Predict Closed Questions on StackOverflow

Sourav Chanduka
201464104
sourav.chanduka@research.iiit.ac.in

Rounak Mundra
201464098
rounak.mundra@research.iiit.ac.in

***Abstract*—Millions of programmers use StackOverflow to get high quality answers to their programming questions every day. There has evolved an effective culture of moderation to safeguard it. More than six thousand new questions is asked on StackOverflow every weekday. Currently about 6% of all new questions end up closed. The goal of this paper is to build a classifier that predicts whether or not a question will be closed given the question as submitted, along with the reason that the question was closed.**

I. Introduction

In recent time question-answer services like StackOverflow are becoming more popular. Knowledge of such services has been steadily growing so it requires more resources to moderate. Some automation of this process would ease this task. The problem solved in this paper is a small step in this direction.StackOverflow is a service where users ask questions about programming and it belongs to StackExchange network which contains many thematic websites. Questions on StackOverflow can be closed as off topic (OT), not constructive (NC), not a real question (NRQ), too localized (TL) or exact duplicate. Off topic is a question that is not on-topic of the site or is related to another site in Stack Exchange network. Example: Is there a way to turn off the automatic text translation at the MSDN library pages ? I do prefer English text but due to having a German IP address Microsoft activates the automatic translation on every new page load which gives me a yellow box with a German translation of the text I am currently hovering over with the mouse. This happens regardless what language is initially set in the right upper corner and regardless of whether I am logged in or not. I cant tell how annoying this is !! Any ideas, anyone ? Too localized is a question that is unlikely to be helpful for anyone in the future; it is only relevant to a small geographic area, a specific moment in a time, or an extraordinary narrow situation that is not generally applicable to the worldwide audience of the internet. Example: Is it time to start using HTML5? Someone has to start sometime but is now the time? Is it possible to use the new HTML5 tags and code in such a way as to degrade gracefully? Not constructive is a question that is not a good fit to QA format. It is expected that the answers generally involve facts, references, or specific expertise; this question will likely solicit opinion, debate, arguments, polling, or extended discussion. Example: What is the best comment in source code you have ever encountered? Not a real question is a question when its difficult to tell what is being asked here. This question is ambiguous, vague, incomplete, overly broad or rhetorical and cannot be reasonably answered in its current form. Example: For a few days Ive tried to wrap my head around the functional programming paradigm in Haskell. Ive done this by reading tutorials and watching screencasts, but nothing really seems to stick. Now, in learning various imperative/OO languages (like C, Java, PHP), exercises have been a good way for me to go. But since I dont really know what Haskell is capable of and because there are many new concepts to utilize, I havent known where to start. So, how did you learn Haskell? What made you really break the ice? Also, any good ideas for beginning exercises?

II. Datasets

For this task the data was provided by kaggle and it includes train data which contains 3664927 posts (instances) and train sample data consisting of 178351 posts. Full train data and sample train data distribution on closed reasons is shown in table.

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

There is about 4 million instances totalling 3.7 GB of data provided by Kaggle.

Data Available in Dataset :

- Post_Id - Id number of post.
- PostCreationDate - Date on which question was posted.
- OwnerCreationDate - Date on which Owner created his profile/account.
- ReputationAtPostCreation - Reputation of Owner at the time of post creation.
- OwnerUndeletedAnswerCountAtPostTime - Number of answers of owner remained un-deleted at the time of post creation.
- Title -Title or Question of post.
- BodyMarkdown - Elaborative explanation of what exactly owner wants to ask.
- Atmost 5 Tags - Topics related to post.
- PostClosedDate - Date of post getting closed (if any).
- Status - Binary value. 0 if open and 1 if closed.

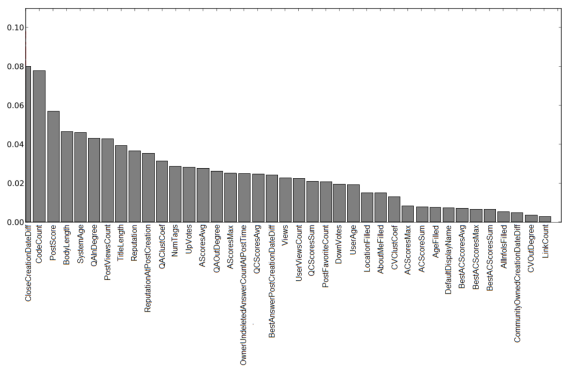

Fig. 1. Relative feature importance.

III. Preprocessing Details

Pre-Processing is essential for efficient Feature Extraction leading to non-redundant Feature Descriptors. The main steps involved in the Pre-Processing Stage of the Pipeline are :

- Tokenizing: converting a document to its atomic elements.
- Stopping: removing meaningless words.
- Stemming: merging words that are equivalent in meaning.

A. Tokenizing

Tokenization, when applied to data security, is the process of substituting a sensitive data element with a non-sensitive equivalent, referred to as a token, that has no extrinsic or exploitable meaning or value. The token is a reference (i.e. identifier) that maps back to the sensitive data through a tokenization system. The mapping from original data to a token uses methods which render tokens infeasible to reverse in the absence of the tokenization system, for example using tokens created from random numbers.

