CMPUT 466/551 Mini Project

Fred Han (xuefei1,G), Brian Lu (blu1,UG)

Problem

For our CMPUT 466/551 project, we plan to investigate the duplicate question problem. As the name suggest, this problem investigates whether two questions in English are in fact, asking the same thing. For example, questions like “Why should I take a machine learning course” and “What are the reasons to take a machine learning course” are duplicate questions. For question answering sites such as StackOverflow or Quora, detecting duplicate questions are essential as it can help clean up their question servers.

This is a binary classification problem, a pair of questions can only be duplicates of each other or non-duplicates of each other.

Dataset

The dataset we selected for our mini-project is Quora's duplicate question pairs dataset:https://data.quora.com/First-Quora-Dataset-Release-Question-Pairs.
The full dataset contains more than 400000 instances. Each instance contains two questions in English and a label of 1 or 0 indicating whether the two questions are duplicates of each other, meaning whether they are asking the same thing. Here are examples of a duplicate question pair and a non-duplicate question pair:

```
How can I be a good geologist?  What should I do to be a great geologist?   1
Should I buy tiago? What keeps childern active and far from phone and video games?  0
```

Since training 400000 instances would take too much time and memory. We randomly sampled 40000 instances from the original dataset as the final dataset for this project. In our smaller dataset half the instances are labelled 1 and the other half are labelled 0.

Models

We selected three classifiers for our problem, Logistic regression, Naive Bayes, and decision tree. The question we are asking is: which one of these three model will achieve the highest F1 score and accuracy on our Quora duplicate questions dataset?

Logistic regression is a generalized linear model with a sigmoid transfer function. It minimize the cross entropy loss function, which is suitable for binary classification problems. We used the logistic regression implementation provided by scikit-learn library.

Naive Bayes is a generative approach that learns a feature distribution p(x|y) and a class label distribution p(y) from the training data. We used the Gaussian Naive Bayes implementation provided by scikit-learn library.

Decision tree is a linear classifier that learns a set of rules to best partition training data. We used the Decision tree classifier implementation provided by scikit-learn library.

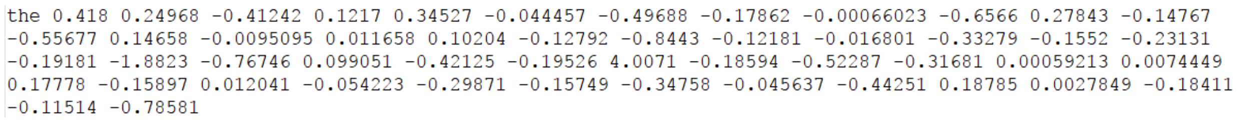
Hypothesis

We make two hypotheses regarding our models:

1. We expect Naive Bayes to have the lowest F1 score and classification accuracy due to its simplicity.
2. We expect logistic regression and decision tree to have equivalent performance. In other words, statistic significance tests should reveal no difference between their f1 scores and accuracies.

Feature extraction

To generate meaningful vector representations from question pairs, we employ pre-trained GloVe English word embeddings: https://nlp.stanford.edu/projects/glove/. Specifically, we use the GloVe 6B 50d embeddings, where each word is represented by a 1x50 vector, and there are 400000 words in the dictionary. Here is an example of the GloVe embedding, the English word 'the' represented as a 1x50 vector of real numbers:



```
the 0.418 0.24968 -0.41242 0.1217 0.34527 -0.044457 -0.49688 -0.17862 -0.00066023 -0.6566 0.27843 -0.14767
-0.55677 0.14658 -0.0095095 0.011658 0.10204 -0.12792 -0.8443 -0.12181 -0.016801 -0.33279 -0.1552 -0.23131
-0.19181 -1.8823 -0.76746 0.099051 -0.42125 -0.19526 4.0071 -0.18594 -0.52287 -0.31681 0.00059213 0.0074449
0.17778 -0.15897 0.012041 -0.054223 -0.29871 -0.15749 -0.34758 -0.045637 -0.44251 0.18785 0.0027849 -0.18411
-0.11514 -0.78581
```

In our dataset, each instance contains two questions, each question consists of one or more sentences, and each sentences consists of multiple words. We also found that the average number of words in a question is 12. Therefore, for each question, we take only its first 12 words, which results in 12x50=600 features. For questions with less than 12 words, we pad the missing features with zeros. Finally, since there are two questions in every instance, we simply concatenate the feature vectors of the two questions together, so each instance in our dataset will have 1200 features.

Evaluation metric

We measured the performance of our models using classification accuracy and F1 score.

Classification accuracy is simply the percent of correct predictions out of all predictions. Since our dataset has a 50-50 class label ratio, accuracy is an intuitive indicator of performance.

