Machine Learning Engineer Nanodegree

Capstone Project

Bharat Mehan
August 13th, 2020

Project Overview

A lot has been said during the past several years about how precision medicine and, more concretely, how genetic testing is going to disrupt the way diseases like cancer are treated.

But this is only partially happening due to the huge amount of manual work still required. Memorial Sloan Kettering Cancer Centre(MSKCC) launched this competition, accepted by the NIPS 2017 Competition Track, because we they help to take personalized medicine to its full potential.

Problem Statement

Once sequenced, a cancer tumour can have thousands of genetic mutations. But the challenge is distinguishing the mutations that contribute to tumour 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.

Hence, we need to develop a Machine Learning algorithm that, using this knowledge base as a baseline, automatically classifies genetic variations.

Metrics

We will optimize and evaluate our solution against **Multi Class Log Loss** between the predicted probability and the observed target. For each ID in our test dataset, we must predict a probability for each of the different classes a genetic mutation can be classified on.

https://scikit-learn.org/stable/modules/generated/sklearn.metrics.log_loss.html

Data Exploration

The datasets are provided by MSKCC on Kaggle competition website. They are free to download.

In order to get the dataset please create a login account to Kaggle and go to this problem statement page(given above) and download 2 datasets ***training_variants.zip*** and ***training_text.zip.***

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

In this competition we will develop algorithms to classify genetic mutations based on clinical evidence (text). And, there are nine different classes a genetic mutation can be classified on.

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

```
#Verifying the load and format by viewing top 5 rows.
data_variants.head()
```

	ID	Gene	Variation	Class
0	0	FAM58A	Truncating Mutations	1
1	1	CBL	W802*	2
2	2	CBL	Q249E	2
3	3	CBL	N454D	3
4	4	CBL	L399V	4

Let's understand the above 4 columns from variants dataset:

- **ID** - Row Id which is used to link this mutation to the clinical evidence.
- **Gene** - The gene where this genetic mutation is located.
- **Variation** - The amino acid change for these mutations.
- **Class** - The class (1-9) on which this genetic mutation has been classified on.

```
#Applying tghe same few basic exploration commands
data_text.head()
```

	ID	TEXT
0	0	Cyclin-dependent kinases (CDKs) regulate a var...
1	1	Abstract Background Non-small cell lung canc...
2	2	Abstract Background Non-small cell lung canc...
3	3	Recent evidence has demonstrated that acquired...
4	4	Oncogenic mutations in the monomeric Casitas B...

```
data_text.info()
```

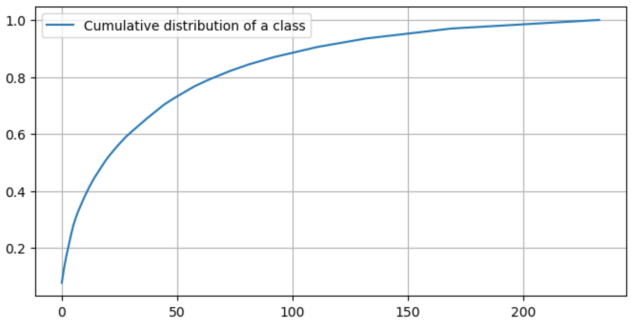
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3321 entries, 0 to 3320
Data columns (total 2 columns):
 #   Column  Non-Null Count  Dtype 
---  ------  --------------  ----- 
 0   ID      3321 non-null   int64 
 1   TEXT    3316 non-null   object
dtypes: int64(1), object(1)
memory usage: 52.0+ KB
```

Both data sets are provided via two different files. One (training variants) provides the information about the genetic mutations, whereas the other (training 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 amino acid 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).

Exploratory Visualization



Algorithms and Techniques

I plan to compare 3 to 4 different models. Because this is a classification problem, below are the main algorithms that I will be Implementing and comparing:

1. **Naïve Bayes:** We will use Multinomial NB classifier and CalibratedClassifierCV with method = 'sigmoid'. Also, we will use a list of different learning rates to calculate the best hyper-parameter and then use it to make predictions.

2. **Logistic Regression:** We will use SGD classifier with parameter loss ='log' which gives us logistic regression. We will use standard regularization 'l2'. We implement this both with class balancing and without class balancing. Also, we will use a list of different learning rates to calculate the best hyper-parameter and then use it to make predictions.

3. **SVM:** We use SGD classifier with loss = 'hinge' which makes it a linear support vector machine. Again, We implement this both with class balancing and without class balancing. We will use a list of different learning rates to calculate the best hyper-parameter and then use it to make predictions.

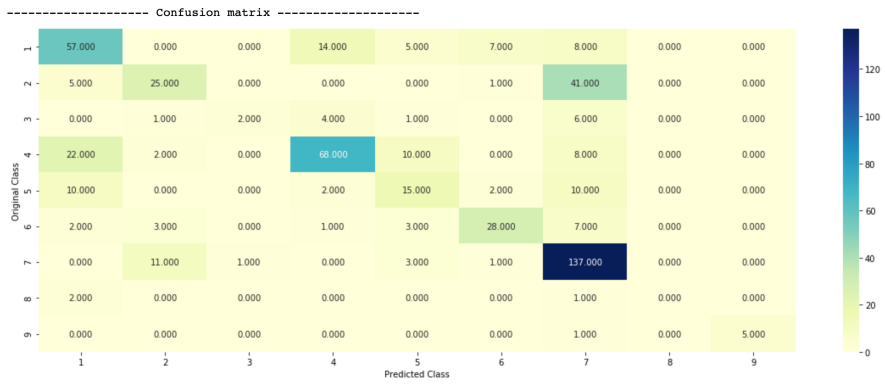
4. **Random Forest Classifier:** Here we use list of estimators and list of depts to try and find the best hyperparameters. We use 'gini' criterion along with CalibratedClassifierCV as 'sigmoid'.

5. Finally, we create a Stacking model where we stack 3 important classifiers and see the result.

Benchmark

Text Classification – We took Naive Bayes as our benchmark model and the evaluation results for our Naïve Bayes implementation are as below:

Log Loss : 1.2282063385536397
Number of mis-classified point : 0.36654135338345867



Data Pre-processing

1. Performed one-hot encoding on the features as columns contains nominal data with fixed number of possible values that are not ordinal.
2. Also, we remove stopwords, multiple spaces and mixed cases from the Text column.

Model Evaluation and Validation

- In each algorithm we use cross validation data to calculate our best hyper parameters for e.g. alpha/learning rate. And then we apply our best solution to test the performance of the given algorithm with that best alpha using test datasets.
- Hence, we are splitting up the datasets into train, validation and test datasets.