B. Stopping

Stop words refer to the most common words in a language. Although, there is no universal list for the Stop Words, but an exhaustive list has been used to ensure better accuracy and efficient feature extraction. They need to be removed because they are most likely to be present in both spam and ham emails and hence, both kind of emails will share these tokens when they are converted into their equivalent feature descriptors. Thus, they are removed as a part of pre-processing giving comparably lower dimensional feature descriptors and hence, improved performance in terms of time.

C. Stemming

In linguistic morphology and information retrieval, stemming is the process of reducing inflected (or sometimes derived) words to their word stem, base or root formgenerally a written word form. The stem need not be identical to the morphological root of the word; it is usually sufficient that related words map to the same stem, even if this stem is not in itself a valid root. Many search engines treat words with the same stem as synonyms as a kind of query expansion, a process called conflation.

IV. Feature Extraction

After the pre-processing stage, sufficient amount of redundancy gets removed from the message content. The mails now need to be converted to equivalent feature descriptors that have sufficient number of dimensions to ensure good classification and low enough to ensure that computation performed on them is tractable.

- Status: (0/1) (Open/Closed)
- TitleLength (After text processing)
- BodyLength
- ReputationAtPostCreation: Reputation of User when he posted a question
- NumberOfTags (atleast 1 and upto 5)
- CloseCreationDateDifference: Number of days it took to get deleted. (PostClosedDate - PostCreationDate)
- UserAge: Profile Age (CurrentDate - OwnerCreationDate)
- OwnerUndeletedAnswerCountAtPostTime
- LinksCount

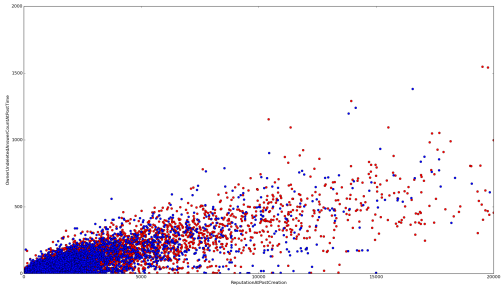

Fig. 2. OwnerUndeletedAnswerCountAtPostTime vs ReputationAtPostCreation

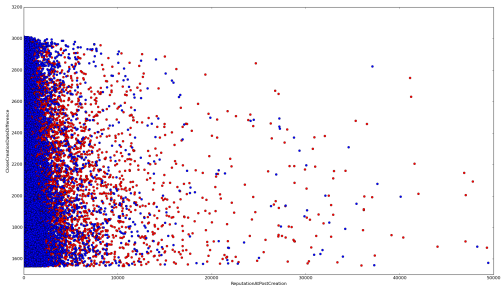

Fig. 3. CloseCreationDateDifference vs ReputationAtPostCreation

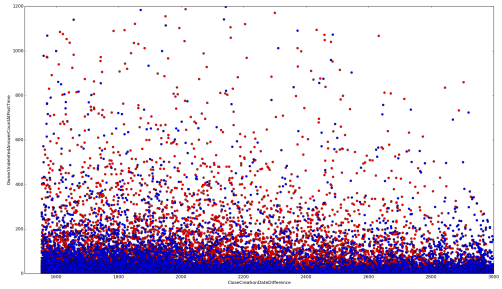

Fig. 4. OwnerUndeletedAnswerCountAtPostTime vs CloseCreationDateDifference

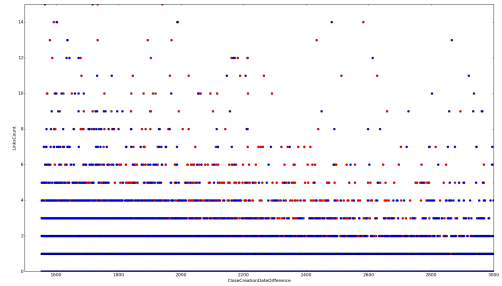

Fig. 5. LinksCount vs CloseCreationDateDifference

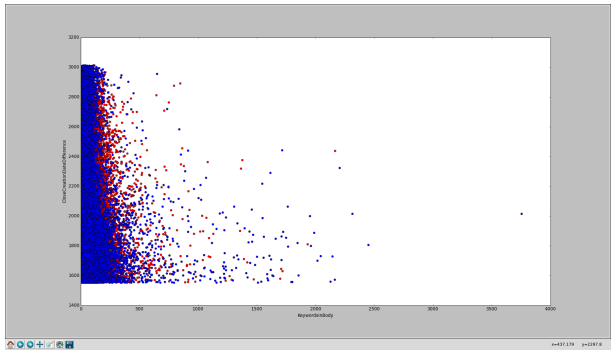

Fig. 6. CloseCreationDateDifference vs KeywordsinBody

V. Used Methods

A. Random Forest

Random forests are a combination of tree predictors such that each tree depends on the values of a random vector sampled independently and with the same distribution for all trees in the forest. The training algorithm for random forests applies the general technique of bootstrap aggregating, or bagging, to tree learners. Given a training set X = x1, ..., xn with responses Y = y1, ..., yn, bagging repeatedly (B times) selects a random sample with replacement of the training set and fits trees to these samples:

For b = 1, ..., B: Sample, with replacement, n training examples from X, Y; call these Xb, Yb. Train a decision or regression tree fb on Xb, Yb.

