Sentimental Analysis on the pharmaceutical drug reviews with Deep Learning and comparative study with ML algorithms

MSc Research Project
Data Analytics

Mobeen Shaik
Student ID: x19217277

School of Computing
National College of Ireland

Supervisor: Martin Alain

1 Introduction

The configuration manual details mentioned in this document has specifications to set up the environment in terms of both the hardware and software needs. On top of that, the document also contains the information about the code modules which states the execution flow, the libraries and packages used for development, the Integrated Development Environment tool used for implementation and so on.

1.1 Project Title:

Sentimental Analysis on the pharmaceutical drug reviews with Deep Learning and comparative study with ML algorithms.

1.2 Project Description:

The developed research model is designed to fill the gap in the drug development lifecycle which is between the pharmaceutical companies and the patients who intake the medicines. The aim of the research was to develop a model that performs the sentiment analysis specific to the review text of the drugs/medicines. On top of that, we have also compared to the results of our model with the pre-trained sentiment analysis library called Vader Sentiment Analysis.

2 Hardware/Software Configuration

Here we mentioned the hardware configuration in which we have developed the application and also the software packages that were used for the development.

2.1 Hardware Specification:

Operating System	**Windows 10**
Random Access Memory (RAM)	8 GB RAM
Processor	Intel Core i7
Processor Generation	8th Generation

Note: We have not tested the application with different other systems configurations or operating system due to the limited available for the academic research.

2.2 Software Specification:

Integrated Development Environment (IDE) – This is an environment which is for the developers to facilitate the development process in terms of organizing the code modules, using the intelli Sense feature in the IDE to auto complete the code and so on. For our research, we have used community version of PyCharm which is free for the developers. It is designed specifically for the python development.

Python Programming – The code is developed using the Python programming, which has wide range of packages supporting the development of the Artificial Intelligence (AI) programming. We have used the python version 3.9.0 for our research development.

Python Libraries – We have used the below packages for the development.

- **Keras**: open-source software library for the development of LSTM Deep Learning Model.

- **Tensorflow**: free and open-source software library for machine learning and artificial intelligence.

- **NLTK**: Natural Language Toolkit used for processing the Human language in the python programming and prepare it for model training.

- **Numpy**: is a python library that is used to perform mathematical calculations on a huge set of arrays and matrices

- **Pandas**: is a library in python programming, which is used to perform the manipulations and analysis on the data.

- **SKLearn**: is a free library in Python programming that is used for apply different machine learning models. For our research, it is used for calculating the evaluation metrics of the developed model.

- **VaderSentiment**: VADER Sentiment Analysis. VADER (Valence Aware Dictionary and sEntiment Reasoner) is a lexicon and rule-based sentiment analysis that is already pre-trained with the text from different social media text from many domains.

2.3 References:

- https://keras.io/
- https://www.tensorflow.org/
- https://www.nltk.org/
- https://scikit-learn.org/stable/
- https://pypi.org/project/vaderSentiment/
- https://www.jetbrains.com/pycharm/
- https://www.python.org/downloads/release/python-390/

3 Software Installation

Here we have defined the step-by-step process that needs to be followed in setting the application and make it run.

Step 01: Download the copy of the project source code that we have submitted in the Artifacts section of the submission.

Step 02: Place the application in the intended location to execute the same.

Step 03: As mentioned in the software and hardware specification chapter, we should’ve installed Python as well the PyCharm IDE.

Step 04: Open the source code with PyCharm IDE

Step 05: We need to set-up the project interpreter (Creating the Virtual environment) pointing to the python version 3.9.0

Step 06: Activate the created environment with the below command:

```
cd env/scripts activate
```

Step 07: Next install the required supporting packages using the PIP command which is given below:

```
pip3 install -r requirements.txt
```

Note: The requirements.txt file has information about all the packages that is required for the application to run.

Step 08: Change the path of the training set folder in the file LSTM_DrugSentimentAnalysis.py as per your system location, as the mentioned path is the absolute path; not the relative one.

Step 09: For initiating the LSTM model training, enter the below command in the PyCharm Terminal window.

```
Python LSTM_DrugSentimentAnalysis.py
```

Step 10: Once the model training is completed, we shall use the web interface for interacting with the model for prediction. Use the below command:

```
Python -m flask run
```

Note: Open the generated localhost URL in browser to interact with the LSTM model. And for testing the Vader model, using the path /vader/ in the web URL.

