Electronics and Communication Engineering

Academic Year 2019 - 2023

Pattern Recognition Project

Online Shoppers Purchasing Intention

Which Classification Model is the Best? (for given Dataset)

Group: 8
Members:
Preethi G - S20190020241
Subash J - S20190020253
Shri Teja Naik - S20190020223

1 Project Description

We are on an era where the online shopping is booming. Everyday countless number of customers visit online stores but only a fraction of them end up making a purchase. So, predicting whether someone will make a purchase or not holds a important place for the company. In this project we have developed several classification models that predicts purchase intentions and picked the best one of them. We assume a two-class prediction problem, where the goal is to predict whether the company makes a Revenue of the visitor or not.

2 Description of the Data

Dataset Information:
The dataset consists of feature vectors belonging to 12,330 sessions.
The dataset was formed so that each session would belong to a different user in a 1-year period to avoid any tendency to a specific campaign, special day, user profile, or period.

Attribute Information:
The dataset consists of 18 attributes. The **Revenue** attribute can be used as the class label.

- "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 pageviews 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 Valentina'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.

3 Methodology

3.1 Downloading the Dataset

The Dataset was downloaded from the UCI Machine Learning Repository. It satisfies the conditions of having more than 10 attributes and consists more than 1000 instances.

3.2 Pre-processing

- The Dataset is read and checked for any missing data points in the dataset. If there was any missing data points we can either delete those rows or replace the NaN values with mean values of respective feature.

	Administrative	Administrative_Duration	Informational	Informational_Duration	ProductRelated	ProductRelated_Duration	BounceRates	ExitRates	PageValues	SpecialDay
count	12330.000000	12330.000000	12330.000000	12330.000000	12330.000000	12330.000000	12330.000000	12330.000000	12330.000000	12330.000000
mean	2.315166	80.818611	0.503569	34.472398	31.731468	1194.746220	0.022191	0.043073	5.889258	0.061427
std	3.321784	176.779107	1.270156	140.749294	44.475503	1913.669288	0.048488	0.048597	18.568437	0.198917
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	0.000000	0.000000	0.000000	0.000000	7.000000	184.137500	0.000000	0.014286	0.000000	0.000000
50%	1.000000	7.500000	0.000000	0.000000	18.000000	598.936905	0.003112	0.025156	0.000000	0.000000
75%	4.000000	93.256250	0.000000	0.000000	38.000000	1464.157213	0.016813	0.050000	0.000000	0.000000
max	27.000000	3398.750000	24.000000	2549.375000	705.000000	63973.522230	0.200000	0.200000	361.763742	1.000000

- There was no missing data points found in the dataset.

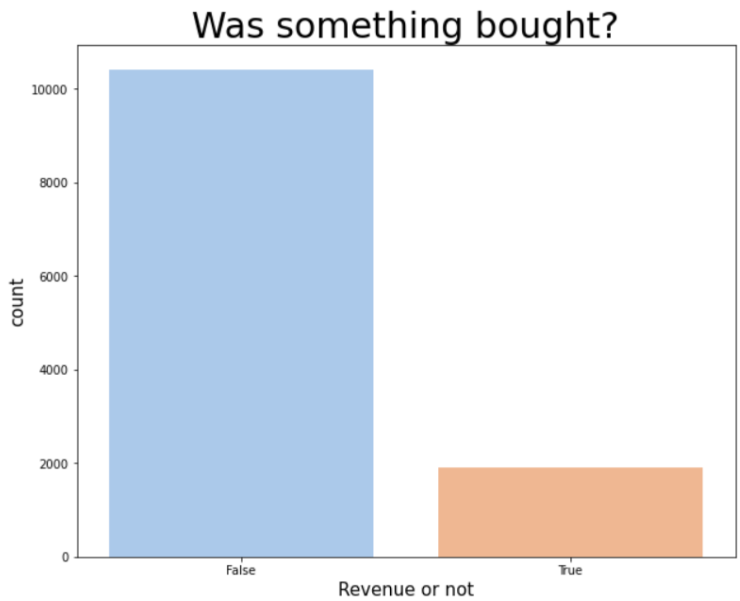
```
Administrative             0
Administrative_Duration    0
Informational              0
Informational_Duration     0
ProductRelated             0
ProductRelated_Duration    0
BounceRates                0
ExitRates                  0
PageValues                 0
SpecialDay                 0
Month                      0
OperatingSystems           0
Browser                    0
Region                     0
TrafficType                0
VisitorType                0
Weekend                    0
Revenue                    0
dtype: int64
```

