Introduction to Data Science

Homework 10: First steps of the project

4N – Negative News Neural Nets

Dan Pavlovič

Darya Pisetskaya

University of Tartu

29.11.2020

Task 1. Setting up

GitHub repository: https://github.com/dannoc96/4N-Negative-News-Neural-Nets---ML-Project

Task 2. Business understanding

Identifying your business goals

Background

Our client is TransferWise. It is an online money transfer service with headquarters in London.

All financial organizations need to do compliance investigations on their customers. Looking for adverse media also known as negative news screening on people and organizations who are their customers or potential customers.

Adverse media is all media, that shows a person in negative way. This can be connection to terrorism or any sort of crime like money laundering or tax evasion.

Looking for adverse media is expensive. This can be done by outsourcing it to other companies or in-house. To do it in-house you need to have sufficient tools to do it fast and cheap enough.

Since manual adverse media checks are slow it's very hard to go through all the data. To make it efficient this process needs to be automated.

Business goals

The goal is to automate the process of negative news screening on people and organizations. This means the articles will be classified as adverse media or non-adverse media.

Business success criteria

The business success criteria were not clearly provided by the company. I would say the success is if we lower the time of adverse media searching. This means the model has to be fast and reliable enough so it can be trusted.

Assessing your situation

Inventory of resources

Two of us are building a model (me and Darya). Kristjan Roosild is in charge of the project on TransferWise side. We have two computers, which are not ready for big data science tasks. For this we will use Google Colab free version. In case this is not enough we can use University of Tartu machines.

Requirements, assumptions, and constraints

The model has to be completed by 14.12.2020 when if will be run on test data by Kristjan Roosild. Unfortunately, TransferWise could not provide any data is it could reveal their clients.

Risks and contingencies

Our project could be delayed by lack of data. We do not know how much data we need to predict accurately enough.

Terminology:

- **Adverse media**: News that projects a person or on organization in negative way. Negative regarding criminal activities (money laundering, bribery, tax evasion, terrorism, …). The person or organization can be investigated, accused, charged or jailed.
    - Example: "*Former Maldives President Yameen charged with money laundering"*
- **Non-adverse media**: News talking about criminal activities regarding adverse media, but are not accusing anyone of doing it.
    - Example: "*Combatting money laundering and terrorist financing"*
- **Random media**: News that doesn't fall in any of previous mentioned categories.
    - Example: "*As insurance companies take over pension plans, are your payments at risk?"*

Costs and benefits

As we are students we are not paid for the work. There could only be the benefit of TransferWise using the model and lower the cost of their work. We do not have any insight in how much would that be for them.

Defining your data-mining goals

Data-mining goals

We will deliver a binary classification model, to which you will input the article and it will classify it as adverse media or non-adverse media. At the end we will present the model and show how it is built and how can it be further improved if necessary.

Data-mining success criteria

The success will be measured by F1 score. It will be assessed by Kristjan Roosild on 50 adverse media articles, 25 non-adverse media and 25 random articles. The success is if we get a score higher of 0.9 on F1 score.

Task 3. Data understanding

Gathering data

TransferWise could not provide us with any dataset, since it would disclose company's customers. We chose to search for the articles by ourselves and to label them manually.

Outline data requirements

Our data should contain 50% of adverse news articles and 50% of non-adverse and random articles. We are going to collect data URLs and Kristjan will scrape them using Scrapinghub.

Verify data availability

Scrapinghub is able to scrape most of the articles. However, some news is not scrapeable and some are protected by paywall. We are going to copy not scrapeable articles by hand and delete paywalled ones.

Define selection criteria

News article URLs will be collected from many news websites such as bbc.com and reuters.com. Articles will be found either by keywords, by names of entities mentioned in these articles or by news categories on the websites. We will also use Kaggle 'All the news' dataset, where we are going to choose articles that are neither adverse or non-adverse and label them as random. All text of the article plus the article's title are relevant for the binary classification. Name of entity and entity type can be used for further improvements such as named entity recognition.