Step 11: Enter sample texts and click submit to view the results generated.

Note: The results could be viewed in the user interface and for more detailed trace we could use the terminal window in the PyCharm which would print all the required execution trace for better understanding of the model execution.

4 Implementation

This chapter contains the information on the dataset used for implementation and the code modules of the model.

4.1 Dataset

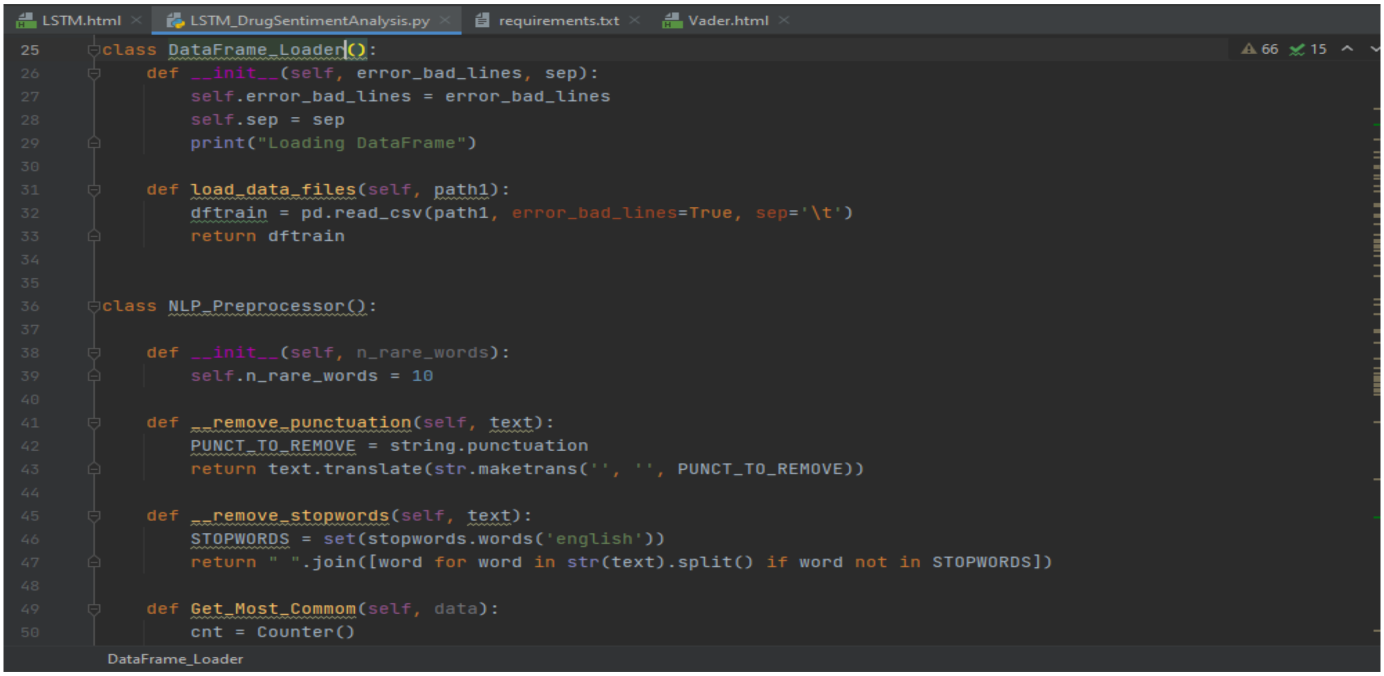
Link – https://archive.ics.uci.edu/ml/datasets/Drug+Review+Dataset+%28Drugs.com%29

The dataset has different columns, which is explained below.

- **Drug name**- Drug Name.
- **Condition**- Condition in medical terms.
- **Review**- Review provided by the patient on the medicine consumed.
- **Rating**- Rating provided by the patient on the medicine consumed.
- **Date**- Date in which review is casted
- **Useful count**- A count on people found the review as a useful one.

