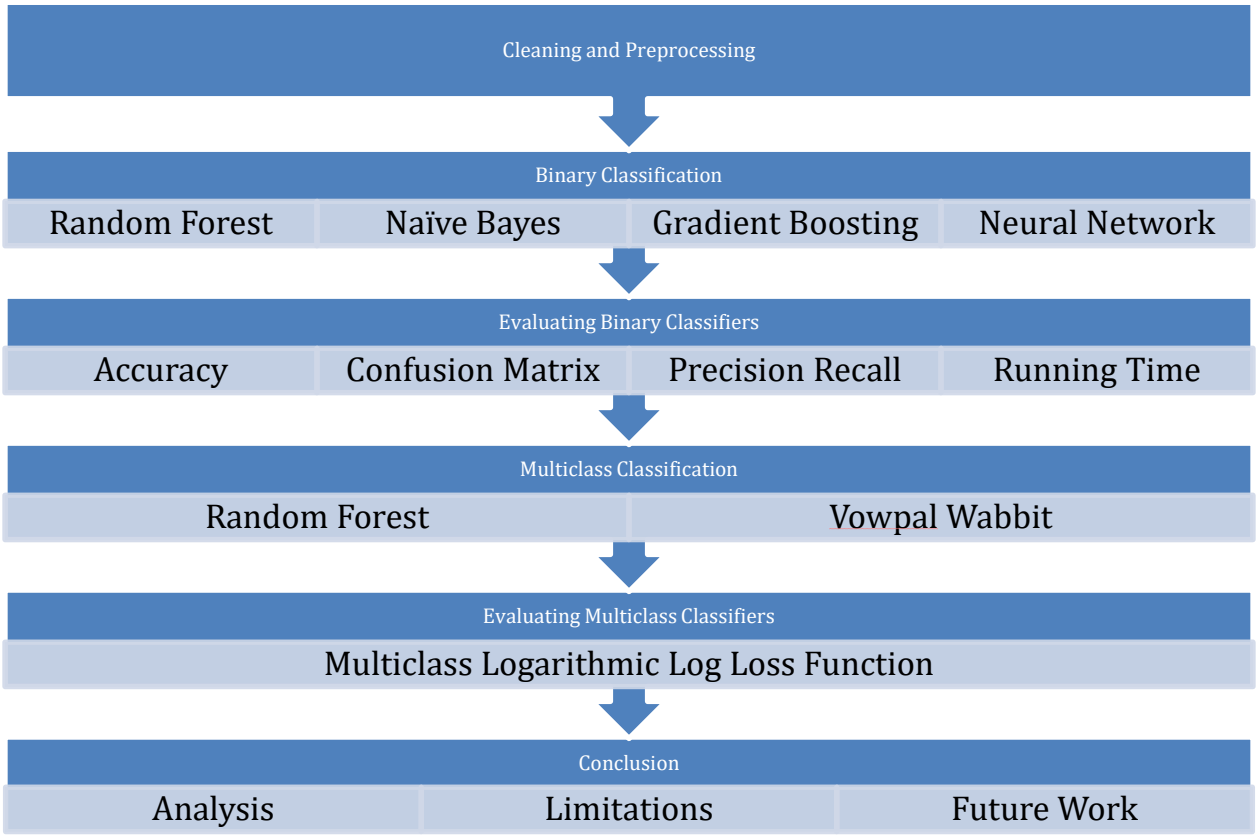
CAP5771/CAP4770:
INTRODUCTION TO DATA SCIENCE

Project: Predict Closed Questions on StackOverflow



Project By:

Group ID: **21**

Mohit Israni (43384979)

Chaitanya Prava Leela (95304385)

PROBLEM STATEMENT

The goal of the project is to build a classifier that predicts whether a question posted by a user on Stack Overflow, a popular question answering service devoted to software programming,

1. will be closed given the question as submitted.
2. Along with the reason that the question was closed.

Also available is additional data regarding the user at question creation time. Questions on StackOverflow can be closed as off topic (OT), not constructive (NC), not a real question (NRQ), too localized (TL) or exact duplicate. Exact duplicate reason is excluded from the project because it depends on posts history, which is very resource intensive.