- Then the data types of the features were checked and we had 2 categorical data. Categorical data can't be used for classification directly. They are usually label encoded and followed by one hot encoding.

```
Administrative               int64
Administrative_Duration    float64
Informational                int64
Informational_Duration     float64
ProductRelated               int64
ProductRelated_Duration    float64
BounceRates                float64
ExitRates                  float64
PageValues                 float64
SpecialDay                 float64
Month                       object
OperatingSystems             int64
Browser                      int64
Region                       int64
TrafficType                  int64
VisitorType                 object
Weekend                       bool
Revenue                       bool
dtype: object
```

```
Index(['Administrative', 'Administrative_Duration', 'Informational',
       'Informational_Duration', 'ProductRelated', 'ProductRelated_Duration',
       'BounceRates', 'ExitRates', 'PageValues', 'SpecialDay',
       'OperatingSystems', 'Browser', 'Region', 'TrafficType', 'Weekend',
       'Revenue', 'Month_Aug', 'Month_Dec', 'Month_Feb', 'Month_Jul',
       'Month_June', 'Month_Mar', 'Month_May', 'Month_Nov', 'Month_Oct',
       'Month_Sep', 'VisitorType_New_Visitor', 'VisitorType_Other',
       'VisitorType_Returning_Visitor'],
      dtype='object')
```

- Next we plot to check the distribution of customers on Revenue and verify the values by label encoding Revenue column.




```
0    10422
1     1908
Name: Revenue, dtype: int64
```

- Now we store the target column in y and remove the target column i.e. Revenue from X. Also some other columns (Operating System, Browser, Region and Traffic Type) are dropped as they don't necessarily make a big impact as a feature in the classification. We then check the shape of X & y and proceed to make train-test-split (80% and 20%).

```
Shape of X: (12330, 24)
Shape of y: (12330,)
```

3.3 Classification Models

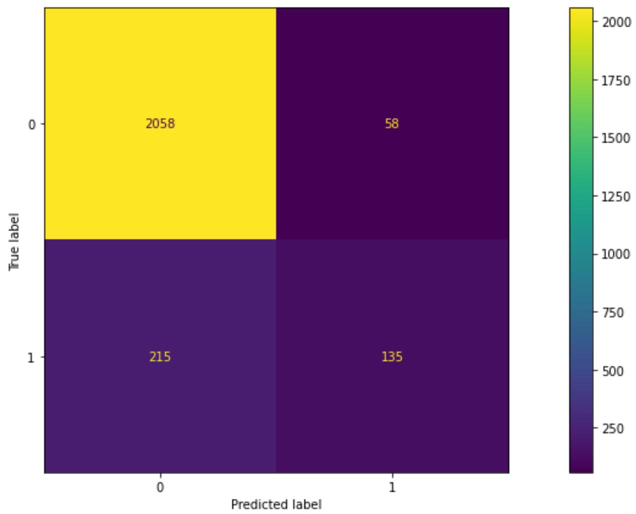
The following classification models were used:

- Logistic Regression
- Linear SVM Model
- Quadratic SVM Model
- K-nearest Neighbors Model
- Random Forest Classifier
- Neural Networks for Classification

4 Observations – Accuracy and Confusion Matrix

4.1 Logistic Regression

```
Training Accuracy : 0.8817923763179237
Testing Accuracy : 0.889294403892944
```



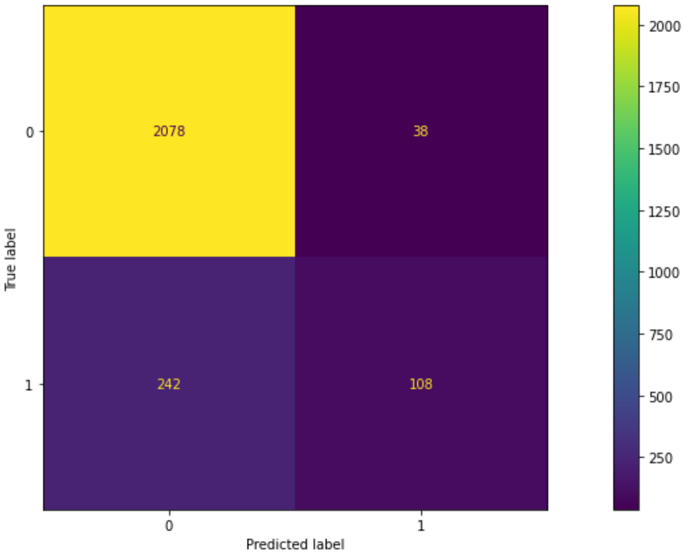

```
              precision    recall  f1-score   support

           0       0.91      0.97      0.94      2116
           1       0.70      0.39      0.50       350

    accuracy                           0.89      2466
   macro avg       0.80      0.68      0.72      2466
weighted avg       0.88      0.89      0.88      2466
```