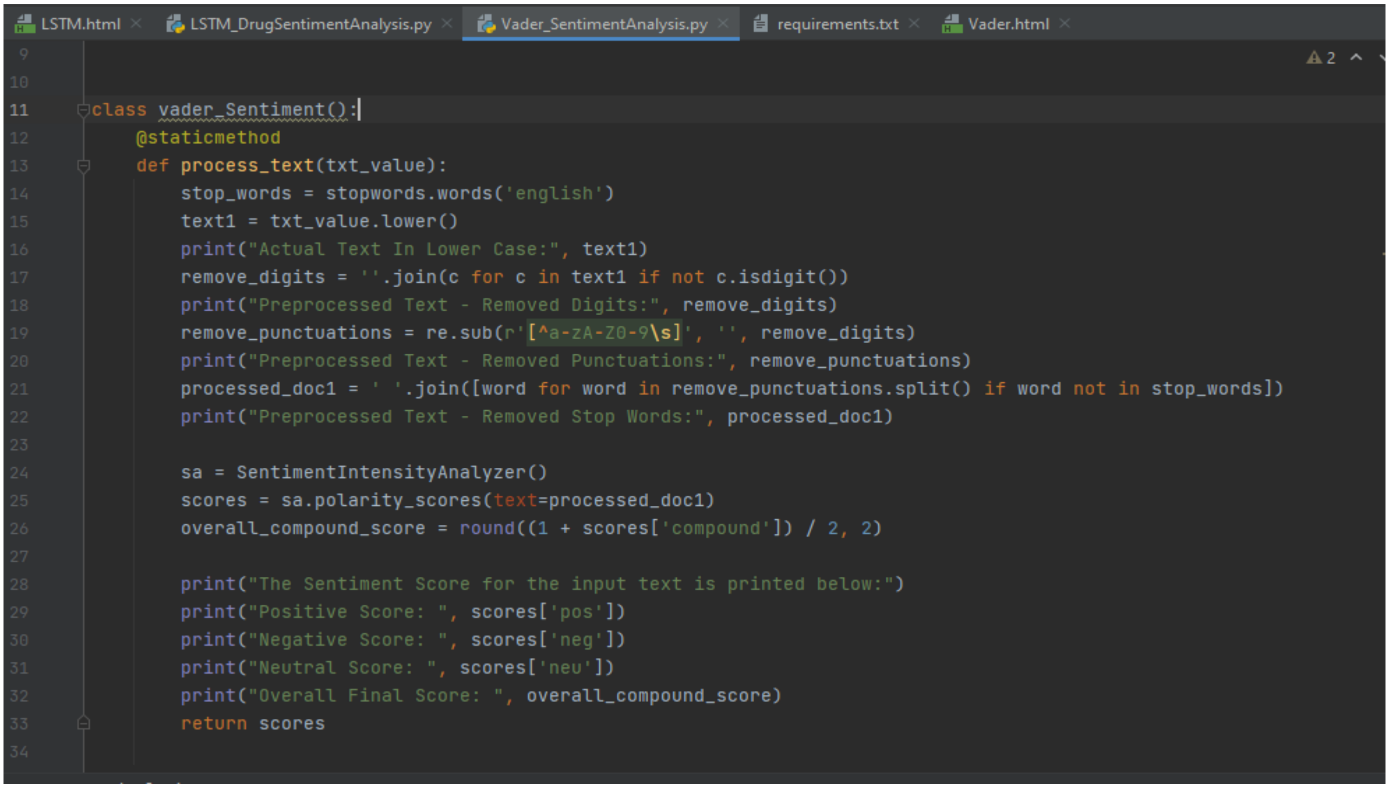
4.2 Code Implementation

LSTM Code Module – Performing the NLP operations like punctuation removal, stop words removal and so on before using the data for model training.



