Machine Learning Engineer Nanodegree
R.DHANUJA
February 21st, 2019
Proposal:
Online News Popularity
Domain Background:
History:

The online consumption news increases daily due to the widespread adoption of smartphones and the rise of social networks. Since it allows an easy and fast spread of information around the globe. Thus, predicting the popularity of online news is becoming a recent research trend.

Online platforms like Medium, Mashable and Buzzfeed etc. publish hundreds of articles every day. These articles include categories like entertainment, technology, sports , education, lifestyle etc. and are posted on different days of the week.

Reference Link:
https://medium.com/@syedsadiqalinaqvi/predicting-popularity-of-online-news-articles-a-data-scientists-report-fac298466e7

This project aims to predict the popularity of an online news article before it is published..

Applications:

This tool will help publishers and editors in maximizing the popularity of their articles and sell advertisement**.** Predicting such popularity is also valuable for authors, content providers, advertisers and even activists/politicians.

Problem Statement:

The main aim of this project is to predict the future popularity of news article prior to its publication estimating the no. of likes,shares and comments etc..(features of an article) . The dataset is publically available at https://archive.ics.uci.edu/ml/datasets/Online+News+Popularity#**.** So my goal is to predict the online news article popularity. Here I am using classification models to find the accuracy of each model and select the best model with high accuracy to predict the popularity. Here the input parameters are training data that we took and the output will be whether the news article is going to be popular or not.

Dataset Information:

The details of the features are:

Number of Attributes: 61 (58 predictive attributes, 2 non-predictive, 1 goal field)

Attribute Information

This dataset is interesting because there is a good mixture of attributes – Categorical, integer and real valued attributes. There are also few missing values. On a total there are 60 attributes. The total number of instances in the datasets is 39,643. Here there are mainly two classes: + (popular) and – (unpopular).

For the best result we will split the data into training set and testing set. On a whole we will assign 70% of the data to training set and 30% of the data to testing set.

Attributes:

0. url: URL of the article (non-predictive)

1. timedelta: Days between the article publication and the dataset acquisition (non-predictive)
2. n_tokens_title: Number of words in the title
3. n_tokens_content: Number of words in the content
4. n_unique_tokens: Rate of unique words in the content
5. n_non_stop_words: Rate of non-stop words in the content
6. n_non_stop_unique_tokens: Rate of unique non-stop words in the content
7. num_hrefs: Number of links
8. num_self_hrefs: Number of links to other articles published by Mashable
9. num_imgs: Number of images
10. num_videos: Number of videos
11. average_token_length: Average length of the words in the content

12. num_keywords: Number of keywords in the metadata
13. data_channel_is_lifestyle: Is data channel 'Lifestyle'?
14. data_channel_is_entertainment: Is data channel 'Entertainment'?
15. data_channel_is_bus: Is data channel 'Business'?
16. data_channel_is_socmed: Is data channel 'Social Media'?
17. data_channel_is_tech: Is data channel 'Tech'?
18. data_channel_is_world: Is data channel 'World'?
19. kw_min_min: Worst keyword (min. shares)
20. kw_max_min: Worst keyword (max. shares)
21. kw_avg_min: Worst keyword (avg. shares)
22. kw_min_max: Best keyword (min. shares)
23. kw_max_max: Best keyword (max. shares)
24. kw_avg_max: Best keyword (avg. shares)
25. kw_min_avg: Avg. keyword (min. shares)
26. kw_max_avg: Avg. keyword (max. shares)
27. kw_avg_avg: Avg. keyword (avg. shares)
28. self_reference_min_shares: Min. shares of referenced articles in Mashable
29. self_reference_max_shares: Max. shares of referenced articles in Mashable
30. self_reference_avg_sharess: Avg. shares of referenced articles in Mashable
31. weekday_is_monday: Was the article published on a Monday?
32. weekday_is_tuesday: Was the article published on a Tuesday?
33. weekday_is_wednesday: Was the article published on a Wednesday?
34. weekday_is_thursday: Was the article published on a Thursday?
35. weekday_is_friday: Was the article published on a Friday?

36. weekday_is_saturday: Was the article published on a Saturday?
37. weekday_is_sunday: Was the article published on a Sunday?
38. is_weekend: Was the article published on the weekend?
39. LDA_00: Closeness to LDA topic 0
40. LDA_01: Closeness to LDA topic 1
41. LDA_02: Closeness to LDA topic 2
42. LDA_03: Closeness to LDA topic 3
43. LDA_04: Closeness to LDA topic 4
44. global_subjectivity: Text subjectivity
45. global_sentiment_polarity: Text sentiment polarity
46. global_rate_positive_words: Rate of positive words in the content
47. global_rate_negative_words: Rate of negative words in the content
48. rate_positive_words: Rate of positive words among non-neutral tokens
49. rate_negative_words: Rate of negative words among non-neutral tokens
50. avg_positive_polarity: Avg. polarity of positive words
51. min_positive_polarity: Min. polarity of positive words
52. max_positive_polarity: Max. polarity of positive words
53. avg_negative_polarity: Avg. polarity of negative words
54. min_negative_polarity: Min. polarity of negative words
55. max_negative_polarity: Max. polarity of negative words
56. title_subjectivity: Title subjectivity
57. title_sentiment_polarity: Title polarity
58. abs_title_subjectivity: Absolute subjectivity level
59. abs_title_sentiment_polarity: Absolute polarity level

60.shares: Number of shares (target)

Solution Statement:

Here, I am predicting the popularity of the online news from the selected dataset. For predicting, we will use different classification models.

In this project, I will implement a classification task for online news popularity prediction using python and machine learning toolbox sklearn. Then we will find the accuracy score for each model. I explore the dataset by using read_csv and for visualization which helps me to better understand the solution, I used matplotlib.pyplot.

Benchmark Model:

Benchmark model is a model which we will take as reference and achieve the best result than the benchmark model .Here Accuracy score will be compared among the different classification models(Adaboost,Random Forest,Logistic regression) and the best model is selected.

Evaluation Metrics:

I want to use accuracy score as evaluation metric for prediction of credit approval. Here the dataset classes (+ and -) are closely balanced, so we can use accuracy as an evaluation metric. Here I am predicting the accuracy score for the selected models. Here we will select a model whose accuracy score is greater than all the other models and we treat it as the best.

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

Project Design:

The project is composed of different steps as follows:

Pre-processing:

The first task is to read the data and perform visualizations on it to get some insights about the data after. Reading the data, clean the data that is removing the unwanted data or replacing null values with some constant values or removing duplicates.

After data exploration I split the data into training and testing sets. After splitting the data we will apply classifying models and then predict the accuracy score for the selected model.

Training and Testing the data:

Here I will use the classification models like random forest, decision tree, Adaboost, logistic regression. After training the data we will test all the models with testing data. After that we will find out the accuracy score for all the models.

Finally, I will declare the model with highest accuracy score as the best model for detecting the credit card approval.