Supervised Machine Learning Techniques

(Dataset: Online Shoppers Purchasing Intention)

Prepared by: Gözde Işıldak,Ferhat Kortak, Seray Esen

Date Due: 20 May 2019

Table of Contents

1. INTRODUCTION

The purpose of the report is explaining how machine learning techniques applied to the dataset. Supervised and unsupervised machine learning algorithms used to make prediction for target feature. There are main categories in machine learning algorithm approaches; information based, similarity based, probability based, error based algorithms and all type of algorithms applied to the dataset.

1.1 Getting Familiar into Dataset

The dataset consists of 10 numerical and 8 categorical attributes. Raw form of the dataset has 12.331 instances. Target value is “revenue” and it has two separate values as TRUE and FALSE. Other features are explained below.

"Administrative", "Administrative Duration", "Informational", "Informational Duration", "Product Related" and ***"Product Related Duration"*** represent the number of different types of pages visited by the visitor in that session and total time spent in each of these page categories. The values of these features are derived from the URL information of the pages visited by the user and updated in real time when a user takes an action, e.g. moving from one page to another. The ***"Bounce Rate", "Exit Rate"*** and ***"Page Value"*** features represent the metrics measured by ***"Google Analytics"*** for each page in the e-commerce site. The value of ***"Bounce Rate"*** feature for a web page refers to the percentage of visitors who enter the site from that page and then leave ***("bounce")*** without triggering any other requests to the analytics server during that session. The value of ***"Exit Rate"*** feature for a specific web page is calculated as for all page views to the page, the percentage that were the last in the session. The ***"Page Value"*** feature represents the average value for a web page that a user visited before completing an e-commerce transaction. The ***"Special Day"*** feature indicates the **closeness** of the site visiting time to a specific special day *(e.g. Mother’s Day, Valentine's Day)* in which the sessions are more likely to be finalized with transaction. The value of this attribute is determined by considering the dynamics of e-commerce such as the duration between the order date and delivery date. For example, for Valentine’s day, this value takes a nonzero value between February 2 and February 12, zero before and after this date unless it is close to another special day, and its maximum value of 1 on February 8. The dataset also includes ***operating system, browser, region, traffic type, visitor type as returning or new visitor***, a Boolean value indicating whether the date of the visit is **weekend**, and **month of the year.**

1.2 Exploratory Data Analysis

Before cleaning the data phase, the raw dataset has to be examined. Firstly, you can see that in the plot below how many missing values for each descriptive features.

```
# Load csv dataset from working directory
shopper_df <-read.table(file="online_shoppers_intention.csv",
header=TRUE, sep=",", dec = ".")
# Print count of missing values for each feature
sapply(shopper_df, function(x) sum(is.na(x)))
```

```
> sapply(shopper_df, function(x) sum(is.na(x)))
        Administrative Administrative_Duration          Informational  Informational_Duration          ProductRelated ProductRelated_Duration
                     0                       0                      0                       0                       0                       0
            BounceRates              ExitRates             PageValues              SpecialDay                   Month         OperatingSystems
                     0                       0                      0                       0                       0                       0
               Browser                  Region            TrafficType             VisitorType                 Weekend                 Revenue
                     0                       0                      0                       0                       0                       0
```

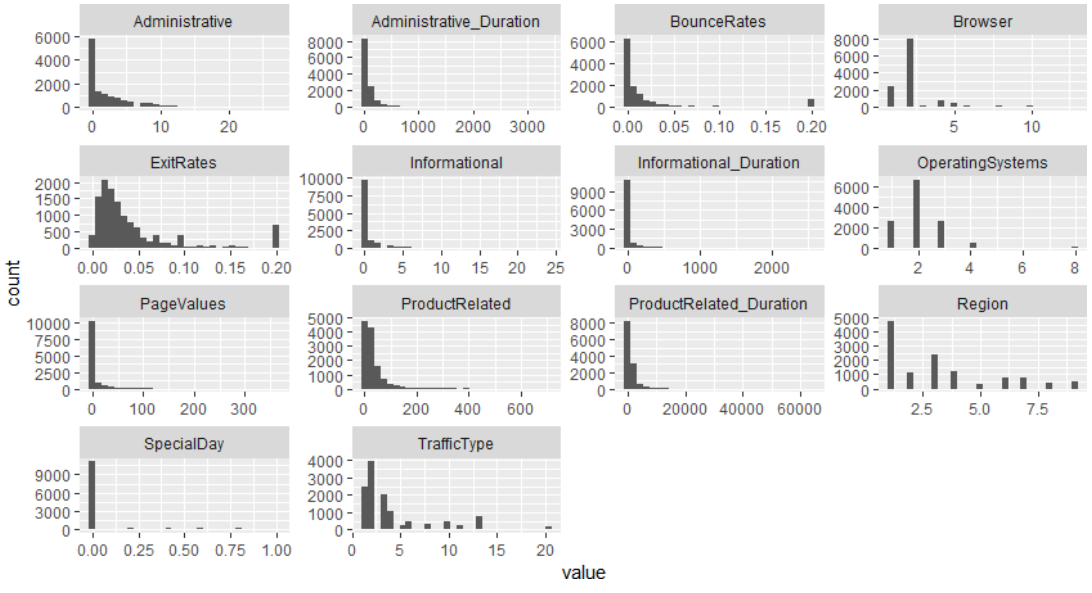
There is no missing values of the raw dataset can be observed in this short summary. In the second step, frequency distribution for each descriptive features are plotted in order to proper feature selection.

1.2.1 Data Visualization

In order to getting intuitive ideas about dataset, features should be plotted. This section includes many types of plots.