4.2 Linear SVM Model

```
Training Accuracy : 0.879257907542579
Testing Accuracy : 0.8864557988645579
```



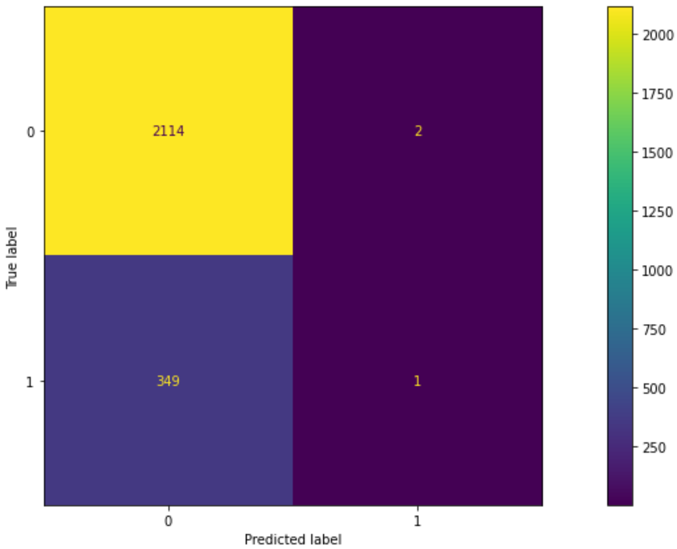

```
              precision    recall  f1-score   support

           0       0.90      0.98      0.94      2116
           1       0.74      0.31      0.44       350

    accuracy                           0.89      2466
   macro avg       0.82      0.65      0.69      2466
weighted avg       0.87      0.89      0.87      2466
```

4.3 Quadratic SVM Model

```
Training Accuracy : 0.8427615571776156
Testing Accuracy : 0.8576642335766423
```



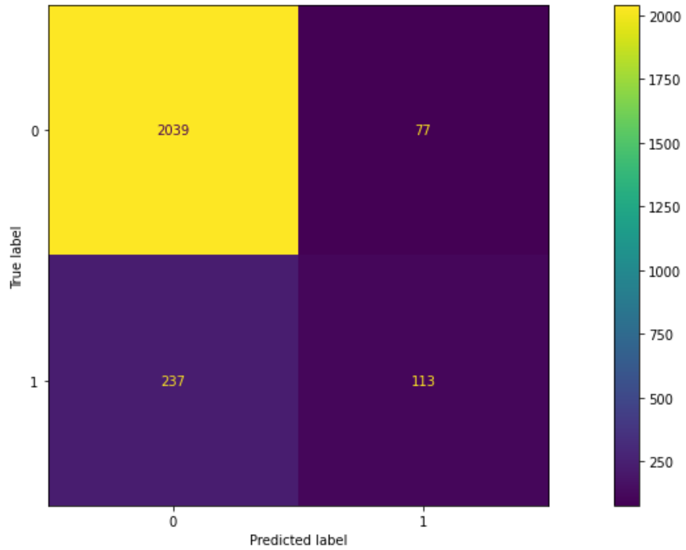

```
              precision    recall  f1-score   support

           0       0.86      1.00      0.92      2116
           1       0.33      0.00      0.01       350

    accuracy                           0.86      2466
   macro avg       0.60      0.50      0.46      2466
weighted avg       0.78      0.86      0.79      2466
```

