Machine Learning Capstone Report

Capstone Report

Linggih Saputro

November 3rd, 2017

Domain Background

The project is about classifying clinically actionable genetic mutations from Kaggle. This competition is actually in precision medicine, which about the genetic testing of cancer. Most of the data is based on the text-based clinical literature, so this will be a combination of text analysis or a natural language processing problem with a classification problem of semisupervised learning, which is half supervised and half unsupervised learning and it divides the data into labeled and unlabeled data sets (Chapelle et al., 2009)

Natural Language Processing (NLP) is the study of mathematical and computational modeling of various aspects of language and the development of a wide range of systems. (Joshi AJ.,1991)

This problem motivates me the most because I study in bioinformatics and I have a particular interest in precision medicine. I hope by taking this project I will get more insights about the domain and more opportunities to explore.

Joshi, A. J. (1991). Natural language processing. *Science*, 1242-1249.

Chapelle, O., Scholkopf, B., & Zien, A. (2009). Semi-supervised learning (chapelle, o. et al., eds.; 2006)[book reviews]. *IEEE Transactions on Neural Networks*, *20*(3), 542-542.

Problem Statement

The main goal in this project is to eliminate the manual work of annotating the classes of the cancer based on evidence of the genetic mutations from the text-based clinical literature. Because of this problem is related to natural language processing as they have a lot of text-based literature, it is a quantifiable by tokenizing and weighing the words.

This is a classification problem because the main goal is to classify the text into its classes. The input is a table which contains columns with string elements and the cancer class is integer. The expected output is a probability for each of the different classes. Like the following format:

```
ID,class1,class2,class3,class4,class5,class6,class7,class8,class
9
0,0.1,0.7,0.05,0.05,0.1,0,0,0,0
1,0.7,0.1,0.05,0.05,0.1,0,0,0,0
2,0.05,0.05,0.1,0.7,0.1,0,0,0,0
3,0,0,0,0,0.05,0.05,0.1,0,7,0,1
etc.
```

Datasets and Inputs

https://www.kaggle.com/c/msk-redefining-cancer-treatment/data

In this competition you will develop algorithms to classify genetic mutations based on clinical evidence (text). There are 3321 samples on the training dataset and 5668 samples on the testing dataset.
There are nine different classes a genetic mutation can be classified on. However, there is more class 7 then the rest.

This is not a trivial task since interpreting clinical evidence is very challenging even for human specialists. Therefore, modeling the clinical evidence (text) will be critical for the success of your approach.

Both, training and test, data sets are provided via two different files. One (training/test_variants) provides the information about the genetic mutations, whereas the other (training/test_text) provides the clinical evidence (text) that our human experts used to classify the genetic mutations. Both are linked via the ID field.

Therefore the genetic mutation (row) with ID=15 in the file training_variants, was classified using the clinical evidence (text) from the row with ID=15 in the file training_text

- training_variants - a comma separated file containing the description of the genetic mutations used for training. Fields are ID (the id of the row used to link the mutation to the clinical evidence), Gene (the gene where this genetic mutation is located), Variation (the aminoacid change for this mutations), Class (1-9 the class this genetic mutation has been classified on)

- training_text - a double pipe (||) delimited file that contains the clinical evidence (text) used to classify genetic mutations. Fields are ID (the id of the row used to link the clinical evidence to the genetic mutation), Text (the clinical evidence used to classify the genetic mutation)
- test_variants - a comma separated file containing the description of the genetic mutations used for training. Fields are ID (the id of the row used to link the mutation to the clinical evidence), Gene (the gene where this genetic mutation is located), Variation (the aminoacid change for this mutations)
- test_text - a double pipe (||) delimited file that contains the clinical evidence (text) used to classify genetic mutations. Fields are ID (the id of the row used to link the clinical evidence to the genetic mutation), Text (the clinical evidence used to classify the genetic mutation)
- submissionSample - a sample submission file in the correct format
- stage2_test_variants - a comma separated file containing the description of the genetic mutations used for training. Fields are ID (the id of the row used to link the mutation to the clinical evidence), Gene (the gene where this genetic mutation is located), Variation (the aminoacid change for this mutations)
- stage_test_text - a double pipe (||) delimited file that contains the clinical evidence (text) used to classify genetic mutations. Fields are ID (the id of the row used to link the clinical evidence to the genetic mutation), Text (the clinical evidence used to classify the genetic mutation)

Unlabeled testing data is provided by Kaggle, I will split the data into training and testing data within the first training data provided. Then I will use the testing data provided to do another test.