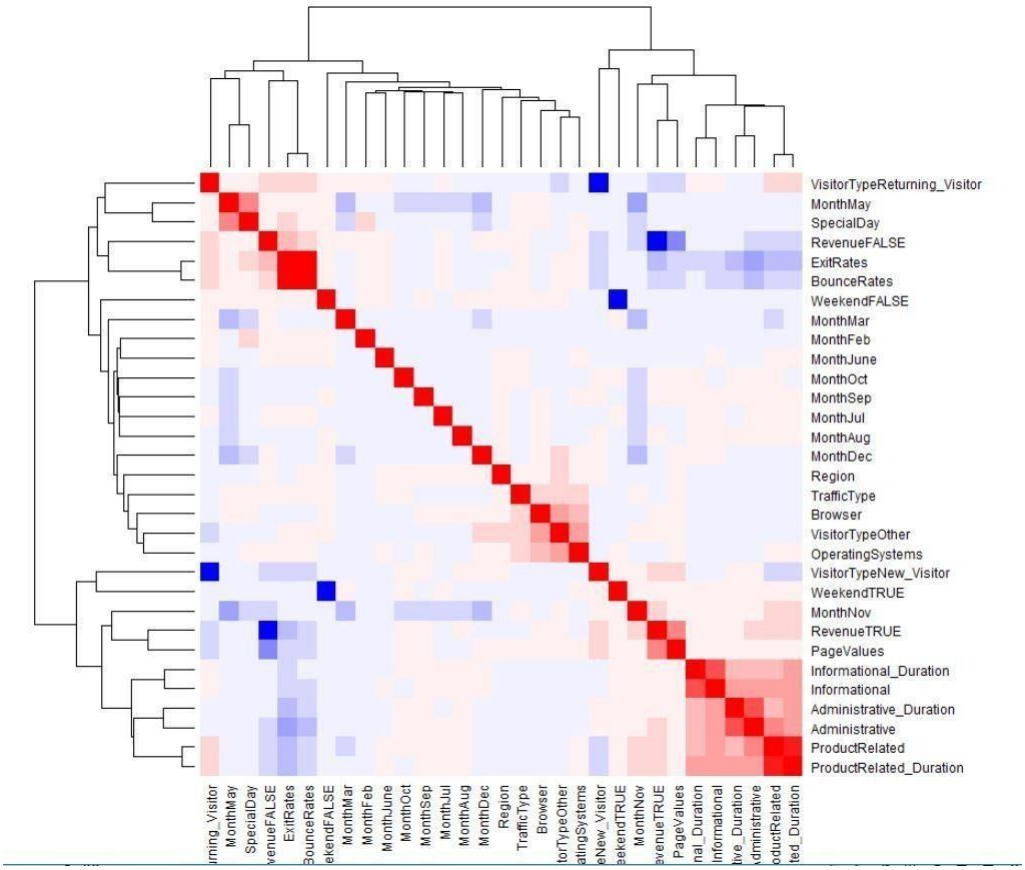
1.2.1.1 Ratio of Descriptive Features

First plot shows the count of values for each features.



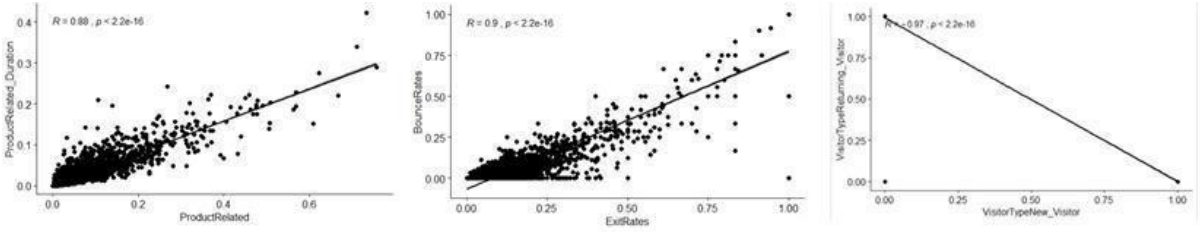
There are too many zeros in each descriptive feature and other values looks like an outlier. To prevent this inconsistent situation, normalization should have applied to each features except target feature before applying machine learning algorithms.

1.2.1.2 Heat Map of Correlation Matrix

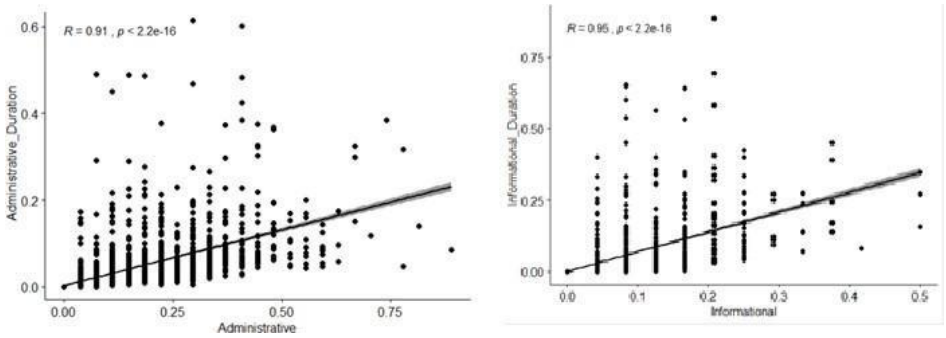


1.2.1.3 Correlation Methods

Firstly pearson correlation matrix is created with the cor function to determine if a relationship and direction exists between the variables. And it showed that There is a strong uphill(positive) linear relationship between ProductRelated and ProductRelated_Duration(r=0.88), ExitRates and BounceRates(r=0.90) at the same time there is a strong downhill (negative) linear relationship between VisitorType, New_Visitor and VisitorType, Returning_Visitor(r= -0.97).Besides p-values were computed to determine whether the correlation between variables are significant.(significance level α=0.05) Three correlation is statistically significant because p-values are less than 0.05.At the same time it
can be said that three correlation is linear relationship.