SKIP TO EXPERIMENTAL RESULTS

RELATED WORK

In the recent years Community Question Answering services has become the subject of numerous studies. Content quality is the central issue of services based on User-Generated Content. The paper by Agichtein E. et al. titled 'Finding high-quality content in social media' addressed the problem of automatic identification of high quality questions and answers in a dataset obtained from Yahoo!Answers. Using a wide range of features – content features, usage statistics, user relationships – they were able to separate high-quality items from the rest with a high accuracy. Other papers have introduced the question dichotomy conversational vs. informational, where the former questions are asked purely to start discussion and the latter are aimed at satisfying an actual information need.That is solved using binary classifier for these question types based on category, question text and asker's social network characteristics

A considerable number of studies have been done recently based on Stack Overflow data.

Task Description

The task along with data and evaluation metrics was offered as an open machine learning competition on Kaggle platform. To solve this problem, a wide range of classification features related to users, their interactions, and post content are employed. Classification is carried out using several machine learning methods. According to the results of the experiment, the most key features are characteristics of the user and topical features of the question. The main task was to build a classifier that assigns a question to one of the five classes: open question and four classes of closed questions. In fact, the requirement was to calculate the probabilities of five class for each question, so as to calculate the multiclass logarithmic loss function which would be the main evaluation metric for the problem.

The question can be closed by a Stack Overflow moderator for one of the reasons:
Off topic (OT), not constructive (NC), not a real question (NRQ), and too localized (TL).
Besides, there is an additional cause exact duplicate (ED), but it is beyond the scope of the study. Out of 6,000 questions posted on the service daily, about 6% end up closed by moderators.

Off topic (OT): questions fall out of the core Stack Overflow focus – software programming. Despite the following question closed as OT is related to programming documentation, it is not about programming per se.
Too localized (TL): question is unlikely to be helpful for anyone in the future; is relevant only to a small geographic area, a specific time point, or an extraordinary narrow situation that is not generally relevant to the global audience.
Not constructive (NC): question does not fit well to Q&A format. While "good" questions imply facts, references, or specific expertise in answers, this sort of questions will likely solicit opinion, debate, arguments, polling, or extended discussion.
Not a real question (NRQ): is an ill-formulated request. This type of questions is ambiguous, vague, incomplete, overly broad or rhetorical and cannot be reasonably answered in its current form.

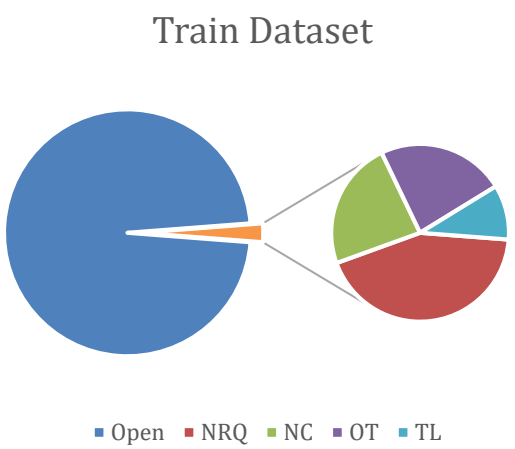
The challenge organized by Stack Overflow on Kaggle platform offered the task of automatic prediction of closed questions along with the reason of closing. The competition was held in August–November 2012 and attracted 167 teams as participants.

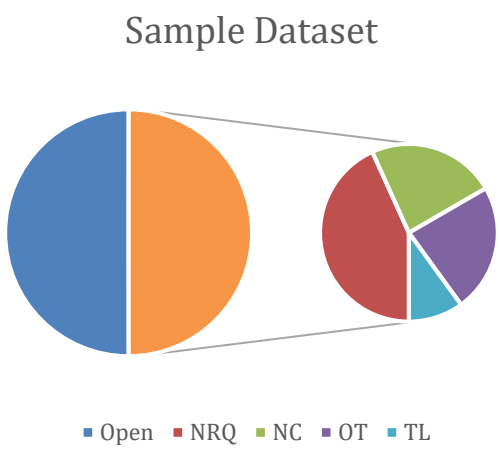
In this project we first using binary classifiers to predict whether the question will be closed or open. Followed by obtaining probabilities of the question being open or closed due to one of the above-mentioned reasons.

DATA SET

The dataset for the project was obtained from Kaggle and it includes train data which contains 3664927 posts and train-sample data consisting of 178 351 posts. Full train data and sample train data distribution on closed reasons is shown in table below

Dataset	NRQ	NC	OT	Open	TL
Train	38622	20897	20865	3575678	8910
Sample	38622	20897	20865	89337	8910





The training data contains data through July 31st UTC, and the public leaderboard data goes from August 1 UTC to August 14 UTC.

