The Lightweight IBM Cloud Garage Method for Data Science

Architectural Decisions Document Template

1. Architectural Components Overview

1.1. Data Source

1.1.1. Technology Choice

The datasource for this project is the UC Irvine Machine Learning Repository (https://archive.ics.uci.edu/ml/index.php).

The dataset was constructed as part of a research project by Surya Kallumadi (Kansas State University) and Felix Grässer (Technical University of Dresden). The result of the research was later published as "Aspect-Based Sentiment Analysis of Drug Reviews Applying Cross-Domain and Cross-Data Learning" in "Proceedings of the 2018 International Conference on Digital Health (DH '18)".

1.1.2. Justification

The dataset provides patient reviews on specific drugs along with related conditions and a 10 star patient rating reflecting overall patient satisfaction. The data was obtained by crawling the online pharmaceutical review site drugs.com.

The dataset was chosen as it is freely available and present review data on medicine within a rating system. It is therefore useful to create a classifier to quantity medicine reviews into a rating system.

1.2. Data Integration

1.2.1. Technology Choice

Batch data processing using the Pandas libraries from Python in a Jupiter Notebook, to read, clean and modify the datafiles. The vectorisation of the text descriptions are done with the TfidVectorizer from the scikit-learn python library.

1.2.2. Justification

Python is used as the programming language due to its many useful and powerful libraries in data processing (Numpy, Pandas), Machine Learning (scikit-learn), Natural Language Processing (nltk) and Deep Learning (TensorFlow/Keras). Python is also among the most used programming languages in Data Science - making it easy and cheaper to maintain for other Data Scientists and is an easy-to-understand programming language. The Jupyter Notebook is used as it allows for writing the code in a text document format which makes it easy to follow and

understand for the purpose of this course. The seaborne library is used for the standard plotting of the data and matplotlib is used when more specific plots are required for the static figures. Both are state-of-the-art Python libaries. All of the ETL script is based on Pandas Dataframes that are easy to manipulate, contains a lot of preprogrammed statistics/manipulations and allow relational access to the data. All allows for a simple and cost effective process.

The data is already split into a train (75 %) and test (25 %) data sets stored in two .tsv (tab-separated-values) files, respectively. These files can be read as Pandas data frames using the API for csv-files which makes it easy to manipulate and get statistics about the data. The cleaning of the data includes cleaning of NaN values, cleaning of empty strings in comments descriptions and problems related to return lines which have to be deleted in the text.

The feature used for the analysis will be the written text reviews of the medicine. The data contains three columns with text descriptions of benefits, side effects and comments, respectively. To include as much text and information as possible, the three text description columns are combined to a combined review column.

The TfidVectorizer from the powerful and popular scikit-learn python package is used to translate the text into a feature space. This method of translation to a vector is used a lot in literature.

As the distribution of text description length is very similar for all rating, this feature is not expected to have any impact on the analysis and is therefore not included.

The labels are set as the final review score (1-10) but transformed to a (1-3) score. One problem about this rating system with 1-10 points is that the classification between the adjacent scores such as 8 and 9. If the drug works well with little to no side effects, the patient might give 8, 9, or 10 without too much thought about it. Within a particular score region, it is entirely subjective and therefore the algorithm is expected to perform better with a smaller range of scores. As a result, we can mitigate this problem by categorizing into 3 groups:
- 1-3 for bad
- 4-7 for medium
- and 8-10 for good results.
With this method, we reduce the noise of the data.

The data also contain columns with scores about effectiveness and side effects of the medicine. These scores are given as strings as e.g. no, mild, moderate etc. - both column only contains 5 unique strings and for that reason they are encoded to scores from 0-4. These scores will be used in a pre-analysis model to set a baseline for the comment text analysis models. The columns for “DrugName” and “Condition” are one-hot encoded instead of label encoded because of sklearn (machine learning packages) implementations that gives meaning to the encoded numbers.

All these data processing steps was done with the state-of-the-art Pandas dataframs.