Secondly sperman correlation matrix is created. And it showed that there is a strong uphill (positive) linear relationship between Administrative and Administrative_Duration(r=0.92), Information and Information_Duration(r=0.95), ProductRelated and ProductRelated_ Duration(r=0.87) at the same time there is a strong downhill (negative) linear relationship between VisitorTypeNew_Visitor and VisitorTypeReturning_Visitor (r= -0.97). It can be said that three correlation are monotonic relationship.



2. Data Preparation

Data cleaning is applied for obtaining results closer to reality in dataset and facilitating analysis. Data cleaning has lots of steps.

2.1. Clean Data

2.1.1. Remove Duplicates

Before modelling the data in the data set, the necessary step is to check whether the data has been repeated. Repeated data is unwanted situation when performing data training and analysis. It affects the project about time and cost poorly.

The process that will done is to determine repeated instances or rows and convert them to just one instance (or row). Therefore, operations will be applied to an instance instead of multiple instances.

Function duplicated () was used for finding repeated data and removing from dataset. In the result after operation is done, 125 instances were removed repetitive each other from dataset that has 12330 instance. The data set was cleared of repetitive data.

2.1.2. Missing Values

This technique is to check whether the data is or not. It was analysed which instance contain null or NA values. In result of done technique, it is applied imputation or central method for removing missing values from dataset.

Function is.na() was used for analysis missing value. Null (NA) values in dataset was resulted such as 'TRUE', the other values (is not NA) was resulted such as 'FALSE'.

The number of instances(row) that contain a null value is shown as follows.

```
apply(is.na(duplicated_online_shopper),2,sum)
```

```
Administrative Administrative_Duration          Informational  Informational_Duration
             0                        0                      0                       0
ProductRelated ProductRelated_Duration            BounceRates              ExitRates
             0                        0                      0                       0
    PageValues              SpecialDay                  Month        OperatingSystems
             0                        0                      0                       0
       Browser                   Region            TrafficType            VisitorType
             0                        0                      0                       0
       Weekend                  Revenue
             0                        0
```

After these results, it was concluded that the data set does not contain null (NA) value. Therefore, no extra method has been applied.

2.2. Normalization

2.2.1 Dummy Attribute

Dataset has 18 descriptive feature. This descriptive features include 3 feature categorical except target feature. Normalization technique was applied for converting descriptive feature' values in dataset to common a scale without breaking difference in range of values. But, before normalization was done, categorical features in dataset were converted dummy attributes. Library 'dummy' was used for this process. No changes were made to the target feature in this use. target feature has been converted to dummy attribute in required models. Target feature has been converted to dummy attribute in required models.

```
dummy_onlineShoppers<- dummy.data.frame(duplicated_online_shopper, names = c("Month","VisitorType","Weekend"),omit.constants=FALSE,dummy.classes =getOption("dummy.classes"))
```

This three categorical feature was converted binary form by using function dummy (). In this way, all of features in dataset became continuous feature. Normalization techniques were applied in created new dataset.

There are two normalization techniques: range normalization and Z -score normalization. Range normalization technique, guarantees all features will have the exact same scale but does not handle outliers well. But, Z-score normalization technique, handles outliers, but does not produce normalized data with the exact same scale. Due to these results, the range normalization technique was used in this data set for having more consistent results.

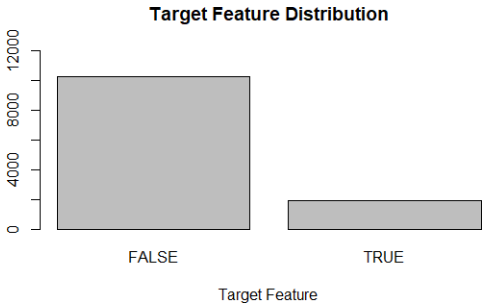
```
prepared_onlineShopper<- as.data.frame(apply(dummy_onlineShoppers[,1:num_descriptiveFeat ure], 2, function(x) (x - min(x))/(max(x)-min(x)))) prepared_onlineShopper[, 'Revenue'] <- as.factor(dummy_onlineShoppers[, 'Revenue'])
```

2.3 Undersampling Methods

There are 2 types in target feature of dataset that is chosen. This types are true and false values. Distributions of target feature values in data set have been analyzed. Results are shown in below.

```
FALSE  TRUE
10297  1908
```

Distribution matrix



Distribution bar plot

In obtained results, irregular distribution is to seen. Undersampling is applied to be equal distribution for2 types value in dataset. Number of true is1908. Number of falseis10297. So, 1908 instances that has target feature is false are chosen randomly. Remaining data has not used. So, undersampling has been applied two twice to use different instances. But, result has not changed much.

2.4 Split Train and Test Sample Groups

2.4.1 K-Fold Cross Validation

Partition operation in data set is to repeated. So, models are analyzed more good with data that are chosen from different area of dataset randomly. Number of fold for cross validation is determined such as 5 since there are 12105 instance. Data for training and test is calculated in every step and this data is applied in all of models.