The train.csv file contains post text and associated metadata which will serve as inputs to the classification technique. It contains the following fields.

- Input
  - PostCreationDate
  - OwnerUserId
  - OwnerCreationDate
  - ReputationAtPostCreation
  - OwnerUndeletedAnswerCountAtPostTime
  - Title
  - BodyMarkdown
  - Tag1
  - Tag2
  - Tag3
  - Tag4
  - Tag5
- Output
  - OpenStatus
- Additional Data
  - PostId
  - PostClosedDate

The file train-sample.csv is a stratified sample of the training data: it contains every closed question and an equally-sized random sample of the open questions in the training data. All questions will have a value in Tag1, but Tags 2 through 5 are optional.

PRE-PROCESSING AND DATA CLEANING

To get a richer description of posts to be classified several given features in the data were modified. The classification features that can be used for StackOverflow can be grouped as follows:

1. User features

2. User Interaction features
3. Post features
4. Word unigrams, bigrams, and trigrams.

However, the we only made the use of first 3 features groups to build our classifier.

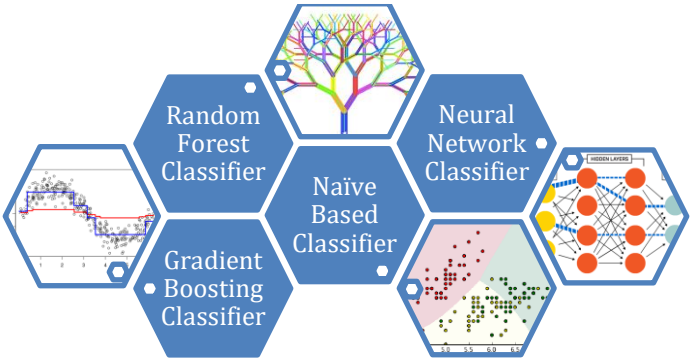
- Currently, the **PostCreationDate** feature is in a month-day-year format. This isn't very useful on its own, so we split this variable into three separate features. It was observed while feature selection that there is a slight penalty for asking questions on weekends.
- We remove the PostCreationDate and OwnerCreationDate from the data and insert a new feature called 'age' which is the difference of two deleted features.
- There were options for tags in the post, tag1, tag2, tag3, tag4, tag5, of which a post always had tag1 and tag2, but rest were optional. So, for our classification, we just counted the number of tags in the post and removed the tags used in the post from our classification features list.
- Given the body of text, length of body is calculated and added as a new feature while the body is dropped as a feature.
- A binary feature is created, stating whether there is code in the body.
- A feature for number of code blocks is created, which is found by parsing the body text in the given features.
- 3 features, for number of lines in the body, number of words in the body and ratio of code to body are added.
- There are many chances, that 'homework' written in the title, body and tags, represent that the user is new to that language and there are many chances of his question getting closed. Hence three separate features for homework in title, body and tags were created, due to different importance to its presence in all three.
- The feature, owner reputation was used as is.

Creation of Small balanced data

As the dataset contained **3575678** records for open status while only **89337** records for closed, it is highly skewed with ratio 94:6. Hence, to create a binary classifier a small dataset was created by undersampling the open status records.

CLASSIFICATION TECHNIQUES USED

Problem 1: BINARY CLASSIFICATION



1. **Random Forest**:

   Random forests or random decision forests are an ensemble learning method for classification, regression and other tasks, that operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees.

   It was trained with the following features:
   n-estimators = 100 -the number of trees in the forest. More estimators mean in general better results but training time increases randomly with estimators.
   maximum tree depth = 15 was specified explicitly, so when building the tree the nodes were expanded until maximum of 15 splits take place till leaves.

2. **Naïve Bayes Classifier**

   It is a classification technique based on Bayes' Theorem with an assumption of independence among predictors. In simple terms, a Naive Bayes classifier assumes that the presence of a feature in a class is unrelated to the presence of any other feature. Even if these features depend on each other or upon the existence of the other features, these properties independently contribute to the probability that a record results in a particular outcome and that is why it is known as 'Naive'.

3. **Gradient Boosting Classifier**

   In gradient boosting, it trains many model sequentially. Each new model gradually minimizes the loss function (y = ax + b + e, e needs special attention as it is an error term) of the whole system using Gradient Descent method. The learning procedure consecutively fit new models to provide a more accurate estimate of the response variable.
   The principle idea behind this algorithm is to construct new base learners which can be maximally correlated with negative gradient of the loss function, associated with the whole ensemble.

