Sri Lanka Institute of Information Technology

Machine Learning

Semester 01 - Year 04 – 2020

Classification of Customer buying pattern using Logistic Regression Algorithm

Assignment 01

Submitted by:

Hariharan V
IT17044400

April 20th 2020

Table of Contents

1. Introduction

1.1 Problem Statement

This is a very common problem that occur in this current world. Many business are based on their customers. It is very challenging to run a business nowadays, because many factors depends on it such as: Customers, type of business, services or products provided, internal factors (Customer relationship / supplier, etc.) and external factors (Natural disasters / any other bad conditions). This assignment is based on predicting the customers buying pattern based on few features on a social network advertisements.

Business in these days uses social media as its secondary plan to reach its customers. So it is very important to plan and evaluate this area. Social media advertisements covers a lot of people within a short period of time. So this assignment has a dataset which is based on few customers who collaborate with social media advertisements. This assignment considers its customer's 'age', 'gender' and 'estimated salary' to predict their buying patterns.

This can be a good prediction when the business is targeting a specific range of customers. They can be taking measures and consider other factors to run their business a profitable one. And they can excel in their business.

Some of the statistics that shows social media advertisement is one of the most crucial thing for the business in these days:

- 3.5 billion People uses social media platform.
- Social media active users:
    - Millennial: 90.4%
    - Generation X: 77.5%
    - Baby Boomers: 48.2%
- In this world averagely people spend 3 hours in social media networks.
- Many (73%) marketers believe that social media is one of the best effective tools for their business.
- Researching products using social media marketing has increase and 54% of social borrowers research on it.

2. Methodology

2.1 Data Collection

2.1.1 Dataset

The dataset for this classification algorithm was taken from Kaggle [2]. This predicts the buying pattern of the customers. This dataset is in the format of CSV. The main feature of this dataset is purchased (1) or not purchased (0).

Dataset from (Source)	[2]
Number of rows or instances	400
Number of features and attributes	05
Number of labels or classes	02
Number of null or missing values	Null
Related tasks	Classification

Table 1: Data characteristics

2.1.2 Description of Dataset

The following table contains the description of the dataset used for this assignment.

Attribute	Description
User ID	Id of the user in the system.
Gender	Gender of the user. (Male / Female)
Age	Age of the user.
Estimated Salary	Salary of the user.
Purchased (Outcome)	0: Not purchased 1: Purchased

Table 2: Dataset attributes description

"Purchased" and "Not Purchased" are the two classes in this dataset. We are going to predict whether a person is purchasing (1) or not purchasing (0) based on their characteristics described in the CSV file.

2.2 Data Preprocessing

Portion of the dataset chosen for predicting the buying patter is shown in Figure 1.

	User ID	Gender	Age	Estimated Salary	Purchased
0	15624510	Male	19	19000	0
1	15810944	Male	35	20000	0
2	15668575	Female	26	43000	0
3	15603246	Female	27	57000	0
4	15804002	Male	19	76000	0

Figure 1: Dataset before preprocessing

Firstly, the dataset was checked for null vales (NaN), there were no null values in any columns from the dataset. For the classification all the data should in the format of numerical value. So the "Gender" column is converted to integer (from Male to 1 and Female to 2).

To convert the "Gender" a (for loop) is run for the whole "Gender" column. It traverse through the columns, when the value matches with male or female its corresponding numeric number is replaced in the cell.

Portion of the dataset after preprocessing and which is chosen for predicting the buying patter is shown in Figure 2.

	User ID	Gender	Age	Estimated Salary	Purchased
0	15624510	1	19	19000	0
1	15810944	1	35	20000	0
2	15668575	2	26	43000	0
3	15603246	2	27	57000	0
4	15804002	1	19	76000	0

Figure 2: Dataset after preprocessing

2.3 Logistic regression algorithm

2.3.1 Selection of the algorithm

Logistic regression is one the most used algorithm in machine learning. This is mainly used for problems like classification. This is a predictive analysis algorithm and based on the concept of probability. This uses a different approach for estimating the parameters, which gives better unbiased results, with lower variance.

Logistic regression has its own advantage compared to other algorithms.

- This performs better when dataset is linearly separable.
- This is less prone to over-fitting.
- This is easy to implement.
- This is easy to interpret.
- This is very efficient to train its dataset.

The main steps followed in the classification process is:

1. Firstly, identified the independent (features) and prediction variable (dependent).
2. Visualize them individually, their relationships and characteristics.
3. Data preprocessing.
4. Breaking the dataset into two parts as dependent (prediction variable) and independent (features).

