Machine Learning Engineer Nanodegree

Capstone Proposal:

Nupur Kamble

4th July 2017

Overview:

Predicting a biological response :Kaggle Dataset.

It aims at predicting a biological response of molecules from their chemical properties. The dataset is hosted on kaggle and made available by *Boehringer Ingelheim.*

Dataset Format: It is provided in form of Train and Test data formats which is csv files.

A general look at Training and Testing data

Training data: Data shape: (3751,1777).

Testing data : Data shape: (2501,1776).

Application of ML in Pharma (link):

https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4107394/

(the use of machine-vision methods to improve information extraction from high-content assays, and the use of active machine learning to drive experimentation.)

Research Properties :

1) The development of new drug largely depends upon trial and error method.
2) It typically involves synthesizing thousands of compounds that finally becomes a drug
3) As a result, this process is extremely expensive and slow
4) Therefore, the ability to accurately predict the biological activity of molecules and understand the rationale behind those predictions are of great value.

   1) Competition Details and Domain Background

   In its search for a model to accurately predict the biological responses of molecules, pharmaceutical company Boehringer Ingelheim turned to an online science community to ask it to come up with an algorithm that would be able to predict a biological endpoint to a molecule by only knowing its structure and composition.

Numerous strategies can be applied to prediction challenges, but the point of this crowdsourcing competition was to find the best one. And sizeable carrots were dangled in front of participants in the form of a prize purse of $20,000.

The competition was well supported with more than 800 scientists submitting more than 9,000 entries during a three-month period. To encourage participants to continually refine their models and to get ahead of their rivals a leaderboard was installed from the outset.

Some of the participants had no formal training in chemistry, yet according to David Thompson, Boehringer's social media strategist some of their results were as good as, if not better than some of those that the academic community produced.
Academic Paper on Machine Learning Approaches in Drug discovery methods and applications
http://csmres.co.uk/cs.public.upd/article-downloads/Machine-learning-approaches-in-drug-discovery-methods-and-applications.pdf
A great research paper on use of ensembles and other classification techniques in drug discovery.

2)Problem Statement:

The objective of this competition is to help build a model so that one can optimally relate the molecular information to biological response. The model can be reproduced using different techniques of learning algorithms if not so with greater accuracies. Metrics such as accuracy_score or rms value are generally chosen. However, we are using log_loss as a metric in order to rank the submissions. The problem contains only quantitative variables and also been scaled in training and testing set. Thus, no categorical variables or date time variables are involved.

This is a classification problem.

Datasets and Inputs:

Features are chemical properties(D1 to D1776)

Target or Label is : Activity(0 and 1)(well balanced classes in target)

Inputs are: 1776 features from test set

Outputs: A submission file classifying each instance of test set as 0 and 1(Molecule ID)in terms of probabilities.Hence, log_loss score is calculated.

Training instance shape:(3751r,1777c)

All training features are quantitative in nature and straight.

Test set shape:(2501r,1776c)

No label for testing data. The testing labels and logloss is calculated on the competition link.We just submit predictions and logloss score is obtained for predictions from kaggle.

So, in order to select best models for stacking procedure, we apply GridSearchCV(validation and testing would be done) on

1)RandomForestClassifier
2)ExtraTreesClassifier
3)Gradient Boosting Model

To find best scoring model for each classifier by tuning hyperparameters.

Link: https://www.kaggle.com/c/bioresponse/data

3) Benchmark Model

Metric used for ranking is log_loss.

This is the multi-class version of the Logarithmic Loss metric. Each observation is in one class and for each observation, you submit a predicted probability for each class. The metric is negative the log likelihood of the model that says each test observation is chosen independently from a distribution that places the submitted probability mass on the corresponding class, for each observation.

logloss=−1N∑i=1N∑j=1Myi,jlog(pi,j)logloss=−1N∑i=1N∑j=1Myi,jlog⁡(pi,j)

where N is the number of observations, M is the number of class labels, loglog is the natural logarithm, yi,jyi,j is 1 if observation ii is in class jj and 0 otherwise, and pi,jpi,j is the predicted probability that observation ii is in class jj.