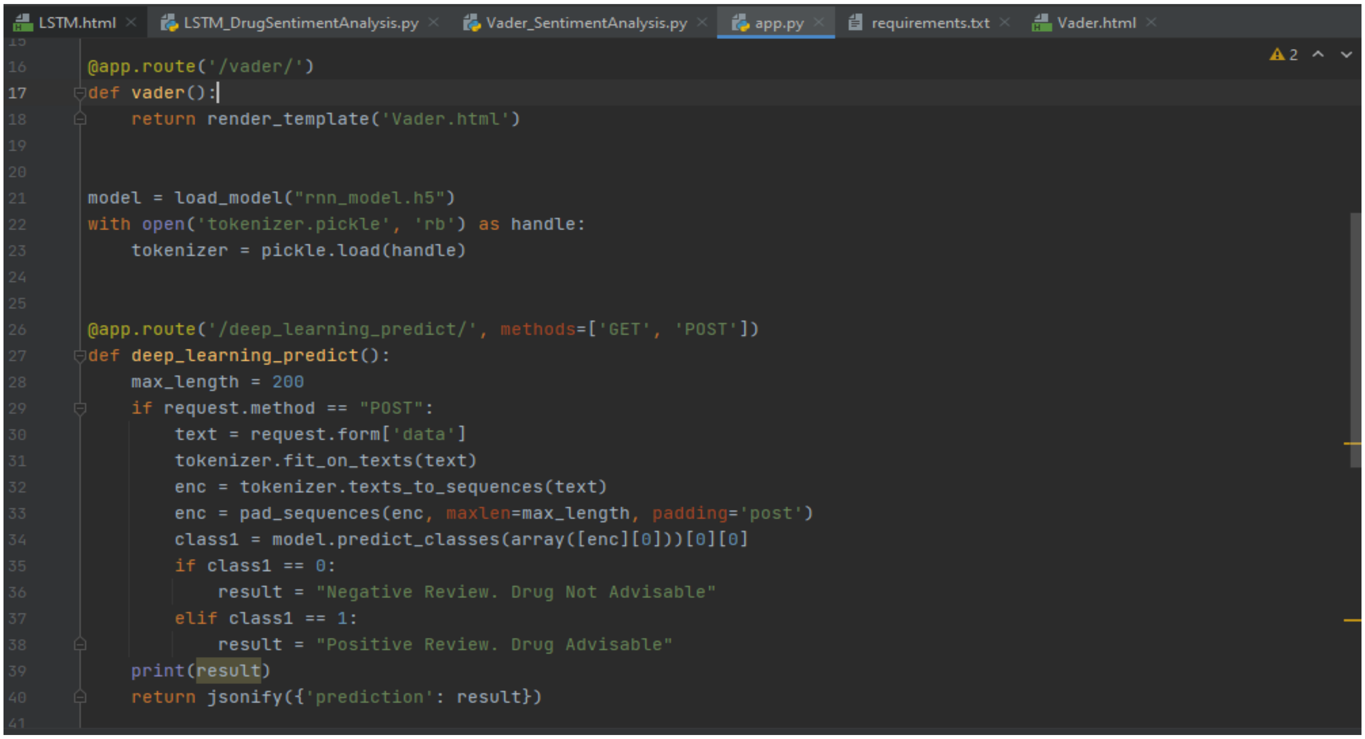
```python
class DataFrame_Loader():
    def __init__(self, error_bad_lines, sep):
        self.error_bad_lines = error_bad_lines
        self.sep = sep
        print("Loading DataFrame")

    def load_data_files(self, path1):
        dftrain = pd.read_csv(path1, error_bad_lines=True, sep='\t')
        return dftrain


class NLP_Preprocessor():

    def __init__(self, n_rare_words):
        self.n_rare_words = 10

    def __remove_punctuation(self, text):
        PUNCT_TO_REMOVE = string.punctuation
        return text.translate(str.maketrans('', '', PUNCT_TO_REMOVE))

    def __remove_stopwords(self, text):
        STOPWORDS = set(stopwords.words('english'))
        return " ".join([word for word in str(text).split() if word not in STOPWORDS])

    def Get_Most_Commom(self, data):
        cnt = Counter()
```

Vader Sentiment Code Module – Using the existing package for calculating the sentiment score:



