Personalized Medicine: Redefining Cancer Treatment

-- from kaggle competition

Domain Background[1]

A lot has been said during the past several years about how precision medicine, there have been a lot of research in this area, among them pathological diagnosis is still gold standard in clinical practice[2], in recent years, the development of genetic technology has brought major influence on the ways diseases like cancer are treated, for example, the science of “pharmacogenomics” was proposed, which identifies individuals who, based on their genotype information, will respond to a specific therapy [3]. But this is only partially happening due to the huge amount of manual work still required, while the collected data is increasing dramatically, the development of related analysis and prediction methods is still slow-moving, that is why NIPS is holding this competition and the problem this capstone project dedicates to resolve.

Problem Statement[1]

Once sequenced, a cancer tumor can have thousands of genetic mutations. But the challenge is distinguishing the mutations that contribute to tumor growth (drivers) from the neutral mutations (passengers).

Currently this interpretation of genetic mutations is being done manually. This is a very time-consuming task where a clinical pathologist has to manually review and classify every single genetic mutation based on evidence from text-based clinical literature.

Based on that, the aim of this capstone project is to develop a Machine Learning algorithm that, using this knowledge base as a baseline, automatically classifies genetic variations.

Datasets and Inputs

The datasets come from the kaggle competition.

File descriptions[4]

* training_variants(24.45 KB) - 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(59.9 MB) - 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(47.47 KB) - 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(98.98 MB) - 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)

Solution Statement

Given the labeled training data, the solution is basically a supervised learning process, to get a better result , we will use the assembling method which includes three major steps.

First, feature engineering. The main idea is to transform the free text into features by applying some common NLP techniques such as Bag of words, TF-IDF, etc. After that, the data of Gene and Variation will be encoded to the same format as the text features, the two kind of features are then to be merged according to IDs.

Second, generating first level models. Several basic ML methods will be applied, the prediction results will be stored for the learning method of next level. We should notice that it’s a process of evaluating the effects of different methods, the final combination of first level models will be decided based on their performance.

Third, second level learning. It’s the process of ensemble learning, XGBoost method will be applied on the prediction results produced by the first level training, than the final predictive model will be trained.

Benchmark Model

The problem is quite new and no existing solution can be found until now, we will choose most pervasive applied methods as the benchmark, specifically, /bag of words/ and /Logistic regression/ are the benchmarks of features extraction from text and the predictive model training separately.

Evaluation Metrics

The performant will be evaluated by Multi Class Log Loss(MCLL) between the predicted probability and the observed target.
The metric can be described mathematically like this,

$$logloss = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{M} y_{i,j} log(p_{i,j})$$

Where N is the number of observations, M is the number of class labels, $y_{i,j}$ is 1 if observation i is in class j and 0 otherwise, and $p_{i,j}$ is the predicted probability that observation i is in class j [5].

As the equation illustrates, MCLL quantifies the accuracy of a classifier by penalising false classifications, this is quite suitable to this problem, because medicine prediction is low false tolerant, especially for clinical decision. Another benefit of this metric is that it uses probability rather than boolean values, which makes the result even more accurate.

Project Design

1.Feature engineering

The clinical evidence data is presented in the type of text, basic idea is to parse the text into words, several techniques will be evaluated in this step.

2.1 Bag Of Words

The first and benchmark model is /Bag of Words/, which aims to represent features by counting the occurrence of each word, the corpus of document can be transferred into a matrix with one row per document and one column per word occurring in the corpus[6].

However, after testing in in the training datasets, we can find that features extracted by this method are too high in dimension. So the follow up step is to reduce dimension, there is a related package in Scikit-learn called ‘decomposition’ , it includes a lot of reduction techniques , among them TruncatedSVD is appropriate in this case , because it works efficiently with sparse matrices and it can transform transforms such matrices to a “semantic” space of low dimensionality[7].

2.2 TF-IDF

In the documents of clinical evidence, some words (e.g. “the”, “is”, “a”) appears a lot but carries very little meaningful information about the actual contents of the documents. If the direct count data is feed directly to a classifier, those very frequent terms would shadow the frequencies of rarer yet more useful terms[6].

To address this problem, it is very common to use the tf-idf transform to re-weight the count features. Tf means term-frequency while tf–idf means term-frequency times inverse document-frequency. Scikit-learn also provides related class called “TfidfVectorizer” that combines all the options of vectorization and transformation in a single model.

Also, a reminder that the dimension reduction method is still needed , because TF-IDF only modifies the weights and has nothing to do with dimension.

2.3 Features merging

Besides text dataset, we also have the variations data which includes Gene and Variation information. To better feed the model training, we should encode these data and combine it with and TF-IDF separately, the encoding tool we use is LabelEncoder provided by sklearn, which can transform non-numerical labels to numerical labels[8]

3. Ensemble Learning

Now that we have training data with labels and formalized features , the next step would be the process of supervised learning, according to recent ML competition results, model ensembling is proved to be a very powerful technique, as it increases accuracy dramatically and reduces the generalization error. So this project will follow the process of ensembles learning, the crucial steps are listed as follows.

3.1 Generating first level models

We choose five models as the basic learning models. These models can be conveniently invoked via the Sklearn library and are listed as follows.

1. Logistic Regression
2. Random Forest
3. AdaBoost
4. Gradient Boosting
5. Support Vector Machine

The 5 base classifiers will be trained separately on the original training and test data, different combinations of these models will be evaluated by the correlation of their prediction results, the lower the better. Then we will get our first level predictions results, which are the input of our next classifier.

3.2 Second level learning via XGBoost

Here we choose the hottest library for boosted tree learning model, XGBoost[9].It was build to optimize large-scale boosted tree algorithms. We call an XGBClassifier provided by the XGBoost packages and feed first-level train and test results to it, finally we will get the ensembled prediction model.

Reference

[1] https://www.kaggle.com/c/msk-redefining-cancer-treatment

[2] Verma M. Personalized Medicine and Cancer. *Journal of Personalized Medicine*. 2012;2(1):1-14. doi:10.3390/jpm2010001.

[3]Schroth W., Goetz M.P., Hamann U., Fasching P.A., Schmidt M., Winter S., Fritz P., Simon W., Suman V.J., Ames M.M., Safgren S.L., Kuffel M.J., Ulmer H.U., Boländer J., Strick R., Beckmann M.W., Koelbl H, Weinshilboum R.M., Ingle J.N., Eichelbaum M., Schwab M., Brauch H. Association between CYP2D6 polymorphisms and outcomes among women with early stage breast cancer treated with tamoxifen. JAMA. 2009;302:1429–1436. doi: 10.1001/jama.2009.1420.

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

[5] https://www.kaggle.com/wiki/MultiClassLogLoss

[6] http://scikit-learn.org/stable/modules/feature_extraction.html

[7] http://scikit-learn.org/stable/modules/decomposition.html#lsa

[8] http://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.LabelEncoder.html

[9] https://xgboost.readthedocs.io/en/latest/get_started/