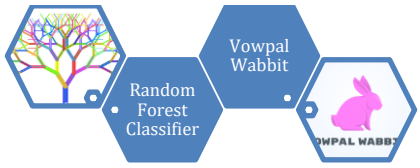
4. **Neural Network Classifier**

   Neural Networks are statistical learning models, which are inspired by biological neural networks. These networks are represented as system of interconnected neurons. These connections within the network are suitable for supervised learning as they can be systematically adjusted based on inputs & outputs. A neural network is a collection of "neurons" with "synapses" connecting them. This collection is organized into mainly three parts i.e. the input layer, the hidden layer and the output layer.

Improvements after the last presentation

Implemented Multiclass Classification, and changed to more accurate metric to analyze the classifier.

Problem 2: MULTICLASS CLASSIFICATION



1. **Random Forest Classifier**
2. **Vowpal Wabbit**

Vowpal Wabbit is a library developed by John Langford. VW focuses on the approach to feed the examples to an online-learning algorithm in contrast to parallelization of a batch learning algorithm over many machines. The default learning algorithm is a variant of online gradient descent.

There are several features of Vowpal Wabbit that (in combination) can be powerful.

Input Format. The input format for the learning algorithm is substantially more flexible than might be expected. Examples can have features consisting of free form text, which is interpreted in a bag-of-words way. There can even be multiple sets of free form text in different namespaces.

Speed. The learning algorithm is pretty fast---similar to the few other online algorithm implementations out there. As one datapoint, it can be effectively applied on learning problems with a sparse terafeature (i.e. 1012 sparse features).

Scalability. This is not the same as fast. Instead, the important characteristic here is that the memory footprint of the program is bounded independent of data. This means the training set is not loaded into main memory before learning starts. In addition, the size of the set of features is bounded independent of the amount of training data using the hashing trick.

Feature Pairing. Subsets of features can be internally paired so that the algorithm is linear in the cross-product of the subsets. This is useful for ranking problems. David Grangier seems to have a similar trick in the PAMIR code. The alternative of explicitly expanding the features before feeding them into the learning algorithm can be both computation and space intensive, depending on how it's handled.

EXPERIMENTAL RESULTS

BINARY CLASSIFICATION:

The metrics for evaluation of classification models used were accuracies and confusion matrix.

	Random Forest Classifier	**Naïve Based Classifier**	**Gradient Boosting Classifier**	**Neural Network Classifier**
Time to Build Classifier	1.99	0.04	12.17	5.81
Accuracy on Train Data	0.79	0.540	0.70	0.68

Accuracy on Test Data	0.69	0.543	0.692	0.67
Precision	0.69	0.87	0.69	0.62
Recall	0.72	0.52	0.71	0.17
F - score	0.70	0.65	0.699	0.27

The binary classification of just predicting whether the question will be closed or not gave the highest accuracy when trained with the Random Forest Classifier. These results were obtained for small dataset which is balanced in terms of open and closed questions.

The precision and recall also was considerably better for random forest and gradient boosting classifier. However, the gradient boosting classifier was computation and time intensive.

Random Forest Classifier			
		Actual Status	
		Closed	Open
Predicted Status	Closed	12008	5870
	Open	5073	12720

Naïve Bayes Classifier			
		Actual Status	
		Closed	Open
Predicted Status	Closed	4368	13510
	Open	2787	15006

Gradient Boosting Algorithm			
		Actual Status	
		Closed	Open
Predicted Status	Closed	12064	5814
	Open	5141	12652

Neural Network Classifier			
		Actual Status	
		Closed	Open
Predicted Status	Closed	16001	1824
	Open	14815	3031

Since the complete data (data_large.csv) was highly biased, the accuracy predicted of 0.97 is not a correct measure for evaluation of the classification model. Hence, instead a Multiclass logarithmic loss function was used, which was also the evaluation metric specified for the kaggle competition that hosted the question.

In a multi-classification problem, we define the logarithmic loss function F in terms of the logarithmic loss function per label Fi as:

$$F=-1N\sum_{iN}\sum_{jM}y_{ij}\cdot Ln(p_{ij}))=\sum_{jM}(-1N\sum_{iN}y_{ij}\cdot Ln(p_{ij})))=\sum_{jM}F_i$$

where *N* is the number of instances, *M* is the number of different labels, *yij* is the binary variable with the expected labels and *pij* is the classificiation probability output by the classifier for the *i*-instance and the *j* -label. The cost function *F* measures the distance between two probability distributions, i.e. how similar is the distribution of actual labels and classifier probabilities. Hence, values close to zero are preferred.