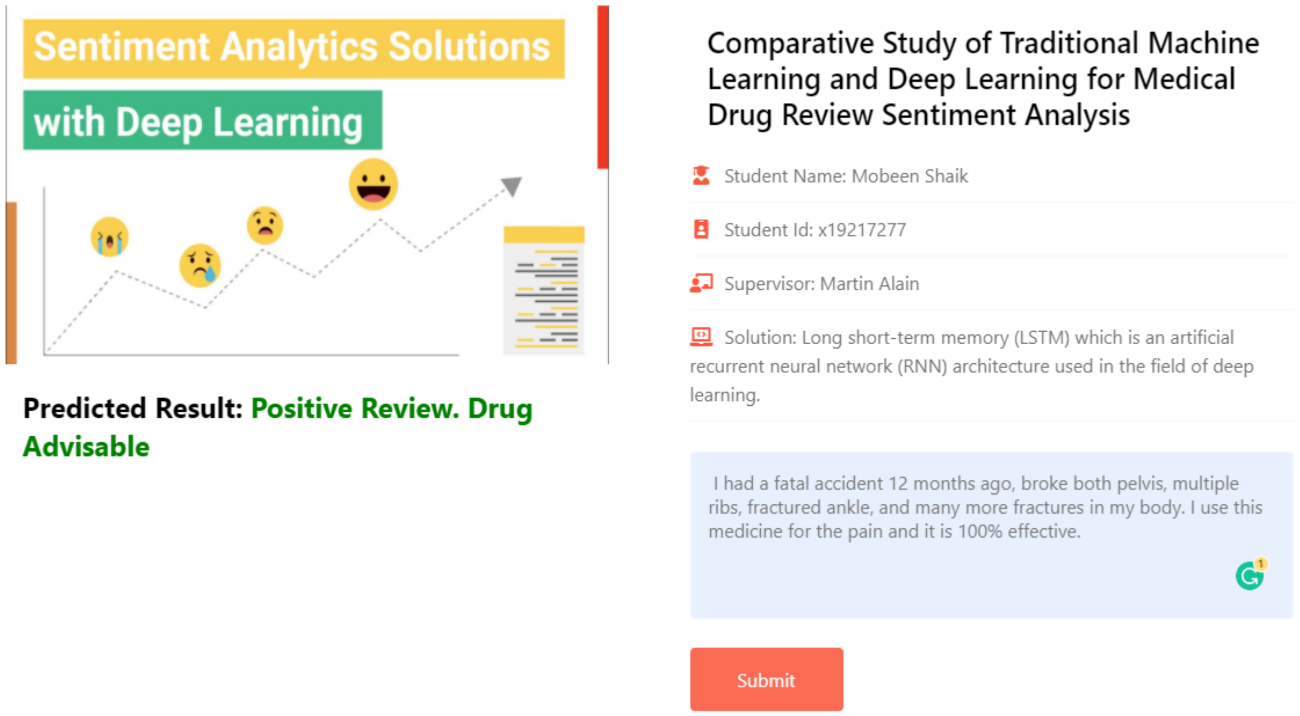
```python
class vader_Sentiment():
    @staticmethod
    def process_text(txt_value):
        stop_words = stopwords.words('english')
        text1 = txt_value.lower()
        print("Actual Text In Lower Case:", text1)
        remove_digits = ''.join(c for c in text1 if not c.isdigit())
        print("Preprocessed Text - Removed Digits:", remove_digits)
        remove_punctuations = re.sub(r'[^a-zA-Z0-9\s]', '', remove_digits)
        print("Preprocessed Text - Removed Punctuations:", remove_punctuations)
        processed_doc1 = ' '.join([word for word in remove_punctuations.split() if word not in stop_words])
        print("Preprocessed Text - Removed Stop Words:", processed_doc1)

        sa = SentimentIntensityAnalyzer()
        scores = sa.polarity_scores(text=processed_doc1)
        overall_compound_score = round((1 + scores['compound']) / 2, 2)

        print("The Sentiment Score for the input text is printed below:")
        print("Positive Score: ", scores['pos'])
        print("Negative Score: ", scores['neg'])
        print("Neutral Score: ", scores['neu'])
        print("Overall Final Score: ", overall_compound_score)
        return scores
```

App.py – Flask App Interface:



```python
@app.route('/vader/')
def vader():
    return render_template('Vader.html')


model = load_model("rnn_model.h5")
with open('tokenizer.pickle', 'rb') as handle:
    tokenizer = pickle.load(handle)


@app.route('/deep_learning_predict/', methods=['GET', 'POST'])
def deep_learning_predict():
    max_length = 200
    if request.method == "POST":
        text = request.form['data']
        tokenizer.fit_on_texts(text)
        enc = tokenizer.texts_to_sequences(text)
        enc = pad_sequences(enc, maxlen=max_length, padding='post')
        class1 = model.predict_classes(array([enc][0]))[0][0]
        if class1 == 0:
            result = "Negative Review. Drug Not Advisable"
        elif class1 == 1:
            result = "Positive Review. Drug Advisable"
    print(result)
    return jsonify({'prediction': result})
```

LSTM Model Result – Web Interface:



Predicted Result: Positive Review. Drug Advisable

Comparative Study of Traditional Machine Learning and Deep Learning for Medical Drug Review Sentiment Analysis