5. Scaling the independent variables (features).
6. Training the model
7. Prediction whether purchased or not.

2.3.2 Implementation of Logistic regression.

1. **Splitting the dataset as features and labels.**

   In here the features are the independent variables they are: age, gender and estimated salary.
   And the label whether the customer is buying or not buying a product.

```
# Independent (Features) Data: Gender, Age, Estimated Salary
# Data for prediction: Purchased
person_data = temp.ix[:,(1,2,3)].values
person_data_names = ['Gender','Age', 'EstimatedSalary']

y = temp.ix[:,4].values
```

   In here we drop the user id and take the features as the “person_data” and label as the ”y”.

2. **Scale the features**

   Scaling the features is a recommended step while making prediction using logistic regression.

```
# Defining X variable

X = scale(person_data)
```

3. **Make prediction**

   Training the model for prediction.
   This steps includes the classification report as well.

```
# Logistic regression prediction

LogReg = LogisticRegression()

LogReg.fit(X, y)
'{:f}'.format(LogReg.score(X, y))
```

```
# Prediction report

y_pred = LogReg.predict(X)
```

```
from sklearn.metrics import classification_report
print(classification_report(y, y_pred))
```

2.4 Testing

If the model is over trained with the dataset, we have to analyze the prediction, accuracy of the classification algorithm. Also we have to process the extent of the time our classifier will accurately predict the new data. This is known as testing

1. **Find the accuracy of the predicted data**

   After training the model, around 85% of accuracy was achieved using logistic regression algorithm.

```
# Prediction report

y_pred = LogReg.predict(X)
from sklearn.metrics import classification_report
print(classification_report(y, y_pred))
```

	precision	recall	f1-score	support
0	0.86	0.92	0.89	257
1	0.84	0.73	0.78	143
accuracy			0.85	400
macro avg	0.85	0.82	0.83	400
weighted avg	0.85	0.85	0.85	400

Figure 3: accuracy of the prediction

2. **Production of classification report for the buying pattern predictions**

   This is a summary which measures the quality from a classification algorithm. The classification report function summarizes and produces a text report displaying the important classification metrics such as: the precision, recall, F1, and model's support score. And the method return the summary of the classification metrics.

- Precision – The capability of the classifier to not label an instance which is negative.
- Recall - The capability of the classifier to search all the positive instances. Ratio of true positives to the sum of positives and false negatives for each class is defined as this ratio.

- f1 score - Class accuracy for classifying samples which belongs to this class judged to other classes is given by f1_score.

Support – The number of actual class happenings in the noted dataset is provided by support

	precision	recall	f1-score	support
0	0.86	0.92	0.89	257
1	0.84	0.73	0.78	143
accuracy			0.85	400
macro avg	0.85	0.82	0.83	400
weighted avg	0.85	0.85	0.85	400

Figure 4: classification report

3. Measuring the Spearman rank correlation coefficient.

Spearman rank correlation coefficient is basically tells the relationship between two variables. If the value is negative that means both the variables are inversely proportional and positive means they are directly proportional.

```
gender = temp['Gender']
age = temp['Age']
estimatedSalary = temp['EstimatedSalary']
var4 = 2.35245

# Spearman Rank Coefficient on Gender and EstimatedSalary

spearmanr_coefficient, p_value = spearmanr(gender, estimatedSalary)
'Spearmanr Rank Correlation Coefficient on Gender and EstimatedSalary {:0.3f}'.format(spearmanr_coefficient)
```

```
'Spearmanr Rank Correlation Coefficient on Gender and EstimatedSalary -0.044'
```

```
# Spearman Rank Coefficient on Gender and Age

spearmanr_coefficient, p_value = spearmanr(gender, age)
'Spearmanr Rank Correlation Coefficient on Gender and Age {:0.3f}'.format(spearmanr_coefficient)
```

```
'Spearmanr Rank Correlation Coefficient on Gender and Age -0.068'
```

```
# Spearman Rank Coefficient on Age and EstimatedSalary

spearmanr_coefficient, p_value = spearmanr(age, estimatedSalary)
'Spearmanr Rank Correlation Coefficient on Age and EstimatedSalary {:0.3f}'.format(spearmanr_coefficient)
```

```
'Spearmanr Rank Correlation Coefficient on Age and EstimatedSalary 0.125'
```

Figure 5: Spearman rank correlation coefficient.

4. Identify the types of errors made by the classifier using confusion matrix

We can use confusion matrix to summarize the performance of the algorithm. This matrix helps us to identify the precision of the classifier model, the correctness and incorrectness. In here. In this case the model predicted 276 instances has not purchased

(0) while only 237 were predicted right. 124 instances has purchased (1) while only 104 were rightly predicted.