MULTICLASS CLASSIFICATION:

The multiclass classification problem considered also the reason due to which the question on Stack Overflow gets closed. The classifiers used were Random forest multiclass classifier and Vowpal Wabbit classifier. The metric used were accuracy (not a good measure for skewed data) and Multiclass Logarithmic Loss function.

		Small Dataset	Large Dataset
Random Forest Classifier	**Runtime**	2.6203s	78.19s
	Accuracy	0.55	0.97
	Logarithmic Loss	1.36	**0.427**
Vowpal Wabbit	**Runtime**	0.551s	9.818s
	Accuracy	N/A	N/A
	Logarithmic Loss	1.201	**0.13**

Random Forest Classifier						
		Actual Status				
		Not a real question	**Not constructive**	**Off Topic**	**Open**	**Too Localized**
Predicted Status	**Not a real question**	49	6	5	7687	3
	Not constructive	9	1	0	4139	0
	Off Topic	11	3	3	4129	1
	Open	241	39	77	714812	14
	Too Localized	2	0	0	1740	1

As can be seen from the tables above, the accuracy, how much ever high (0.97) for random classifier does not imply that the classifier is good, as can be seen from the confusion matrix, most of the data is confused as open. Hence, we use multiclass logarithmic loss(MLL) function which is calculated based on probabilities of every record belonging to a class. The smaller the value of logarithmic loss function, better the classifier. Hence, Vowpal Wabbit is a better classifier in terms of predication as compared to Random Classifier as its MLL value was just 0.13 for the large dataset when compared to 0.427 for the Random Classifier.

Scalability

1. Preprocessing:
   Map function was used to improve the reading and pre-processing of the train and test data, parallelizing the process. Due to which the processing time of large data was brought down from 4 hours to under 5 mins on the AWS EC2 32 core instance.
2. Multiclass Classification:
   Vowpal wabbit can be run on very large data size due to its following features:
   - Out-of-core online learning: no need to load all data into memory
   - The hashing trick: feature identities are converted to a weight index via a hash
   - Exploiting multi-core CPUs: parsing of input and learning are done in separate threads.

CONCLUSION

The best results were obtained using Vowpal Wabbit – an implementation of online learning based on stochastic gradient descent.

LIMITATIONS

After some visual analysis it was noted that several questions that as per normal standards should be closed but were however open. Since not every question asked on Stack Overflow is scrutinized for validity, there are many chances that a question remains open. So, classification on such peer based review and monitored is difficult for Community Question Answering Services.

Again, since the data is peer monitored, two questions with two completely same feature set can have different true outcomes, which can raise error in the classifier.

FUTURE WORK

Several features in the classifier can be added, which might include certain words in the body, title, which would require further text mining.

Also, sometimes it is very hard to determine too localized question because it can be seen from the text of the post, although sometimes it is enough to look at the code included in the post body. We did not analyze the content of the code in any way during the classification. It is a nontrivial task to determine if the code works for specific conditions and will never be useful for anyone else in the future. It might be helpful to analyze stack traces that often appear in code blocks and are a good signal of too specific questions.

In our classification problem it has been assumed that a question if closed is due one of the stated reasons, however there is a possibility that the closing of question is due to more than one reasons, which can be considered to improve the prediction of closing of questions.

REFERENCES

Dataset:

https://www.kaggle.com/c/predict-closed-questions-on-stack-overflow/data

Papers:

- https://arxiv.org/pdf/1307.7291.pdf
- http://ceur-ws.org/Vol-1031/paper2.pdf
- http://kansas.ru/pb/paper/kazan2013.pdf
- Agichtein E., Castillo C., Donato D., Gionis A., Mishne G. Finding high-quality content in social media // Proc. 2008 Int. Conf. on Web Search and Data Mining. – N. Y.: ACM, 2008. – P. 183–194.
- Correa D., Sureka A. Fit or unfit: Analysis and prediction of “closed questions” on stack overflow. – 2013. – arXiv:1307.7291.
- Barua A., Thomas S.W., Hassan A.E. What are developers talking about? An analysis of topics and trends in Stack Overflow // Empir. Software Eng. – 2014. – V. 19, No 3. – P. 619–654.