Student Name: Mobeen Shaik

Student Id: x19217277

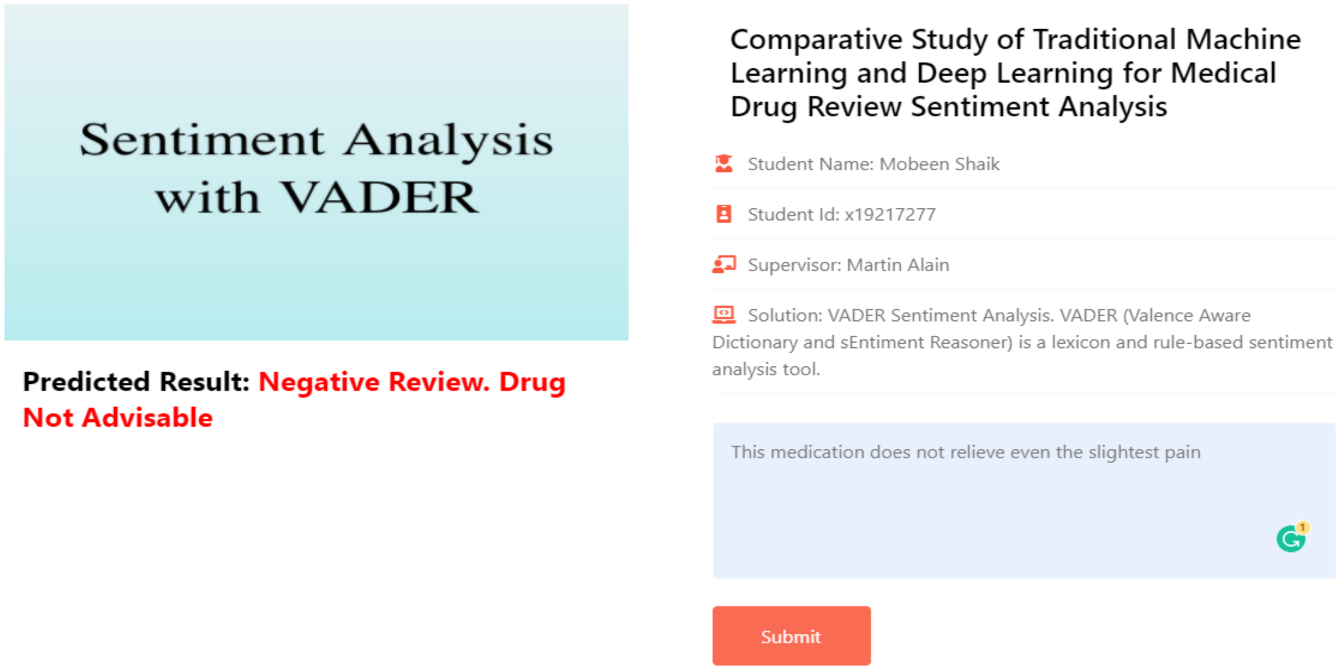
Supervisor: Martin Alain

Solution: Long short-term memory (LSTM) which is an artificial recurrent neural network (RNN) architecture used in the field of deep learning.

I had a fatal accident 12 months ago, broke both pelvis, multiple ribs, fractured ankle, and many more fractures in my body. I use this medicine for the pain and it is 100% effective.

Submit

Vader Sentiment Result – Web Interface:



Predicted Result: Negative Review. Drug Not Advisable

Comparative Study of Traditional Machine Learning and Deep Learning for Medical Drug Review Sentiment Analysis