Solution Statement

The solution I propose is to use the scikit-learn module in NLP to tackle this. Then after all the data is transformed from text to array, the data will be splitted to training and testing data. I will use XGBoost to do the classification. XGBoost is a tree boosting method which approximate the tree learning based on sparsity-aware algorithm and weighted quantile sketch (Chen and Guestrin, 2016).

Chen, T., & Guestrin, C. (2016, August). Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*(pp. 785-794). ACM.

Benchmark Model

For the benchmark model, simple decision tree will be used to compare it with xgboost

Decision tree is good for small datasets, but it was proven that there is no class 7 from the predictions of training data and it will severe the result.

Evaluation Metrics

(approx. 1-2 paragraphs)

I will implement the classification metrics from sklearn.metrics to measure the classification performance, specifically the log loss. Log loss can be used in logistic regression and can also be extended into the neural networks application. The competition also use this multi-class log loss. The formula for the log loss is:

$$-log\ P(yt|yp) = -(yt\ log(yp) + (1 - yt)\ log(1 - yp))$$

Project Design

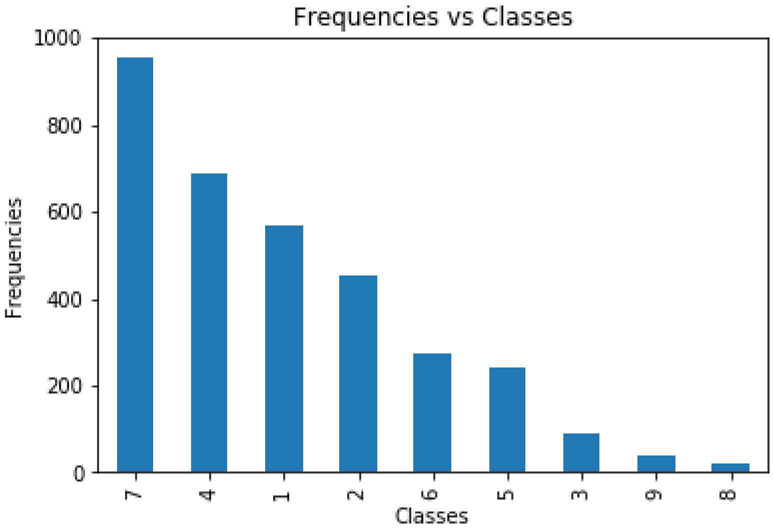
The theoretical workflow will start with the input data.

1. Preprocessing Input data
    a. For the variations and genes of the cancer, I will use one-hot encoding for the decision tree classifier and
    b. Change the input text and transform it into vector using HashingVectorizer. I use Hashing Vectorizer because vocabulary can be enormous so by using this it will convert the hash of the words to integers. The downside is this is a one way function, so encoding it back to word is impossible.
2. Classification using XGBoost and parameters tuning, ultimately for the multi:softmax.
3. Performing the Evaluation Metrics using the mlogloss in the XGBoost parameters.