Describing data

Our data has three separate datasets: 517 non-adverse articles, 781 adverse articles and 300 random articles from Kaggle dataset. Labelling in non-adverse and adverse datasets was done manually by people who collected articles and further quality assurance (QA) of this labelling was done by one of the other people working on this project. We intend to gather additional non-adverse articles to balance the dataset.

Random articles dataset contains 3 fields:

1. 'label' is article's category.
2. 'article' is the text of the article.
3. 'title' is the title of the article.

Both adverse and non-adverse datasets contain 10 fields:

1. 'source' is the name of the person who collected the article.
2. 'entity_name' is a name of the entity that is discussed in the article.
3. 'entity_type' specifies whether the entity is an organisation, a company or a person.
4. 'label' is article's category.
5. 'url' contains URL of the article.
6. 'title' is the title of the article.
7. 'article' is the text of the article.
8. 'explanation' is the reason why article falls into one of the categories.
9. 'full_response' is the JSON response from Scrapinghub.
10. 'accessor' is the name of the person who did QA.

Only 'article', 'title' and 'label' are used to train classification models.

Exploring data

After QA was done in adverse and non-adverse datasets, some articles were deleted and some articles were re-labelled. Then we merged all three datasets together and obtained our train dataset of 1494 articles. It has 697 adverse articles, 397 non-adverse articles and 400 random articles. This identified the need to collect 100 additional adverse articles.

Since our data is in text format, the amount of exploration is limited. We concatenated each article and its title to derive column 'text'. Average length of text is 760 words, with average length for adverse, non-adverse and random articles being 639, 685 and 1045 respectively.

Most frequent words in non-adverse articles are 'corruption', 'money' and 'said'. 'Bank', 'money' and 'said' are most frequent in adverse and 'mr',' trump' and 'said' in random. 'Money', 'mr' and 'said' and most frequent in all dataset.

These results show that our data might be a bit biased, which is explained by time period when the articles were collected. We might add words like 'said' or 'mr' in out stop words list to improve classification accuracy.

Verifying data quality

QA was done before scraping, therefore most of the articles that had problems were already detected and removed from the dataset. When looking at our data and at classification results, we discovered some badly scraped articles, articles on languages other than English and articles with wrong labels. Articles with wrong labels were re-labelled, badly scraped and non-English articles were deleted.

Task 4. Planning your project

1. Meet and align goals with Kristjan (5h – Dan and Darya)
2. Project introduction presentation (1h – Darya)
3. Gather the data (15h – Dan and Darya)
4. Learn about NLP (30h – Dan and Darya)
5. Prepare the data – repeatable (8h – Dan)

In this step we need to prepare the data for further use for modelling. First of all, we will clean the data from badly scraped articles and re-label some articles. We will merge three separate datasets together and derive column 'text' from articles and their titles. In order to use text for modelling, we need to remove stop words and punctuation, lower the case and stem the words. Some models require counts of the words in the dataset or tokenized text.

6. Make a baseline model – repeatable
    - Logistic Regression – a simple not neural baseline for classification. (10h – Dan)
    - LSTM – a reasonably simple neural baseline that was state-of-the-art some time ago. (15h – Darya)

As long as our dataset is not balanced, we are going to use Matthews correlation coefficient as metric for the binary classification. Since amount of data is already small and Kristjan has test set for final assessment, we will use cross validation to choose the best model and then send this model to Kristjan.

7. Check for mislabeled data after model evaluation (5h – Dan)

This step will help us to see where our models are failing and whether they are making the same mistakes. This might also help to identify mistakes in labelling or scraping.

8. Make a deep network model (BERT, RoBERTa) – repeatable (40h – Darya, 15h - Dan)

These models are current state-of-the-art in NLP and can work good on small datasets. Find best parameters to fine tune them.

9. Check for mislabeled data after model evaluation (3h – Dan)
10. Gather mode data if necessary and evaluate models again (6h – Dan and Darya)
11. Make ensemble out of produced models – repeatable (15h – Dan and Darya)
12. Final project presentation (2h – Dan)