Student Name: Mobeen Shaik

Student Id: x19217277

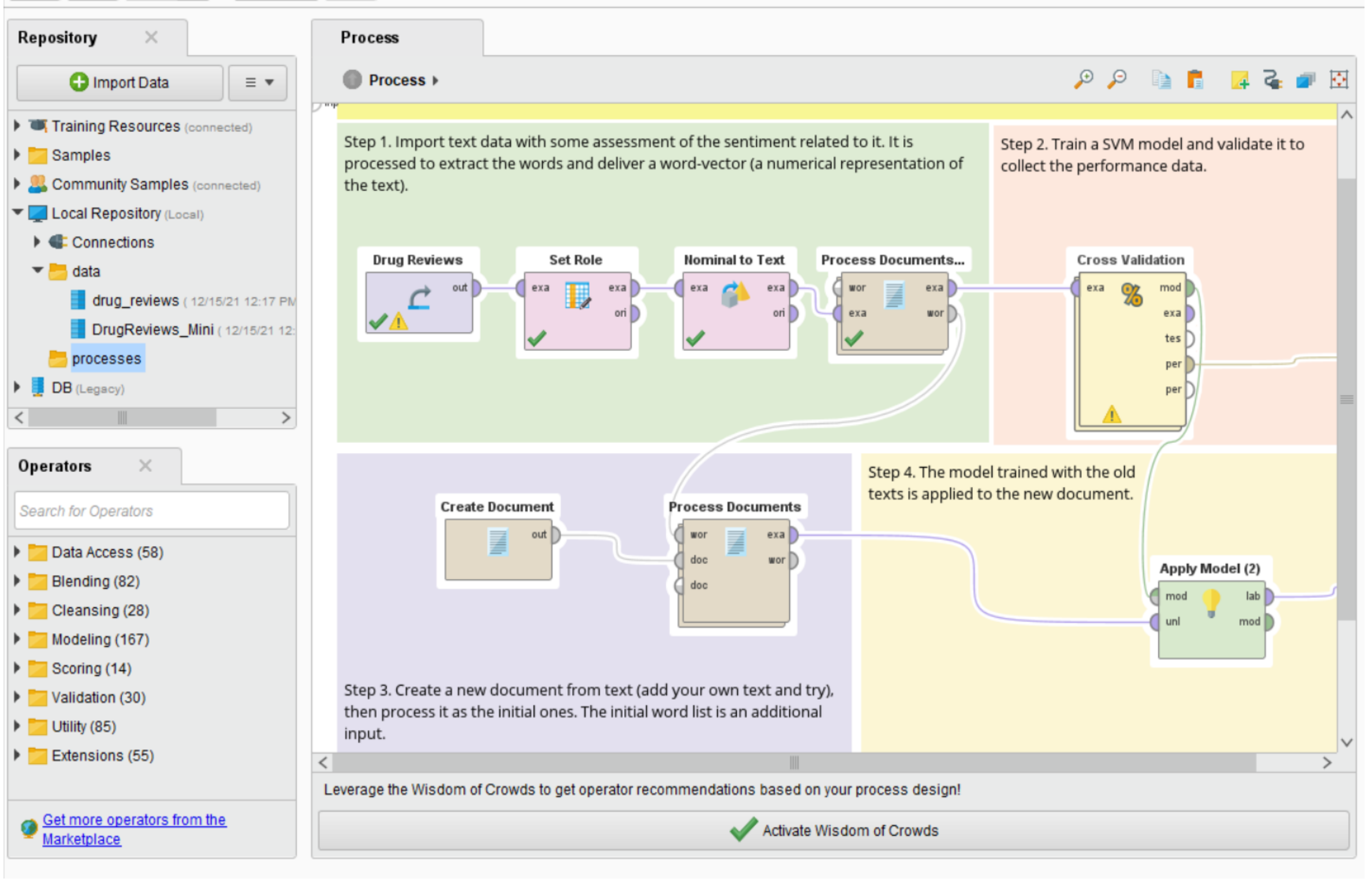
Supervisor: Martin Alain

Solution: VADER Sentiment Analysis. VADER (Valence Aware Dictionary and sEntiment Reasoner) is a lexicon and rule-based sentiment analysis tool.

This medication does not relieve even the slightest pain

Submit

RapidMiner Process: RapidMiner is a data science software platform that is used to apply the machine learning and deep learning programming without the code modules. Here we have used it for comparing the developed LSTM deep learning model with other machine learning techniques.



Step 1. Import text data with some assessment of the sentiment related to it. It is processed to extract the words and deliver a word-vector (a numerical representation of the text).

Step 2. Train a SVM model and validate it to collect the performance data.

Step 3. Create a new document from text (add your own text and try), then process it as the initial ones. The initial word list is an additional input.

Step 4. The model trained with the old texts is applied to the new document.

Leverage the Wisdom of Crowds to get operator recommendations based on your process design!

Result from RapidMiner for the SVM Machine Learning Model:

Classification error	Value	Standard deviation
accuracy	83.3%	±2.5%
classification_error	16.7%	±2.5%
AUC	92.7%	±1.3%
precision	81.1%	±3.1%
recall	87.2%	±5.2%
f_measure	83.9%	±2.4%
sensitivity	87.2%	±5.2%
specificity	79.3%	±4.8%