1.3. Data Repository

1.3.1. Technology Choice

The data is stored in standard .tsv files and the code in Jupiter notebooks in a git repository.

1.3.2. Justification

Data is already stored in .tsv format from the source and can easily be read and manipulated with the Panda API in Python. This format does not cause any implications for the project in this stage of the project. A git repository is used for version control which allows more people to work on the project in a structured manner. The Jupiter notebooks allows for easy access to code embedded in a text document structure. Its is optimal for showing the code to fellow peers and for other people to easily understand the code. The code can also easily be run in sections.

1.4. Discovery and Exploration

1.4.1. Technology Choice

Seaborne/matplotlib python libraries for plotting of static graphs and statistics. Pandas for illustration of dataframes. Scikit-learn and Pandas for calculation of metrics.

1.4.2. Justification

Matplotlib and Seaborne are state-of-the-art packages for visualisation of static graphic in Python. Seaborne has a high-level API with minimal effort for beautiful and presentable figures. Matplotlib is used when a more low-level visualisation tool is needed.

The needed figures are figures for statistics of distribution of scores (histogram) to get an idea about the skew of the dataset. A figure (histogram and bar plot) for then correlation between length of text descriptions and score is made to decide if this feature is worth including in the analysis. Pandas data frames are used to present the data at all steps of the analysis. Panda data frames are well structured with column and row names.

Using scikit-learn and pandas, all state-of-the-art metrics are supported. Metrics are chosen as Accuracy, Recall and F1-score. F1-score is chosen for the overall model performance for all models as is common practice in machine learning. The score is compared for different models to select the most accurate one.

Only model performance need to be shared with stakeholders.

1.5. Actionable Insights

1.5.1. Technology Choice

Python was chosen as the programming language.
The nltk package was used for natural language processing
The scikit-learn package was used for machine learning models.
The Keras high-level API of TensorFlow was used for the deep learning model.

1.5.2. Justification

Python with has become the de-factor standard for open source based data science for a long period of time. Python is much cleaner than for e.g. R and Python skills are widely available in Data Science reducing the cost of the programs. The Sci-kit learn module nicely groups all necessary machine learning algorithms together and is nicely written so everything sticks together in a pipeline where you can easily tweak and change the model. Linear Algebra is done through the NumPy Library. The NLTK package of Python can be used for a variety of natural language processing operations.

TensorFlow is one of the most widely used DeepLearning frameworks. In its core, it is a linear algebra library supporting automatic differentiation. TensorFlows python driven syntax is relatively complex. Therefore, Keras provides an abstraction layer on top of TensorFlow. The models can be run in parallel through e.g. Apache Spark and supports GPUs as well. TensorFlow/Keras skills are still relatively rare and applications can be expensive to create and maintain. It is however more common that the alternatives.

Coding skills include machine learning, data analysis and deep learning. All skills are acquired in the course.

The nltk package is used to reduce the word vector space by removing unnecessary stopwords and punctuations in the review text fields + Tokenisation and lemmatisation of the words.

In a bag-of-word model, which is the model we will use, we treat words in no orders like pouring all of the words into a bag. This means that we consider the frequency of word appearance instead of word order. There are several way to count word frequency but we will use the most basic one: count vectorization. It has been shown that Multimodal Navie Bayes classifier works well within text classification and is also common amongst spam email detection. Everything was implemented in the scikit-learn framework.

The feature vector extracted from the text with TfidfVectorizer (scikit-learn) was used with logistic regression in a machine learning pipeline to predict the review score.

Lastly a TensorFlow/Keras based Deep Learning network was used for training on the text fields.

1.6. Applications / Data Products

1.6.1. Technology Choice

Python script that can be implemented with a UI for ease of use

1.6.2. Justification

Python script to take a single or a bunch of test subject comments and convert them to a score from 1-3. Can be used to quantify the comments from human test object of new medicine. Its enough to make it work as batch processing. The UI is outside the scope of this project at this time.