The F1 score is a balanced measure of precision and recall, formally defined as 2*(precision*recall)/(precision+recall). Precision is computed as the number of true positives/(number of true positives + number of false positives). Recall is computed as the number of true positives/(number of true positives + number of false negatives). We believe that the F1 score is a strong and comprehensive metric for classification problems because it considers three components in a confusion matrix, true positives, false negatives and false positives.

Experimental design

This section is divided into three subsections. In the first subsection we will introduce how we splitted training and testing data. The second subsection will discuss details related to hyper-parameter tuning. The third subsection will introduce our statistical significance tests.

Training/Testing
We employ a 10-fold cross validation technique to train and evaluate each model. When partitioning our data into 10 folds, we also use stratified sampling to ensure that each fold has the same positive to negative label ratios as the original dataset. So each fold has 4000 instances with 2000 duplicate questions pairs and 2000 non-duplicate pairs.

With 10-fold CV, training and testing is performed 10 times. Every time we leave a different fold out as test set and train on the remaining 9 folds. For each test set we compute the F1 score and accuracy. So we will have 10 samples of F1 score and accuracy for every learning methods after running our 10-fold CV.

Hyper-parameter tuning
We also employ a 3-fold internal cross validation technique for hyper-parameter tuning. We did not do 10-fold internal CV because we found that it would take too much time. Training data is further partitioned into 3 folds, each folds contains 12000 instances. Our parameter tuning algorithm can be summarized by the following pseudo-code:

```
Input: training_data

For h_i in all hyper-parameter combinations:
    Three_folds = partition(training_data)
    For i in range(len(Three_folds)):
        Train model on all folds but the i-th fold
        Test model on the i-th fold, store F1 score
    Compute average F1 score from 3-folds CV
    If average F1 > current highest:
        Update best hyper-parameters
Train model on all training data with best hyper-parameters
```

We use F1 score as the metric to evaluate different hyper-parameters. When partitioning training data into 3 folds, we also used stratified sampling to ensure the same positive to negative label ratio.

Finally, we'll introduce the hyper parameters being tuned on our models.

Logistic regression
C: Inverse regularization strength, smaller value means stronger regularization. Values used for tuning: [0.001, 0.005, 0.01, 0.05, 0.1, 0.5, 1.0, 1.5, 2.0, 10.0]
Penalty: Regularizer. Values used for tuning: ['l1', 'l2']

Naive Bayes
Prior: Distribution of class labels, if none, this value will be learned from training data. Values used for tuning: [None, [0.5, 0.5]]

Decision tree
Criterion: Metric used to split data into two groups. Values used for tuning: [‘gini’, ‘entropy’]
Max_features: Maximum percent/number of features considered. Values used for tuning: [None, 0.005, 0.01, 0.25, 0.5, 0.75, 0.9, 0.95, 'sqrt', 'log2']
Max_depth: Maximum depth of the tree, for regularization. Values used for tuning: [None, 5, 10, 20, 50, 100]

Statistical significance tests
We employ two statistical significance tests, confidence interval overlapping tests and Anova tests. We set the threshold of our p-value = 0.05. Our confidence interval is set at 95%.

To ensure that we get enough samples of F1 score and accuracy measures. We will also run the training/testing operation 10 times. Meaning in each run, we will conduct 10-folds CV to acquire 10 samples of F1 and accuracy. Also in each run we will reload the 10-folds. This ensures that we will get 100 distinct samples of F1 score and accuracy, and it should be enough for the statistical significance tests.

Results
F1 score mean values and confidence intervals for our models:

Algorithm	F1 score mean value	F1 score confidence interval
Logistic regression	0.663981041665	(0.662523914678, 0.665438168652)
Naive Bayes	0.611714631206	(0.610315460885, 0.613113801527)
Decision tree	0.617827424944	(0.615217561714, 0.620437288174)

Accuracy mean values and confidence intervals for our models:

Algorithm	Accuracy mean value	Accuracy confidence interval
Logistic regression	0.655205	(0.653792599737, 0.656617400263)
Naive Bayes	0.60007	(0.598692042059, 0.601447957941)
Decision tree	0.6076225	(0.605788752292, 0.609456247708)

P-values of Anova tests

	F1 score	Accuracy
P-value	7.05132004826e-126	3.74280799516e-154

Conclusion
From the confidence interval tables above, we can observe that for both F1 score and accuracy, there is no overlapping confidence intervals between any two approaches. This test suggests that with enough statistical evidence, the three methods tested are all different from each other. This helped disprove our second hypothesis about logistic regression and decision tree being equal methods.

From the results of our Anova tests, we can that the p-values for F1 score and accuracy are far less than the set threshold 0.05. Small p-values suggest that with strong statistical evidence that the means of all input distributions are different. This again verifies that three learning methods we investigated have different performance results.

So we can confidently conclude that our hypothesis 1 is correct. Performance data with strong statistical evidence suggest that Naive Bayes is the worst performing model on our dataset.
Our hypothesis 2 is incorrect, data suggest that logistic regression is a better model than decision tree, in terms of F1 score and accuracy.