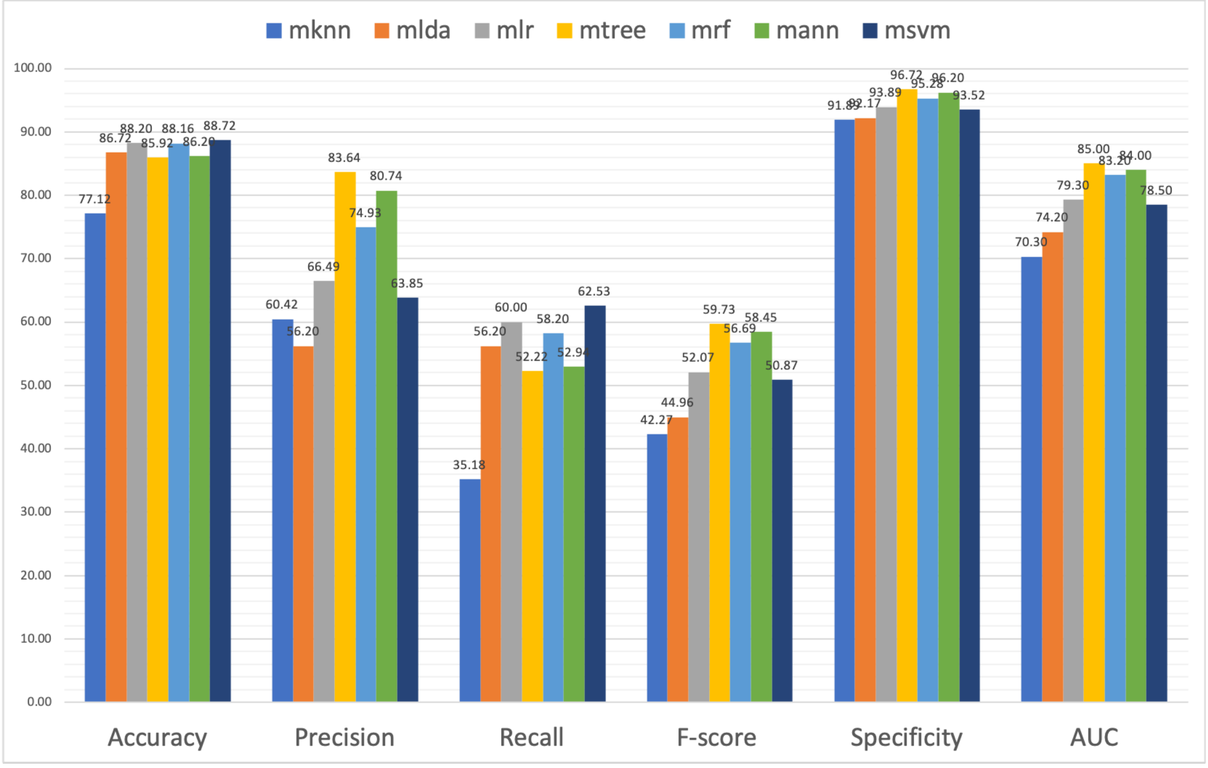


Figure 5. Evaluation of different classification algorithms, the values show in percentage

Algorithm	Accuracy (%)
K Nearest Neighbors	77.12
Linear Discriminant Analysis	86.72
Logistic Regression	88.20
Classification Trees	85.92
Random Forests	88.16
Artificial Neural Networks	86.20
Support Vector Machines	88.72

Table 1. Accuracy of different classification methods

We have calculated different performance metrics for 7 classification algorithm: K-nearest neighbors, Linear discriminant analysis, Logistic regression, Classification trees, Random forests, Artificial neural networks, Support vector machines. The results are shown in Figure 5. In comparison among all the metrics, Classification trees give us the best performance. Although the accuracy of Classification trees is not best for the dataset, it shows relative high performance in precision, F-score, specificity and AUC. The high precision indicates the Classification trees has good repeatability and Reproducibility. And from the F-score we see that Classification trees has the best balance between precision and recall. According to specificity, the Classification trees works better to find the customers who don’t yield revenue. High AUC tell us the Classification trees has a good measure of separability. In this case, the recall is not very high because we balanced our training dataset by using sampling.