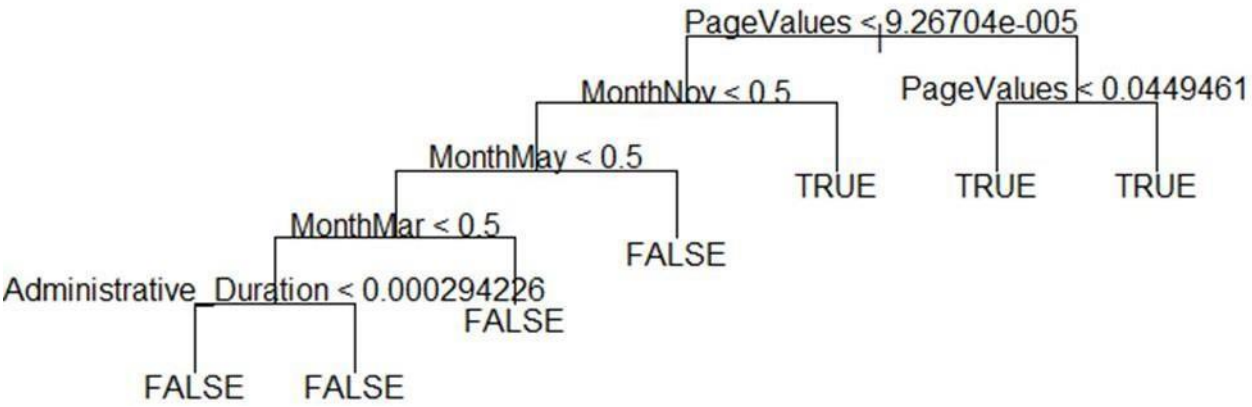
2.4.2 Data Training and Testing

After pre-processing process is applied, training and testing steps are applied to train and test our dataset. In this process, %70 of dataset for training and %30 of dataset for testing are used. Division operations in dataset is carried out randomly and after that this training and test sets are used for 5 different model.

3. Machine Learning

3.1. Decision Tree

Datasets were trained with decision tree and classification tree shown below is created.



Four features were just used when creating decision tree although there are thirty features.Other features do not influence to training.Besides total path of tree is 3.28. All metrics of datasets and means are as follows.

Decision tree	Accuracy	Precision	Recall	Fscore	AUC	Runtime(s)	F/F	F/T	T/F	T/T
Sample1	0.8829	0.7902	0.9260	0.8527	0.8953	0.05686	388	31	103	623
Sample2	0.8777	0.7849	0.9192	0.8468	0.8664	0.051	387	34	106	618
Sample3	0.8672	0.7651	0.9130	0.8326	0.8549	0.0548	378	36	116	615
Sample4	0.8637	0.8993	0.8172	0.8329	0.8820	0.0568	389	87	69	600
Sample5	0.8689	0.7958	0.8792	0.8354	0.8740	0.0568	386	53	99	607
Mean	0.8720	0.8070	0.8909	0.84	0.8745	0.0552				

3.2. Information Based Learning – Random Forest Classifier

Hyper parameters influence to performance of random forest algorithm and especially mtry and ntrees parameters are very popular in random forest

Mtry: Number of variables randomly sampled as candidates at each split. Note that the default values are different for classification (sqrt(p) where p is number of variables in x) and regression (p/3).

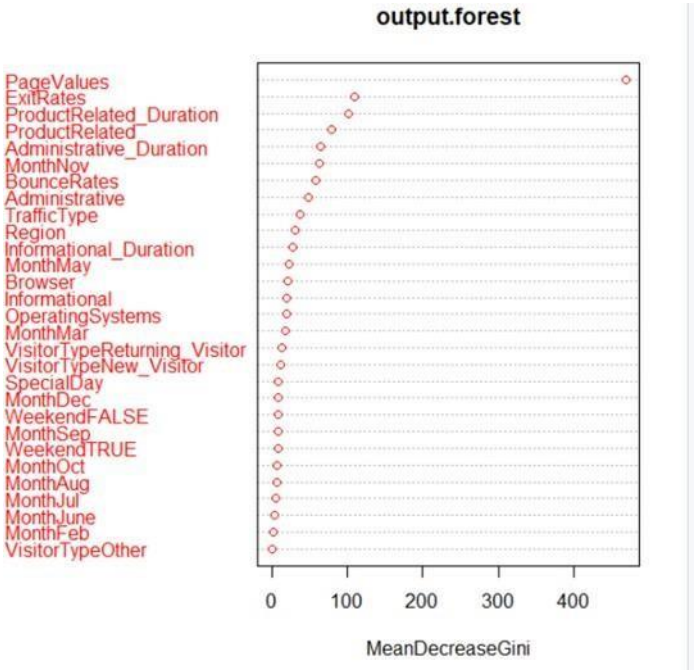
So firstly optimum mtry was selected as 5. Because decision tree algorithm use only 5 features and select greater than 5 is be senseless.

The number of trees were selected as 500 because the runtime is too high when we choose the number of trees too large.

Datasets was trained with random forest

After training a random forest, importance of each predictor is calculated by using importance function. Variables with high importance have a significant impact on the outcome values. By contrast, variables with low importance might be omitted from a model, making it simpler and faster to fit and predict.

Importance of each predictor is showed at the following graphic.



According to graphic the first value is high significant on the model when examined the graphic. Besides ExitRates and ProductRelated_Duration is significant on the model but Values close to 0 like MonthFeb,Visitor_TypeOther,MonthJune is not significant on the model and these feature should drop from the model.

All metrics of datasets and means are as follows.

random	Accuracy	Precision	Recall	Fscore	AUC	Runtime(s)	F/F	F/T	T/F	T/T	mtry
Sample1	0.9030	0.8370	0.9298	0.8796	0.8948	2.599	411	31	80	623	5
Sample2	0.8917	0.8438	0.8984	0.8702	0.8858	3.28	416	47	77	605	5
Sample3	0.8925	0.8299	0.9131	0.8695	0.8850	3.3535	400	37	94	614	5
Sample4	0.8873	0.8449	0.8694	0.8571	0.8802	3.4908	387	58	71	629	5
Sample5	0.8847	0.8412	0.8812	0.8607	0.8789	3.0444	408	55	77	605	5
Real	0.8918	0.8393	0.8983	0.8674	0.8849	3.1535					

3.3. Similarity Based Learning – K-Nearest Neighbour

Knn algorithm is applied 100 different times in dataset. This number of repetitions is defined according to number of instances in dataset. Same test and train sets are applying 100 different times with knn algorithms. Best k value is analysed according to different evaluation metrics and is to visualized with plot.