The model that has been selected has a logloss of 0.37356 ranking first in 729 competitiors.This is my benchmarking model.
Source: Kaggle leaderboard: https://www.kaggle.com/c/bioresponse/leaderboard

My submission produces a logloss score of 0.37751
This logloss score ranks around top 16 members out of 699 teams with 792 competitors.

4)Evaluation Metrics

Using log_loss for evaluation of model
We import log_loss from sklearn as a metric our competitions choice.
sklearn.metrics.**log_loss**(*y_true*, *y_pred*, *eps=1e-15*, *normalize=True*, *sample_weight=None*)

Log loss, aka logistic loss or cross-entropy.

This is the loss function used in (multinomial) logistic regression and extensions of it such as neural networks, defined as the negative log-likelihood of true labels given a probabilistic classifier’s predictions. For a single sample with true label yt in {0,1} and estimated probability yp that yt = 1, the log loss is
-log P(yt|yp) = -(yt log(yp) + (1 - yt) log(1 - yp))

5)Project Design

Solution Statement

1)Data Preprocessing:

a)The data doesn’t have missing values and Nan. All the data is quantitative in nature and none of them is categorical except the classification label Activity.

b)Observing the skewness of data using data.skew(), it can be observed that most of quantitative variables are between -1 to 1 and some are just close. Apply np.log to features won’t generate much leverage.

c)The data is scaled. No large gap between the values of different features are seen.

Hence, the data provided is clean.

PCA can be used since 1776 features are present. However, it is better to use all the features for training model.

Here is the flow of stacking algorithm

```
            X = Features set , y=response and test_set= test
blend_train: np.zeros((X.shape[0],len(classifiers))
blend_test:np.zeros(test.shape[0],len(classifiers))
1)Generate crossvalidation indices using skf=StratifiedShuffleSplit for k=10
2)for i, classifier in enum(classifierlist)
   blend_test_j = np.zeros(test.shape[0],len(skf))
     For j,(trainind, testind) in skf:
          Xtrain = X[trainind]
          Ytrain = y[trainind]
          Xtest = X[testind]
          Ytest = y[testind]
          Clf.fit(xtrain,ytrain)
          blend_train[:,i]=clf.predict_proba(xtest)[:,1]
          blend_test[:,j] = clf.predict_proba(test)[:,1]
      blend_test[:,i] = blend_test_j.mean(1)#mean along x axis
```

We have generated meta features. Now, use another classifier to train on (blend_train, Y) and predict on blend_test

```
clf = LogisticRegression().fit(blend_train,Y).predict_proba(blend_test)[:,1]
```

Thus, we classified the molecules using stacking algorithm

We save the predictions

Weighted Ensemble Model

```
Algorithm: 1) Save the predictions of each classifier as
               predictions =[p1,p2,p3,p4]
           2)  Now, import minimize function from scipy.optimize
           3)Set the constraints: a)starting values(weights) = 0.5 weight for every prediction
                                  b)weights are bound between 0 and 1
                                  c)con=({'type':'eq','fun':lambda w: 1-sum(w)})
                                  d) Select the SLSQP method for minimizing the log_loss_func which takes weights as numpy array.
           4)run: res = minimize(log_loss_func, starting_values, method='SLSQP', bounds=bounds, constraints=cons).
           5)extracts weighs from res as weights = res['w']
           6)now,preds= weights[0]*predictions[0]+weights[1]*predictions[1]+ weights[2]*predictions[2]+weights[3]*predictions[3]
           7)Submit these preds to log_loss function in order to calculate log_loss
```

Hence, the weighted ensemble algorithm.

Classifiers used for stacking in both algorithms are RandomForests, GradientBoost and ExtraTreesClassifier.

RandomForests: Versatile, Easy to train on large and almost require no parameter tuning.

GradientBoost: works on principle of learning rate though it is a bit slow, it achieves more accuracy on biclassification labels than random forests.

ExtraTreesClassifier: Splits are selected randomly unlike with criterion in RandomForests. Performances are quite comparable to RandomForest and sometimes a bit better.

Also, since this is from kaggle competition, where accuracy has more importance, GradientBoost, Randomforests are first choices of competitors.

Hence, the entire description of stacking and ensembles.