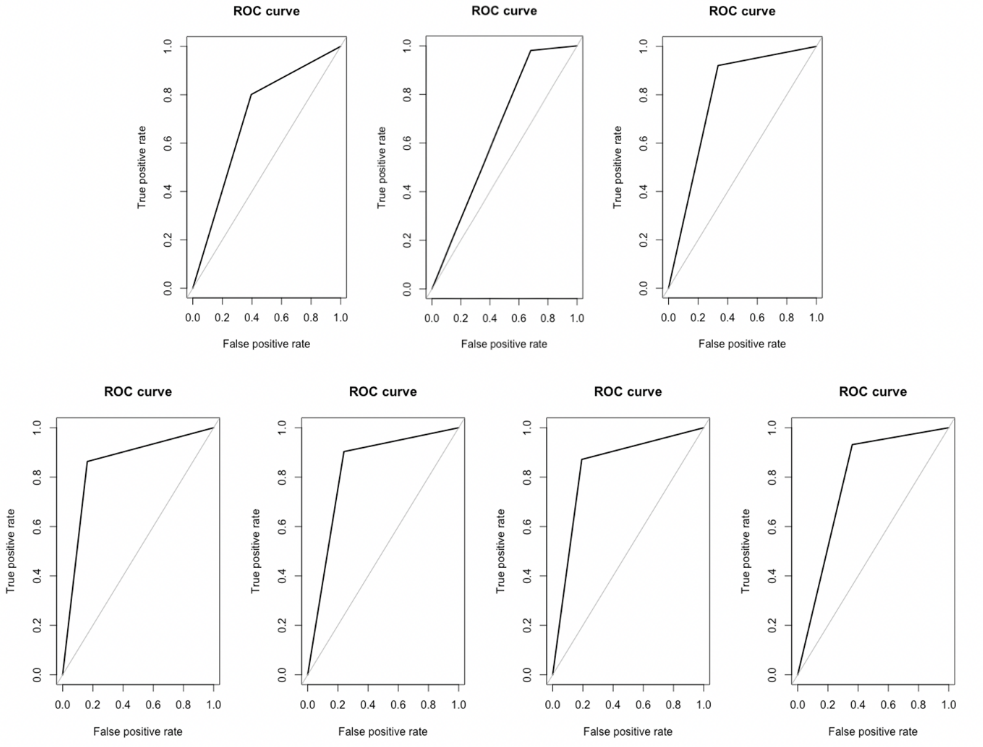


Figure 6. ROC curve of different classification algorithms, from left to right and from upper to bottom, mknn, mlda, mlr, mtree, mrf, mann and msvm model, respectively

Algorithm	AUC (%)
K Nearest Neighbors	70.30
Linear Discriminant Analysis	74.20
Logistic Regression	79.30
Classification Trees	**85.00**
Random Forests	83.20
Artificial Neural Networks	84.00
Support Vector Machines	78.50

Table 2. Accuracy of different classification methods

From the ROC curve and AUC, we can see the Classification trees, Random forests, Artificial neural networks have the excellent performance to distinguish between the positive and negative classes and again the Classification trees are the best.

In the end, we can have a look of mtrees model since it is the best classification model for the Revenue dataset in this case.

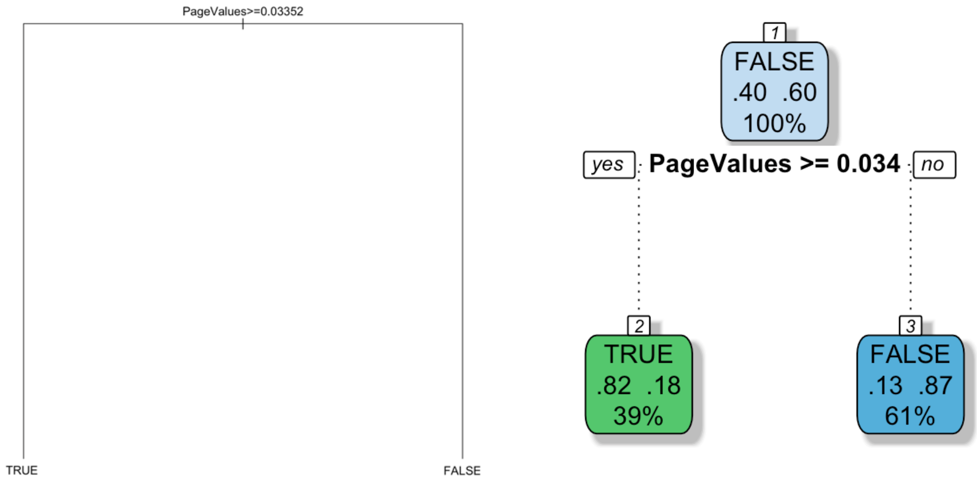


Figure 7. The tree structure of classification trees algorithm, normal plotting on left, rpart plotting on right

We can see the classification trees algorithm gives us very simple tree to handle this binary classification problem. The tree only takes “PageValues” parameter to make a prediction. The tree structure shows us, we can make a good prediction about “Revenue” with “PageValues”. The "Page Value" feature represents the average value for a web page that a user visited before completing an e-commerce transaction. Each node shows us the predicted class (FALSE or TRUE), the predicted probability of buying and the percentage of observations in the node.

Conclusions

In this assignment, we used 7 different classification methods from the conventional to the most advanced approaches. The same dataset including training and testing are used during the experiment. The different configurations in the same method are implemented to select the best one and the final best model is computed by comparing among all the different classification methods based on several classical metrics. In the final, the Classification tree algorithm shows best performance. It is relatively accurate and precise with a good trade-off between them. There is also a good balance between precision and recall. Moreover, the values of specificity and AUC are highest.