The evaluation metric values fo rall iteration of knn algorithm are shown in a table. Average of metric values for 100 iterations have been calculated. Run time has been calculated for training data.

#Fold	Accuracy	Precision	Recall	F-score	AUC	F/F	F/T	T/F	T/T	Run Time (secs)
1	0.6112664	0.4986151	0.55217	0.5190288	0.597228	253	190	238	464	0.13212
2	0.6208	0.4817	0.5717	0.5147	0.6038	244	175	249	477	0.12691
3	0.6248	0.4918	0.5766	0.5277	0.6088	246	178	248	473	0.12731
4	0.6165	0.5285	0.5202	0.5157	0.6019	248	214	210	473	0.1274016
5	0.6376	0.491	0.587	0.5252	0.6182	246	155	239	505	0.1332205

Results and run times of evaluation metrics (with undersampling)

When calculating the metrics value for the knn algorithm, 100 different k values has been calculated. Different accuracy values have been found according to these k values. Maximum accuracy has been found to determine the most efficient k value.

As a result of the analysis, the metrics values of the 5 iteration is in below. This values have been determined by taking averages of all accuracies as a result of 5 iterations.

	Accuracy	Precision	Recall	F-score	AUC
Mean	0.6221432	0.498443	0.5614429	0.520507	0.520507

Averages of results and run times of evaluation metrics (with undersampling)

K value is important for this algorithm and this value has different values for different metrics. Metrics have been had maximum k value in 100 iterations in knn algorithm. This k value and maximum metric values in each iteration is in table that is below.

K-fold	K values	Max Accuracy	K values	Max Precision	K values	Max Recall	K values	Max F-score	K values	Max AUC
1	98	0.628821	52	0.5173116	98	0.5753425	1	0.5203762	98	0.6144173
2	54	0.6401747	2	0.515213	54	0.599022	1	0.5292887	54	0.6227118
3	51	0.641048	1	0.5121457	55	0.6004843	1	0.5458468	51	0.6250163
4	88	0.631441	79	0.5611354	88	0.5378151	1	0.5329018	88	0.6193595
5	76	0.6567686	75	0.5154639	80	0.6167513	91	0.5150502	76	0.6374883

K values and maximum values of evaluation metrics (with undersampling)

In the results obtained from the table, the model applied to the undersampling data has not showed an effective performance for all metrics. Algorithm has been tried for 100 neighbours but it was analysed that the maximum results that are found in different k values have not showed not good performance.

Knn algorithm has been applied dataset without undersampling. The results obtained in the modelling without Undersampling are shown in the table below.

	Accuracy	Precision	Recall	F-score	AUC	F/F	F/T	T/F	T/T	Run Time (secs)
1	0.8467	0.9922	0.8511	0.9158	0.5238	3088	559	5	10	1.200419
2	0.8444	0.9891	0.8509	0.9141	0.5259	3075	559	15	13	1.200419
3	0.8471	0.9895	0.8533	0.9158	0.5288	3086	554	10	12	1.160909
4	0.8466	0.9914	0.8513	0.9156	0.5302	3079	560	8	15	1.158105
5	0.8471	0.991	0.8525	0.9157	0.5229	3094	553	5	10	1.144377

Results and run times of evaluation metrics (without undersampling)

Data were analyzed in 2 species such as with and without undersampling. As a result of this analysis, the rate of knowing the false target is increasing due to the high number of false in the data set. This causes the value of accuracy and other parameters to increase.

The results of the evaluation metric without undersampling as a result of 5 iteration are shown in the table.

	Accuracy	Precision	Recall	F-score	AUC
Mean	0.6221432	0.498443	0.5614429	0.520507	0.520507

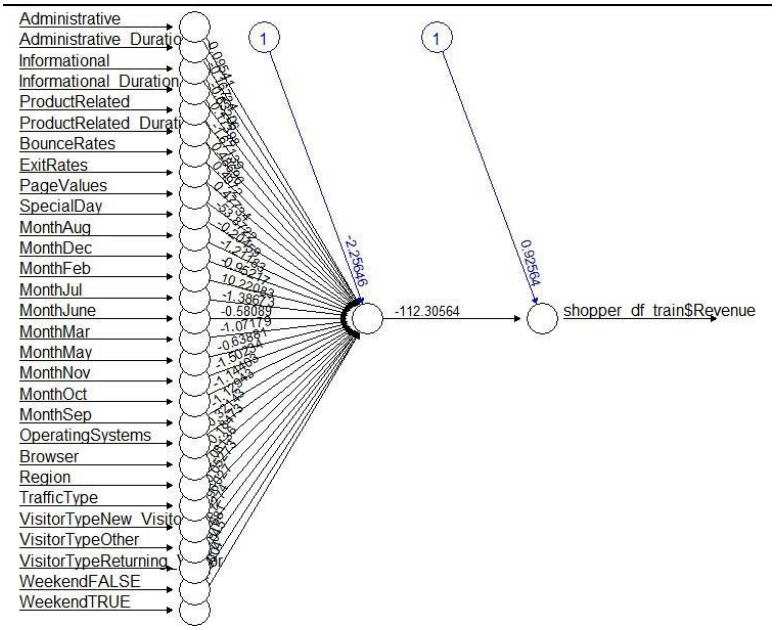
Results' means of evaluation metrics (without undersampling)

3.4. Error Based Learning – Artificial Neural Network

In the first part, data cleaning, normalization, categorical to dummy, processes are done, which means that our dataset is ready to apply ANN algorithm. In addition to it, to make probabilistic predictions (0 and 1) target feature should represented with 0s and 1s.

