Machine learning engineer nanodegree
Project Proposal

Quora Question Pairs (kaggle)

Domain Background:

Quora is a place to gain and share knowledge—about anything. It's a platform to ask questions and connect with people who contribute unique insights and quality answers. This empowers people to learn from each other and to better understand the world.

Over 100 million people visit Quora every month, so it's no surprise that many people ask similarly worded questions. Multiple questions with the same intent can cause seekers to spend more time finding the best answer to their question, and make writers feel they need to answer multiple versions of the same question. Quora values canonical questions because they provide a better experience to active seekers and writers, and offer more value to both of these groups in the long term.

Problem Statement:

The goal of this project is to predict whether question pairs are duplicate or not i.e predict if the question pairs are of the same meaning. I am going to use NLP techniques for this problem. Doing so will make it easier to find high quality answers to questions resulting in an improved experience for Quora writers, seekers, and readers.

Datasets and Inputs:

The datasets are provided by Quora on Kaggle.

Input Data fields:

- id - the id of a training set question pair

- qid1, qid2 - unique ids of each question (only available in train.csv)

- question1, question2 - the full text of each question

- is_duplicate - the target variable, set to 1 if question1 and question2 have essentially the same meaning, and 0 otherwise.

Solution Statement:

The solution is to predict whether the question pair is duplicate or not. First I will use feature extraction techniques like TF-IDF, wordEmbedding to process all the texts and use some visualization to get better understanding of the data.

For training the model I will use binary classification algorithms like Logistic Regression, SVM etc. Then perform some tuning to achieve better accuracy.

Benchmark Model:

The model given in the Quora problem statement in random forest model. I will use that as a benchmark model, I will use x percent chance prediction, where x is the proportion of “duplicate” among all question pairs in the training set.

Evaluation metrics:

Predicted results are evaluated based on the log loss between the predicted values and ground truth. The ground truth is the set of labels that have been supplied by human experts. The ground truth labels are inherently subjective, as the true meaning of sentences can never be known with certainty. Human labeling is also a 'noisy' process, and reasonable people will disagree. As a result, the ground truth labels on this dataset should be taken to be 'informed' but not 100% accurate, and may include incorrect labeling. We believe the labels, on the whole, to represent a reasonable consensus, but this may often not be true on a case by case basis for individual items in the dataset.

Project Design:

Before starting the model I would understand the shape and format of the data. Then I will apply some NLP techniques like TF-IDF, word count, character count etc.. to extract the features. Also techniques like PCA is not necessary since the feature set in small.

For training the model, I am going to use classification algorithms like regression, decision trees, SVM and random forest. I will use cross validation to find which model performs best and tweak the parameters. The accuracy is calculated against the test set provided by Kaggle.

I expect to use 40% of the time understanding the data and feature extraction, 40% of the time to train and tweak the model and 20% of the time for final presentation etc..

References:

1. https://www.kaggle.com/c/quora-question-pairs
2. https://www.analyticsvidhya.com/blog/2017/01/ultimate-guide-to-understand-implement-natural-language-processing-codes-in-python/
3. https://www.quora.com/
4. https://towardsdatascience.com/document-feature-extraction-and-classification-53f0e813d2d3
5. https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html
6. https://scikit-learn.org/stable/modules/feature_extraction.html