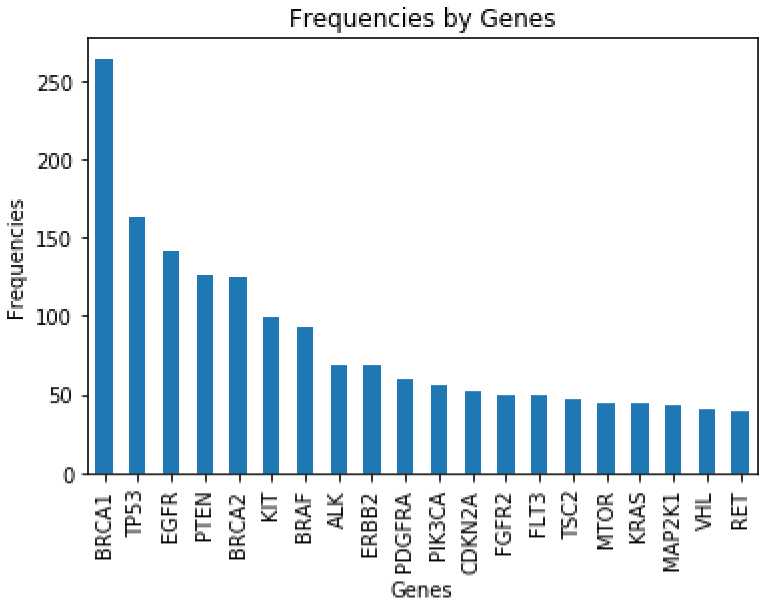
Analysis

Data Exploration

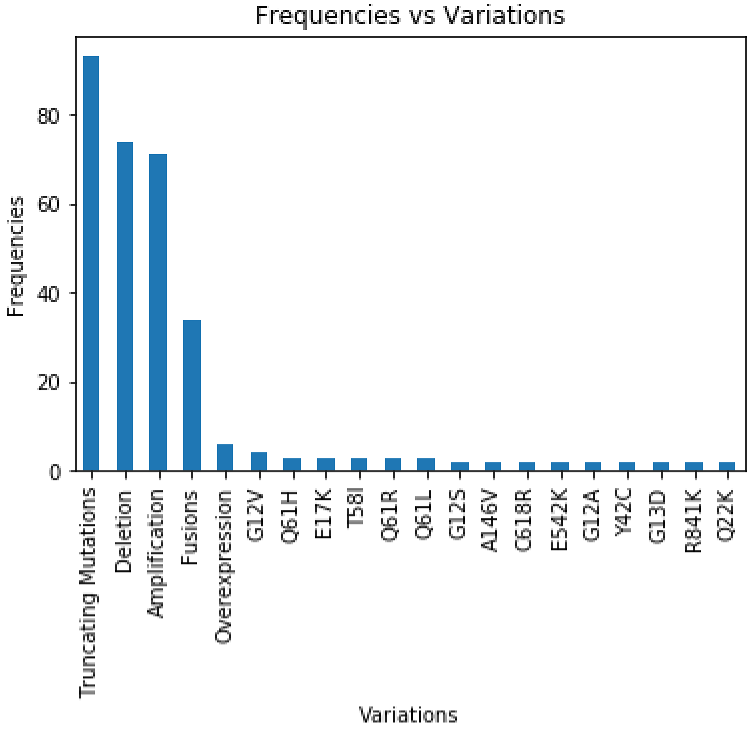
The frequency of classes from the data is skewed to the Class 7. This disproportionate weight on Class 7 will make the Class 8 and Class 9 harder to be classified unless there are specific variations and genes towards it.



The data has BRCA1 as the most frequent genes appears in the text, this might affect the weighting process.



The four most common variations in the dataset are truncating mutation, deletion, amplification and fusions.



Results

<u>Algorithms and Techniques</u>

The dataset is taken from biomedical literature, so it has the template like a scientific journal and needs to be ‘clean’ before the classification process. The common English words and some words like ‘‘figure’, which is used to mention a plot or a scheme, need to be omitted.

After cleaning the datasets, the most common words are compared to see the insight. There are specific words come up such as, mutations, samples, etc.

```
Doc  0
[('cyclin', 90), ('cdk10', 81), ('ets2', 54), ('expression', 43), ('1', 42), ('protein', 38), ('levels', 35),
('star', 33), ('fam58a', 33), ('2', 32)]

 Doc  1
[('c-cbl', 153), ('mutations', 66), ('lung', 56), ('egfr', 43), ('mutation', 43), ('1', 41), ('met', 33),
('samples', 33), ('cancer', 32), ('5', 29)]
```

After the cleaning, tokenization is the next step because in dealing with text data, it is crucial to convert the text into something that the computer understands best, which is numbers.
In this project, Term Frequency-Inverse Document Frequency (TF-IDF) is used to calculate values for each word in a document through an inverse proportion of the frequency of the word in a particular document to the percentage of documents the word appears in (Ramos, J.). Tf-idf is required because frequent terms would shadow rarer frequencies, which might add interesting result.

$$\text{tf-idf}(\text{t,d}) = \text{tf}(\text{t,d}) \times \text{idf}(\text{t}).$$

The next step is to build features, because it will give a flexibility to choose some good features derived from the dataset. In here, factorization, tfidf and truncated singular value decomposition (SVD).

<u>Benchmark result</u>
The decision tree predictions does not even count the class 7

Conclusion

Xgboost can be applied as a solution to this problem and it is better than the standard decision tree.