Dataset is under-sampled, but in order to compare how efficient the process of under- sampling, we must perform ANN algorithm with complete dataset. Train and Test splits are selected as 70% and 30% respectively.

Firstly, it is a good start to make 1 hidden layered neural network with one neuron to make a first impression.



There are 2920+384 = 3304 ***correctly predicted*** instances, however 382 ***misclassified*** instances for this model and train/test samples.

Then, ***accuracy*** obtained as nearly 90%, but there are another metrics to measure our model like precision, fscore, recall and auc values.

After the model creation is done, metrics are obtained as Precision: 0.9332, Recall: 0.9401, F-Score: 0.9366 and AUC: 0.7938354. AUC is quite low for this accuracy value because of in balance ratio between accuracy of True and False separately. Accuracy of False predictions are well, but accuracy of True is not good, it is nearly 0.64%. In order to prevent this inbalanced situation, dataset must undersampled.

	Prediction: ***False***	Prediction: ***True***
Actual: ***False***	2920	186
Actual: ***True***	209	384

Accuracy	AUC	F-Score	Precision	Recall	F/F	F/T	T/F	T/T	Runtime(s)	*Hidden 1*	*Hidden 2*	*Hidden 3*
88.995	0.8831	0.8668	0.9011	0.835	410	81	45	609	4.7355 secs	*1*	-	-
89.869	0.8922	0.8779	0.9125	0.8458	417	76	40	612	3.1709 secs	*1*	-	-
88.558	0.8777	0.8608	0.9060	0.8198	405	89	42	609	3.5609 secs	*1*	-	-
89.08297	0.8843	0.8619	0.8725	0.8515	390	68	57	630	3.3630 secs	*1*	-	-
88.38428	0.8782	0.8599	0.8793	0.8412	408	77	56	604	4.0202 secs	*1*	-	-
89.08297	0.8841	0.868	0.9013	0.8371	411	80	45	609	42.4827 secs	*5*	-	-
78.51528	0.7750	0.7377	0.7775	0.7018	346	147	99	553	39.8284 secs	*5*	-	-
87.51092	0.8667	0.8477	0.8944	0.8057	398	96	47	604	11.5870 secs	*5*	-	-
77.99127	0.7718	0.7267	0.722	0.7314	335	123	129	558	3.17957 mins	*5*	-	-
88.38428	0.8782	0.8599	0.8793	0.8412	408	77	56	604	30.8176 secs	*5*	-	-
89.34498	0.8887	0.8732	0.8917	0.8554	420	71	51	603	3.00483 mins	*2*	*3*	-
89.95633	0.8935	0.8793	0.9109	0.8499	419	74	41	611	59.1360 secs	*2*	*3*	-
88.29694	0.8825	0.8571	0.9054	0.8137	387	107	55	596	1.2354 mins	*2*	*3*	-

After the undersampling process, our models are need to re-computed. There are several hyperparameters to tune; learning rate, threshold, and stepmax, and all hyperparameters of models are set to same values except hidden. Hidden parameters determines how many hidden layers and number of neurons for each layer

$(a_x,b_y,c_z….m_n)$ *: **xth** layer has **a** neurons…, **nth** layer has **m** neurons*

In order to find out how many layers and neurons with each layers respectively, a couple of trials should calculated. Selected combinations of the layer and neurons are (1), (5) and, (2,3). Obtained results of the combinations can analyse below.

First of all, you can see that there is critical change in accuracy of true portions of predictions. It effects AUC in a good way, which means that undersampling version of the dataset must used to build a model.

On the other hand, in the calculations there are no major changes between combinations by measurement metrics, hence we can use one hidden layer with one neuron for efficiency of computing performance.

As conclusion, one hidden layer with one neuron meets requirements properly, and there is not need to add hidden layer or neurons to the model. Average of each metrics are Accuracy: 88.97785, Precision: 0.89428, Recall: 0.83866, F-Score: 0.86546, Runtime: 3.7701 and the most important observation is AUC: 0.8831. These values compared with other algorithms in Conclusion part.

3.1. Probabilistic Approach – Naïve Bayes Classifier

Undersampling model has been modelled with naïve Bayes classifier algorithm. Function naiveBayes() has been used. This algorithm is to calculate a probability for each feature (Bayes Theorem) on this data set. After calculation, dataset that is divided for test has been given to model and test dataset was analysed according to this probability. The model creates the conditional probability for each feature separately. We also have the a-priori probabilities which indicates the distribution of our data.

