# SET09122 2020-1 TR1 001 Artificial Intelligence

Jerzy Bajer 40428545

# Table of Contents

## Introduction

The purpose of this project is developing and evaluate a sentiment analysis model. Sentiment analysis focuses on predicting whether text has positive or negative sentiment.

## 1. Evaluation

Provided dataset contains 25 000 reviews and labels. Processing of such high volume of data in a reasonable time is beyond capabilities of a computer which was used to develop this model. This is why there was used sample of data instead. There was randomly picked one hundred reviews and corresponding to them labels.

NLP pipeline consists of input, preprocessing, ML and outputs. Input consists of text (reviews) and associated labels (positive or negative sentiment). Before input can be passed to machine learning classifier text from input must be preprocessed. Text must be turned into meaningful representation.

One-hot encoding was chosen as a way to preprocess text. One-hot encoding is good choice for short text such as short reviews. In case of larger text such as paragraphs or full documents bag of words seems to be better option as it allows to count number each word(s) appear in the paragraph or document. One-hot encoding is simple to be implemented and allows to represent each sentence (review) as a binary vector where 1 means certain vocabulary's word(s) (in this case bigram) is present in sentence, 0 means certain vocabulary's word(s) (in this case bigram) is not present in sentence.

First there was created vocabulary based on bigrams generated for sample dataset. There were used bigrams instead of unigrams to capture differences between sentences for instance: negation good and not good. Additionally bigrams allow to capture context of words used for instance minister tests and minister’s tests.

Vocabulary size (l) is the number of unique bigrams vocabulary has. For each sentence in sample dataset there was initialized a vector of vocabulary’s size (l) with values 0. Then for each bigram present in each sentence which is in vocabulary 0s were replaced with 1s.

As a result there was created array of 0s and 1s. Prepared vectors and corresponding to them labels must be split into training and testing datasets. The split in this model is 80/20 (20 + 1 to be more precise). As it was mentioned above the sample data for this model is 100 reviews and corresponding to them labels. So there was used 80 randomly selected vectors and corresponding to them labels for training (data and labels) and 20 remaining vectors and corresponding to them labels +1 for testing (data and labels). Split 80/20 seems to be good choice as there is enough data for machine learning classifier to learn and there is enough data to make prediction to check how good is model’s prediction. At this point data can be passed to machine learning classifier.

Once machine learning classifier finish job prediction can be made. The model draws a text report showing the rules and decision tree (Both presented in Appendix.).

Then the model calculates and displays all evaluation metrics.

```
====================== Confusion Matrix ================================
[[ 7  2]
 [ 2 10]]
========================================================================
```

Figure 1. Confusion Matrix

Evaluation of the model starts with looking at confusion matrix. Figure 1 shows that seven sentiment values were correctly classified as negative(0) and two were wrongly classified as negative(0). Ten sentiment values were correctly classified as positive (1) and two were wrongly classified as positive (1).

```
============================ Accuracy ==================================
The prediction accuracy is:  80.95238095238095 %
========================================================================
==================== Values predicted by model =======================
[0 0 1 0 1 1 0 0 1 0 1 1 1 1 1 0 1 1 0 1 0]
========================================================================
======================= Real targeted values ===========================
[0, 0, 1, 0, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 1, 1, 0]
========================================================================
```

Figure 2. Accuracy and Prediction

Accuracy is important as it shows how many sentiment values (both positive and negative) were predicted correctly. Figure 2 shows prediction's accuracy which is about 81% which is pretty good. There are also displayed predicted sentiment values and real sentiment values for comparison. Accuracy can be calculated by dividing number of correct predictions (positive and negative) by number of all predictions. Information provided by confusion matrix can be used to calculate accuracy. For 10 predictions eight were correct.

```
==================== Evaluation Metrics ====================
              precision    recall  f1-score   support

           0       0.78      0.78      0.78         9
           1       0.83      0.83      0.83        12

    accuracy                           0.81        21
   macro avg       0.81      0.81      0.81        21
weighted avg       0.81      0.81      0.81        21

============================================================
```

Figure 3. Evaluation metrics

Accuracy may not be good indicator of model performance if class distribution is imbalanced. In such case precision and recall are handy.

Precision can be calculated by dividing correct positive prediction by all positive predictions. For negative sentiment (0) precision is 78%, for positive sentiment (1) precision is 83%. This means that 78% prediction of 0 and 83% prediction of 1 were correct.

Recall can be calculated by dividing correct positive predictions by all positives. Recall values are the same as precision values as it can be seen in Figure 3. This means that 78% of 0 and 83% of 1 were predicted correctly.

F1-score, which is the weighted average of Precision and Recall, has exactly same values as precision and recall.

It looks like sample data which was randomly selected was balanced.

```
============================== ERRORS ==============================
Mean Absolute Error: 0.19047619047619047
Mean Squared Error: 0.19047619047619047
Root Mean Squared Error: 0.4364357804719847
====================================================================
```

Figure 4. Errors

Figure 4 shows errors. Mean Absolute Error (about 19%) is absolute value of the difference between the predicted value and actual value. The smaller the value the better prediction. Mean Squared Error (about 19%) tells how good is prediction. The smaller the value the better the prediction. Root Mean Squared Error value is about 44%.

## 2. Reflection

Coming to conclusion, it is very important to notice that sample data randomly selected has huge impact on how well model performs. With different sample data randomly selected from the original dataset, the evaluation metrics may look completely different. The model was used many times and each time evaluation metrics were different. For instance accuracy was between 40% and up to 81%. This proves it is hard to create the good model for 25 000 reviews and labels using only the small sample of 100 reviews and labels.