In the customer buying pattern dataset the results of confusion matrix are shown as follows:

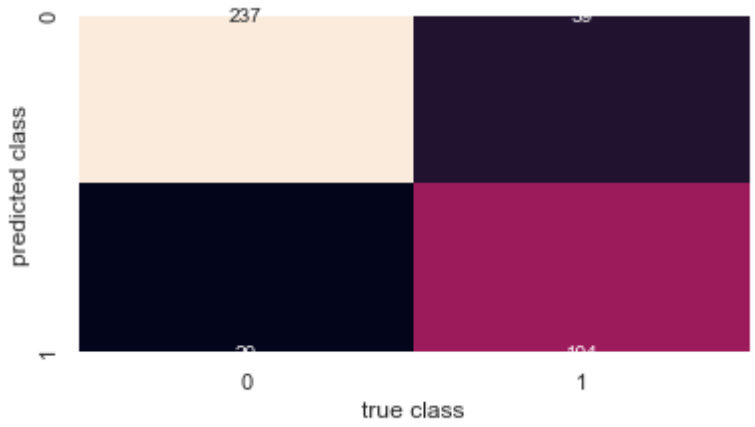

Figure 6: Confusion matrix screen shot.

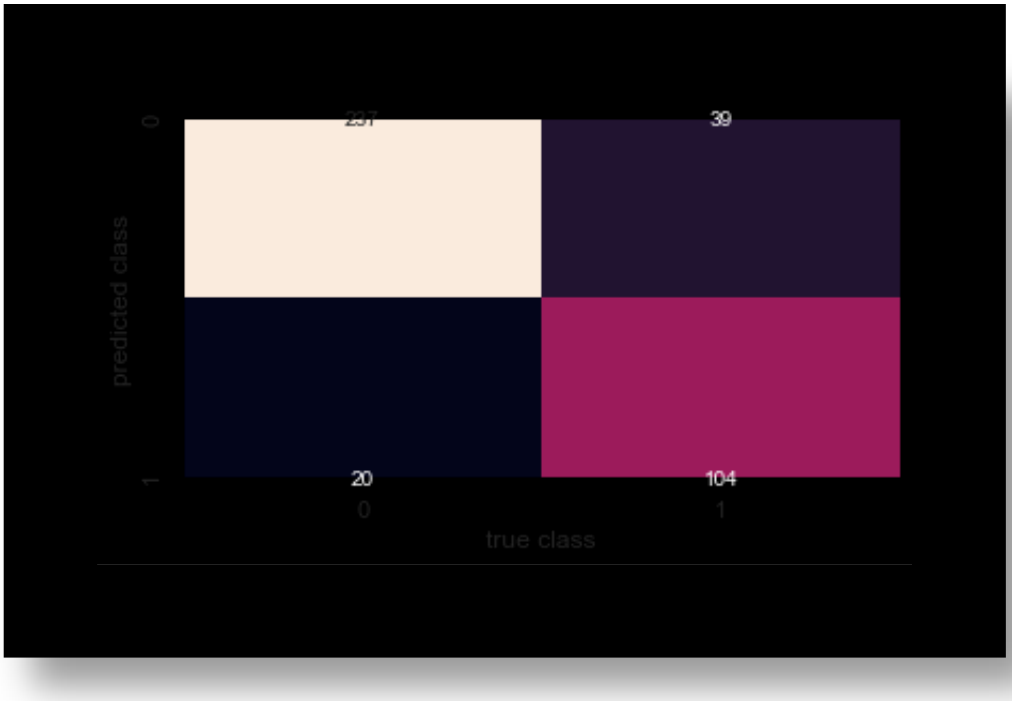

Figure 7: Confusion matrix original image.

3. Evaluation

3.1 Assessment of the project result

Binary logistic regression is the most appropriate statistical technique for purchase prediction since the prediction will be made for a categorical variable with two outcomes.
It is able to give 0.85 accuracy for the test set and it also less over fitting data. Overall accuracy of this classifier can be enhanced by experimenting with the feature extraction. Also having a large number of other features related to this purchase prediction and having a large number of dataset will be better to achieve a good accuracy.

3.2 Lesson learned

- Gained experience in learning data preprocessing
- Learned on how to train and predict using algorithm
- Learned a lot about pandas, sklearn, matplotlib.
- Learned a lot about visualizing the data.
- Learned about confusion matrix, classification_report and othertexting reports.