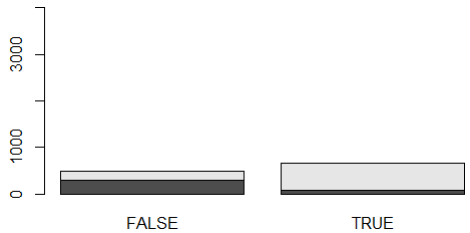
```
Naive Bayes Classifier for Discrete Predictors

Call:
naiveBayes.default(x = X, y = Y, laplace = laplace)

A-priori probabilities:
Y
    FALSE      TRUE
0.4223137 0.5776863

Conditional probabilities:
       X
Y           [,1]     [,2]
  FALSE 4948.303 2981.565
  TRUE  7054.016 3239.819

       Administrative
Y             [,1]      [,2]
  FALSE 0.07945889 0.1199765
  TRUE  0.13064977 0.1399996

       Administrative_Duration
Y             [,1]       [,2]
  FALSE 0.02144386 0.04603095
  TRUE  0.03702901 0.06315896

       Informational
Y             [,1]       [,2]
  FALSE 0.01769356 0.04659872
  TRUE  0.03415965 0.06554594

       Informational_Duration
Y             [,1]       [,2]
  FALSE 0.01271515 0.05709263
  TRUE  0.02353357 0.07084040

       ProductRelated
Y             [,1]       [,2]
  FALSE 0.03970751 0.05428380
  TRUE  0.06855302 0.08553329

       ProductRelated_Duration
Y             [,1]       [,2]
  FALSE 0.01604951 0.02481135
  TRUE  0.02926409 0.03739325

       BounceRates
Y             [,1]       [,2]
  FALSE 0.11138832 0.23833627
  TRUE  0.02650823 0.06152321

       ExitRates
Y             [,1]       [,2]
  FALSE 0.22323242 0.23988054
  TRUE  0.09789671 0.08328147

       PageValues
Y             [,1]       [,2]
  FALSE 0.005347226 0.02708661
  TRUE  0.074981538 0.09527485

       SpecialDay
Y             [,1]       [,2]
  FALSE 0.08191489 0.2248751
  TRUE  0.02488658 0.1278485

       MonthAug
Y             [,1]       [,2]
  FALSE 0.03546099 0.1850239
  TRUE  0.04471808 0.2067512
```

To analyze performance of data in dataset that was calculated with conditional probability has been applied predict() function. So , Result of test data in model has been shown as matrix. In first iteration, result of data that is predicted is in below.

Prediction	Actual FALSE	TRUE
FALSE	302	84
TRUE	189	570

Naïve Bayes Classifier Confusion matrix

In this iteration , 84 instances of 386 instances that target feature is false have been misclassified. At the same time ,189 instances of 759 instances that target feature is true have been misclassified. Predictions of target features are visualized as follows.



Prediction of Naïve Bayes Classifiers Algorithm

The other metrics have been calculated in dataset that was modelled, too. Results of analyse is in below table. For each iteration, results of all evaluation metrics and run time is defined in table.

	Accuracy	Precision	Recall	F-score	AUC	F/F	F/T	T/F	T/T	Run Time (secs)
1	0.7616	0.6151	0.7824	0.69	0.743	302	84	189	570	0.090756
2	0.7703	0.5862	0.8305	0.69	0.75	289	59	204	593	0.1156609
3	0.7467	0.581	0.7757	0.66	0.73	287	83	207	568	0.0857689 4
4	0.7415	0.5328	0.7485	0.62	0.71	244	82	214	605	0.1233358
5	0.7581	0.5711	0.8006	0.6667	0.7333	277	69	208	591	0.114979

Values and run time of all of metrics (with undersampling method)

After 5-fold cross validation, optimum values have been analyzed by calculating averages of all of metrics.

	Accuracy	Precision	Recall	F-score	AUC
Mean	0.75564	0.57724	0.78754	0.67134	0.73326

Mean values of all of metrics (with undersampling method)

Results of models that are worked in dataset which has not been undersampling are specified in below table. When analyse is done between two table, for naïve Bayes classifier algorithm, dataset that is applied undersampling has been seen that has high performance values more.

	Accuracy	Precision	Recall	F-score	AUC
Mean	0.6895685	0.6700398	0.9455473	0.7841801	0.7841801

Mean values of all of metrics (without undersampling method)

3.2. Discriminative Classifier – Support Vector Machine

Support vector machine linear, polynomial and radial kernel were applied to five undersampling datasets respectively. Different hyper parameters were used to optimize our algorithms in svm.

The most common parameters are as follows:

- Degree: parameter needed for kernel of type polynomial (default: 3)
- Gamma: parameter needed for all kernels except linear (default: 1/(data dimension))
- Cost: cost of constraints violation (default: 1) - it is the 'C'-constant of the regularization term in the Lagrange formulation.

The value of gamma and C should not be very high because it leads to the overfitting or it shouldn't be very small (underfitting). Thus we need to choose the optimal value of C and Gamma in order to get a good fit.

A: Support Linear Kernel

Firstly datasets were trained with svm linear.Tuned function is used to optimum cost value and it is found as 4. All metrics of datasets and means are as follows.

Svm/linear	Accuracy	Precision	Recall	Fscore	AUC	Runtime(s)	F/F	F/T	T/F	T/T	cost
Sample1	0.8532	0.8289	0.8289	0.8532	0.8502401	1.2712	407	84	84	570	4
Sample2	0.8471	0.8215	0.8231	0.8480	0.8440	1.9164	405	87	88	565	4
Sample3	0.8244	0.8187	0.7606	0.7941	0.8235	1.0262	375	118	83	569	4
Sample4	0.8393	0.8016	0.8215	0.8496	0.8347	1.2673	396	86	98	565	4
Sample5	0.8410	0.8288	0.8023	0.8274	0.8394	1.70	402	99	83	561	4
Mean	0.8410	0.8199	0.8072	0.8344	0.8383	1.4362					4

B: Support Polynomial Kernel

Firstly optimum gamma and cost were found by using tune function for svm polynomial and datasets were trained with optimum gamma,cost and 3 different degree.

```
tuned_parameters <- tune.svm(Revenue~., data online_shopper_training,kernel="polynomial" ,gamma = 2^(-5:-5), cost = 2^(-5:5))
```

All metrics of datasets and means are as follows.