Using a more powerful computer and bigger sample of data could help to create a model which would be less dependent on randomly selected sample. Using all 25 000 reviews in the model would allow to show how good the model really is and how good the prediction of the model is.

## Appendix

```
|--- very well <= 0.50
|   |--- movie is <= 0.50
|   |   |--- i did <= 0.50
|   |   |   |--- and the <= 0.50
|   |   |   |   |--- in a <= 0.50
|   |   |   |   |   |--- 've ever <= 0.50
|   |   |   |   |   |   |--- years later <= 0.50
|   |   |   |   |   |   |   |--- me this <= 0.50
|   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |--- me this >  0.50
|   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |--- years later >  0.50
|   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |--- 've ever >  0.50
|   |   |   |   |   |   |--- class: 1
|   |   |   |   |--- in a >  0.50
|   |   |   |   |   |--- do n't <= 0.50
|   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |--- do n't >  0.50
|   |   |   |   |   |   |--- her 'mommy <= 0.50
|   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |--- her 'mommy >  0.50
|   |   |   |   |   |   |   |--- class: 1
|   |   |   |--- and the >  0.50
|   |   |   |   |--- badly it <= 0.50
|   |   |   |   |   |--- made it <= 0.50
|   |   |   |   |   |   |--- the cast <= 0.50
|   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |--- the cast >  0.50
|   |   |   |   |   |   |   |--- leisen the <= 0.50
|   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |--- leisen the >  0.50
|   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |--- made it >  0.50
|   |   |   |   |   |   |--- class: 0
|   |   |   |   |--- badly it >  0.50
|   |   |   |   |   |--- class: 0
|   |   |--- i did >  0.50
|   |   |   |--- class: 0
|   |--- movie is >  0.50
|   |   |--- class: 0
|--- very well >  0.50
|   |--- class: 1
```

Figure 5. Text report showing the rules of a decision tree

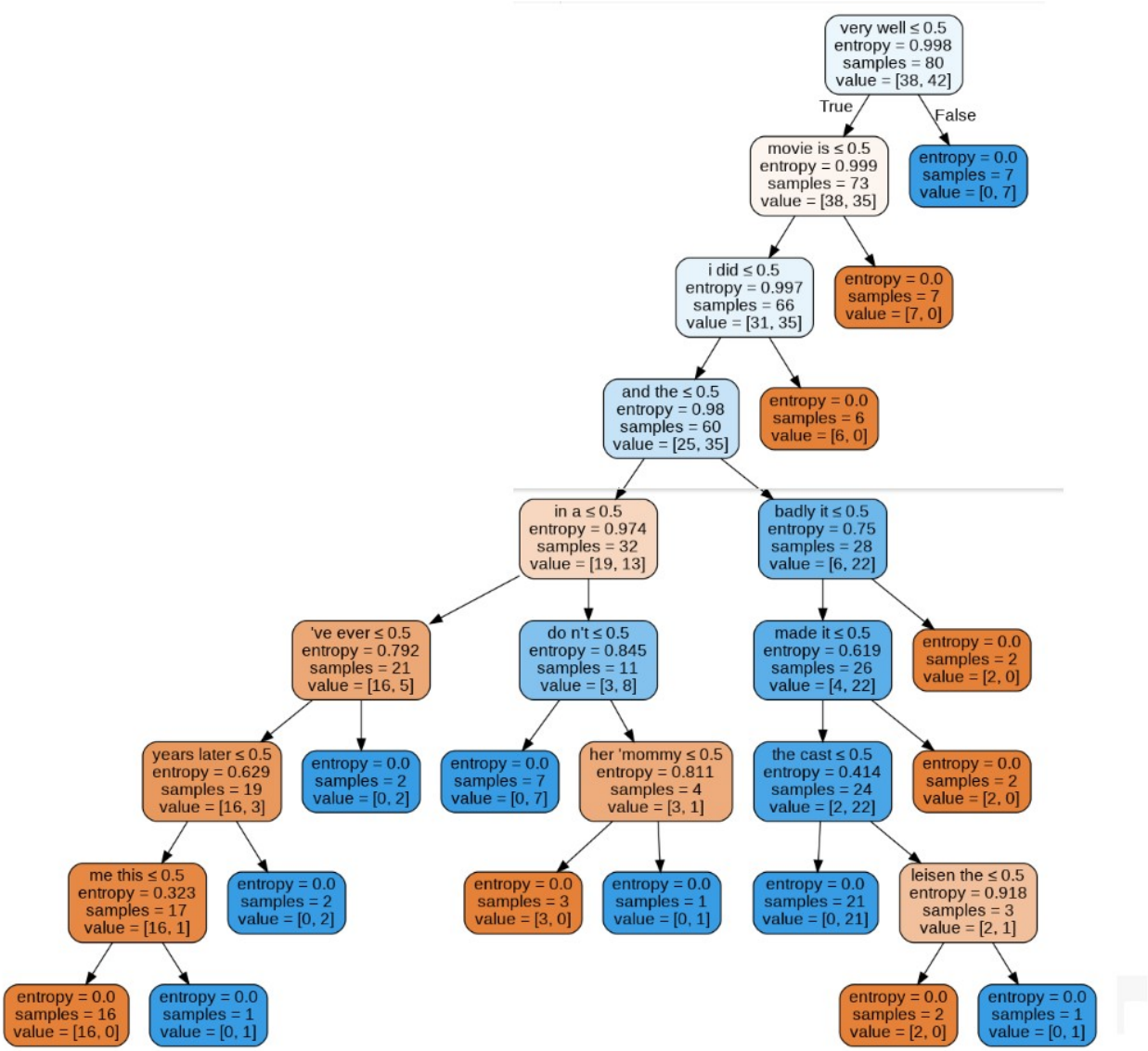


Figure 6. Decision tree