3.3 Future work

- Rather than having few features if we have a lot of features related to the prediction it can help in a high accurate prediction.
- Rather than having a small dataset if we have a very large dataset that will also increase the accuracy.
- Using neural networks and deep learning techniques can help in a better result.
- Finding the insight of each and every feature and the relationship within the other feature and doing feature engineering to pick up the right most features will also increase the accuracy.
- Trying out with different algorithms and collaborating more than one algorithm for prediction can help in picking up the best model and to increase the accuracy.

4. References

[1] Mohsin, M., 2020. *10 Social Media Statistics You Need To Know In 2020 [Infographic]*. [online] Oberlo. Available at: <https://www.oberlo.com/blog/social-media-marketing-statistics> [Accessed 15 April 2020].

[2] Kaggle.com. 2020. *Social Network Ads*. [online] Available at: <https://www.kaggle.com/rakeshrau/social-network-ads> [Accessed 15 April 2020].

5. Appendix

5.1 Appendix A: Code Listings

```
# ID: IT17044400
# Name: Hariharan Vasudevan

import numpy as np
import pandas as pd

from pandas import Series, DataFrame

import scipy
from scipy.stats import spearmanr

from pylab import rcParams
import seaborn as sb
import matplotlib.pyplot as plt

import sklearn
from sklearn.preprocessing import scale
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
from sklearn import metrics
from sklearn import preprocessing
```

```
%matplotlib inline
rcParams['figure.figsize'] = 5, 4
sb.set_style('whitegrid')
```

```
# Reading the dataset to a temporary variable

address = 'F:/UNI/4th year 1st semester/ML/Study/Labs/Final Assignment/IT
17044400/Social_Network_Ads.csv'
temp = pd.read_csv(address)
temp.head()
```

```
# Independent (Features) Data: Gender, Age and Estimated Salary
# Data for prediction: Purchased
# Graphical data interpretation

# Seaborn plot on the Gender, Age and Estimated Salary
```

```
sb.relplot(x="Age", y="EstimatedSalary", hue="Gender", data=temp);
```

```
# Graphical interpretation the prdiction vriable Purchased

sb.countplot(x='Purchased', data = temp, palette = 'hls')
```

```
gender = temp['Gender']
age = temp['Age']
estimatedSalary = temp['EstimatedSalary']
var4 = 2.35245

# Spearman Rank Coefficient on Gender and EstimatedSalary

spearmanr_coefficient, p_value = spearmanr(gender, estimatedSalary)
'Spearmanr Rank Correlation Coefficient on Gender and EstimatedSalary {:0
.3f}'.format(spearmanr_coefficient)
```

```
# Spearman Rank Coefficient on Gender and Age

spearmanr_coefficient, p_value = spearmanr(gender, age)
'Spearmanr Rank Correlation Coefficient on Gender and Age {:0.3f}'.format
(spearmanr_coefficient)
```

```
# Spearman Rank Coefficient on Age and EstimatedSalary

spearmanr_coefficient, p_value = spearmanr(age, estimatedSalary)
'Spearmanr Rank Correlation Coefficient on Age and EstimatedSalary {:0.3f
}'.format(spearmanr_coefficient)
```

```
# Checking all the columns for data preprocessing

null_columns=temp.columns[temp.isnull().any()]
temp[null_columns].isnull().sum()
```

```
# Data preprocessing

# creating a dict file
gender = {'Male': 1,'Female': 2}

# Traversing through dataframe
# Gender column and writing
# values where key matches
temp.Gender = [gender[item] for item in temp.Gender]
temp.head()
```

```
# Independent (Features) Data: Gender, Age, Estimated Salary
# Data for prediction: Purchased
person_data = temp.ix[:,(1,2,3)].values
person_data_names = ['Gender','Age', 'EstimatedSalary']

y = temp.ix[:,4].values
```

```
# View of person data

person_data
```

```
# View of y variable
```

```
y
```

```
# Defining X variable
# Scaling the features

X = scale(person_data)
print(type(X))
X
```

```
# Logistic regression prediction

LogReg = LogisticRegression()

LogReg.fit(X, y)
'{:f}'.format(LogReg.score(X, y))
```

```
# Prediction report

y_pred = LogReg.predict(X)
from sklearn.metrics import classification_report
print(classification_report(y, y_pred))
```

```
from sklearn.metrics import confusion_matrix
import seaborn as sebn; sebn.set()
get_ipython().run_line_magic('matplotlib', 'inline')
import matplotlib.pyplot as plot

conf_mat = confusion_matrix(y, y_pred)
sebn.heatmap(conf_mat.T, square=True, annot=True, fmt='d', cbar=False)
plot.xlabel('Class_True')
plot.ylabel('Class_Predicted')
```