Svm/polynomial	Accuracy	Precision	Recall	Fscore	AUC	Runtime(s)	F/F	F/T	T/F	T/T	gamma	cost	degree
Sample1	0.8524	0.8268	0.8285	0.8492	0.8492	0.6971	406	84	85	570	0.03125	32	1
Sample1	0.7816	0.7209	0.7580	0.7390	0.7740	1.2676	354	113	137	541	0.03125	32	2
Sample1	0.8183	0.7637	0.8029	0.7828	0.8115	0.7280	375	92	116	562	0.03125	32	3
Sample2	0.8445	0.8215	0.8181	0.8198	0.8417	0.6686	405	90	88	562	0.03125	16	1
Sample2	0.7720	0.7079	0.7489	0.7278	0.7642	1.0259	349	117	144	535	0.03125	16	2
Sample2	0.8183	0.7586	0.8077	0.7824	0.7824	0.8110	374	89	119	563	0.03125	16	3
Sample 3	0.8366	0.8016	0.8164	0.8089	0.8324	0.6680	396	89	98	562	0.03125	16	
Sample 3	0.7720	0.6943	0.7571	0.7243	0.7626	1.08212	343	110	151	541	0.03125	16	2
Sample 3	0.8157	0.7510	0.8082	0.7785	0.8079	0.7240	371	88	123	563	0.03125	16	1
Sample 4	0.8235	0.8187	0.7591	0.7878	0.8227	0.9414	375	119	83	568	0.03125	32	1
Sample 4	0.7563	0.7096	0.6996	0.6906	0.7485	1.2776	325	146	133	541	0.03125	32	2
Sample 4	0.7903	0.7445	0.7349	0.7396	0.7827	0.6721	341	123	117	564	0.03125	32	3
Sample 5	0.8410	0.8305	0.7988	0.8165	0.8425	0.7188	405	102	80	558	0.03125	32	
Sample 5	0.7729	0.6969	0.7494	0.7222	0.7628	1.0351	338	113	147	547	0.03125	32	

Sample 5	0.8139	0.7381	0.8063	0.7707	0.8039	0.7393	358	86	127	574	0.03125	32	
Mean	0.8072	0.8579	0.7795	0.7693	0.7992	0.8704							

Thehighestvalues ofall metrics areequal todegree 1. this datasetis bestrepresented by a linear model

C: Support Radial Kernel

In the same way data sets were trained with optimum gamma and cost by using svm radial kernel. All metrics of datasets and means are as follows.

Svm/radial	Accuracy	Precision	Recall	Fscore	AUC	Runtime(s)	F/F	F/T	T/F	T/T	gamma	cost
Sample1	0.8331	0.7963	0.8112	0.8036	0.8285	1.5378	391	91	100	563	0.03125	4
Sample2	0.8454	0.7971	0.8361	0.7824	0.8395	1.2586	393	77	100	575	0.03125	4
Sample3	0.8427	0.7854	0.8398	0.8117	0.8358	1.1300	388	74	106	577	0.03125	4
Sample4	0.8366	0.8050	0.7901	0.7978	0.8315	1.0920	369	98	89	589	0.03125	4
Sample5	0.8331	0.8	0.8024	0.8024	0.8287	1.1471	388	94	97	566	0.03125	4
Real	0.8331	0.7962	0.8159	0.7995	0.8328	1.2331						

4. Conclusion

All supervised learning techniques are evaluated; information based, similarity based, error based, probabilistic approach and discriminative classifier. In order to compare these models there are several metrics; Accuracy, Precision, Recall, F-Score, and AUC. It is a mistake to take only one metric as Accuracy, because there can be in balance accuracy between true/positive and false/positive values.

There is a comparison table of each algorithm below. These values are average of each metric.

	Accuracy	Precision	Recall	F-Score	AUC	Runtime
KNN	0.6221432	0.498443	0.5614429	0.520507	0.520507	1.1902
RF	0.8918	0.8393	0.8983	0.8674	0.8849	3.1535
ANN	0.88.97785	0.89428	0.83866	0.86546	0.8831	3.7701
Naïve B.	0.75564	0.57724	0.78754	0.67134	0.73326	0.0974
SVM	0.8410	0.8199	0.8072	0.8344	0.8383	1.4362

First of all, KNN, and Naïve Bayes algorithms are requiring less computational power than other algorithms. However, average values of metrics are quite low than other models. It means that, we should apply more complex computation than these techniques. It is seen that all metrics of random forest and ann algorithms are quite high. Especially accuracy and auc metrics are very close together but random forest is a bit faster than ann so random forest is better than others.

References

Dataset: https://archive.ics.uci.edu/ml/datasets/Online+Shoppers+Purchasing+Intention+Dataset

Kelleher, John, et al. Fundamentals of Machine Learning for Predictive Data Analytics. The MIT Press, 2015.

Machine Learning Tutorial for Beginners. (n.d.). Retrieved from https://www.kaggle.com/kanncaa1/machine-learning-tutorial-for-beginners

Mitchell, T. (1997). Machine learning. Singapore: McGraw-Hill.

Sakar, C.O., Polat, S.O., Katircioglu, M. et al. Neural Comput & Applic (2018). https://doi.org/10.1007/s00521-018-3523-0

Warwick.ac.uk. (2019). Plotting the Iris Data. [online] Available at: https://warwick.ac.uk/fac/sci/moac/people/students/peter_cock/r/iris_plots/

Towards Data Science. (2019). Activation functions and it's types-Which is better?. [online] Available at: https://towardsdatascience.com/activation-functions-and-its-types- which-is-better-a9a5310cc8f

Neural Networks in R Tutorial – Learn by Marketing. (2019). Retrieved from http://www.learnbymarketing.com/tutorials/neural-networks-in-r-tutorial/

Understanding Activation Functions in Neural Networks. (2019). Retrieved from https://medium.com/the-theory-of-everything/understanding-activation-functions-in-neural-networks-9491262