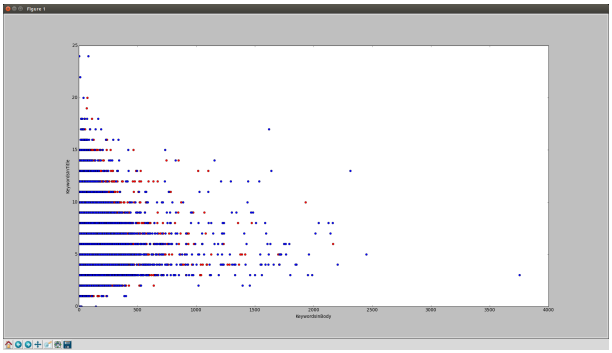

Fig. 7. KeywordsinTitle vs KeywordinBody

After training, predictions for unseen samples x' can be made by averaging the predictions from all the individual regression trees on x':

$$\hat{f} = \frac{1}{B}\sum_{b=1}^{B} \hat{f}_b(x')$$

This bootstrapping procedure leads to better model performance because it decreases the variance of the model, without increasing the bias. This means that while the predictions of a single tree are highly sensitive to noise in its training set, the average of many trees is not, as long as the trees are not correlated. Simply training many trees on a single training set would give strongly correlated trees (or even the same tree many times, if the training algorithm is deterministic); bootstrap sampling is a way of de-correlating the trees by showing them different training sets.

The above procedure describes the original bagging algorithm for trees. Random forests differ in only one way from this general scheme: they use a modified tree learning algorithm that selects, at each candidate split in the learning process, a random subset of the features. This process is sometimes called "feature bagging". The reason for doing this is the correlation of the trees in an ordinary bootstrap sample: if one or a few features are very strong predictors for the response variable (target output), these features will be selected in many of the B trees, causing them to become correlated. An analysis of how bagging and random subspace projection contribute to accuracy gains under different conditions is given by Ho.

B. Support Vector Machines

Support Vector Machines are based on the concept of decision planes that define decision boundaries. A decision plane is one that separates between a set of objects having different class memberships.

In machine learning, the (Gaussian) radial basis function kernel, or RBF kernel, is a popular kernel function used in various kernelized learning algorithms. In particular, it is commonly used in support vector machine classification.

The RBF kernel on two samples x and x', represented as feature vectors in some input space, is defined as:

$$K(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{||\mathbf{x} - \mathbf{x}'||^2}{2\sigma^2}\right)$$

$$K(\mathbf{x}, \mathbf{x}') = \exp(-\gamma||\mathbf{x} - \mathbf{x}'||^2)$$

Since the value of the RBF kernel decreases with distance and ranges between zero (in the limit) and one (when x = x'), it has a ready interpretation as a similarity measure.[2] The feature space of the kernel has an infinite number of dimensions; for $\sigma = 1\sigma = 1, itsexpansionis$:

$$\exp\left(-\frac{1}{2}||\mathbf{x} - \mathbf{x}'||^2\right) = \sum_{j=0}^{\infty} \frac{(\mathbf{x}^\top \mathbf{x}')^j}{j!} \exp\left(-\frac{1}{2}||\mathbf{x}||^2\right) \exp\left(-\frac{1}{2}||\mathbf{x}'||^2\right)$$

C. Vowpal Wabbit

Vowpal Wabbit (VW) is a library and algorithms developed at Yahoo! Research by John Langford.The default learning algorithm is a variant of online gradient descent.As mentioned above the algorithm used in Vowpal Wabbit is a modified stochastic gradient descend algorithm.We have used Logistic loss function.

Logistic	$\log(1 + e^{-yp})$

VI. Evaluation Measures

A. Precision

Precision is the number of True Positives divided by the number of True Positives and False Positives. Put another way, it is the number of positive predictions divided by the total number of positive class values predicted. It is also called the Positive Predictive Value (PPV).

B. Recall

Recall is the number of True Positives divided by the number of True Positives and the number of False Negatives. Put another way it is the number of positive predictions divided by the number of positive class values in the test data. It is also called Sensitivity or the True Positive Rate.

C. F1 Score

The F1 Score is the 2*((precision*recall)/(precision+recall)). It is also called the F Score or the F Measure. Put another way, the F1 score conveys the balance between the precision and the recall.

–decay_learning_rate <d> [= 1]
–initial_t <i> [= 1]
–power_t <p> [= 0.5]
-l [–learning_rate] <l> [= 10]

$$\eta_e = \frac{ld^{n-1}i^p}{(i + \sum_{e'<e} i_{e'})^p}$$

VII. Results

A. Random Forest

	Accuracy	Precision	Recall	F1 Score
Train	65-75%	0.65-0.7	0.55-0.65	0.75-0.85

B. SVM

	Accuracy	Precision	Recall	F1 Score
Train	65%	0.6-0.65	0.5-0.6	0.77-0.82