4.4 K-nearest Neighbors Model

```
Training Accuracy : 0.8929440389294404
Testing Accuracy : 0.8726682887266829
```



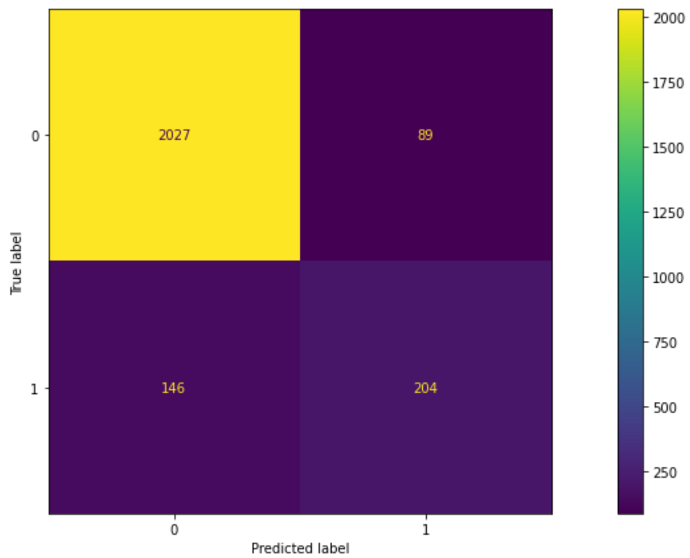

```
              precision    recall  f1-score   support

           0       0.90      0.96      0.93      2116
           1       0.59      0.32      0.42       350

    accuracy                           0.87      2466
   macro avg       0.75      0.64      0.67      2466
weighted avg       0.85      0.87      0.86      2466
```

4.5 Random Forest Classifier

```
Training Accuracy : 0.9997972424979724
Testing Accuracy : 0.9047039740470397
```




```
              precision    recall  f1-score   support

           0       0.93      0.96      0.95      2116
           1       0.70      0.58      0.63       350

    accuracy                           0.90      2466
   macro avg       0.81      0.77      0.79      2466
weighted avg       0.90      0.90      0.90      2466
```

4.6 Neural Networks for Classification

```
Training Accuracy : 0.8947688341140747
Testing Accuracy : 0.8990267515182495
tf.Tensor(
[[1992  124]
 [ 125  225]], shape=(2, 2), dtype=int32)
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense (Dense)               (None, 64)                1600

 dropout (Dropout)           (None, 64)                0

 dense_1 (Dense)             (None, 32)                2080

 dropout_1 (Dropout)         (None, 32)                0

 dense_2 (Dense)             (None, 16)                528

 dropout_2 (Dropout)         (None, 16)                0

 dense_3 (Dense)             (None, 1)                 17

=================================================================
Total params: 4,225
Trainable params: 4,225
Non-trainable params: 0
_________________________________________________________________
```

Testing precision = 0.9414
Testing F1 Score = 0.9412

5 Performance Comparisons

S.No.	Classification Model	Testing Accuracy
1	Logistic Regression	0.889294403892944
2	Linear SVM	0.8864557988645579
3	Quadratic SVM	0.8576642335766423
4	K-nearest Neighbors	0.8726682887266829
5	Random Forest	0.9047039740470397
6	Neural Networks	0.8990267515182495

6 Summary

By the above mentioned Performance Comparison table, we can conclude that the Random Forest model reaches the highest accuracy of 90.47% on the given data, followed by the Neural Networks classification model with an accuracy of 89.90%.
It can also be observed that all the models is better at predicting the negative examples as the recall value is fairly high for every model.
The other models also perform fairly well with the accuracies being more than 85% for each. But it is to be noted that the Quadratic SVM model predicts the majority of the label to be one class. This happens because the Quadratic SVM model optimizes the loss and quickly realises that if the Revenue is 'False' for so many data-points, the outputs should most likely be 'False' to achieve a great result. Thus we should also consider the Precision & the F1-score and we can choose the Random Forest Classifier to be the Model that makes the best fit for the chosen two-class prediction problem.

7 Contributions

- Subash J
  Pre-processing the dataset
  Training Linear and Polynomial Support Vector Machine Models
  Calculating Accuracy and Confusion Matrix of the respective models

- Shri Teja Naik
  Training Logistic Regression Model on the data
  Training K-nearest Neighbors Model
  Calculating Accuracy and Confusion Matrix of the respective models

- Preethi G
  Training Random Forest Model and building Neural Networks based model
  Calculating Accuracy and Confusion Matrix of the respective models
  Organising the final report

8 References

1. Abstract: `https://link.springer.com/article/10.1007/s00521-018-3523-0`
2. `https://scikit-learn.org/stable/`
3. Label encoder and One-hot encoder