Cloud Computing for Data Analysis Fall 2022 Project Proposal (Revised)

Title: ***Sentiment Analysis of Social Media Responses (Twitter) With Thorough Explanation Using Spark and Python***

Rezaur Rashid

Introduction

With the popularity of Big-data in recent years, we are experiencing rapid growth in data collection and they are becoming more available publicly and their usage in various applications is omnipresent which is creating new challenges in researcher communities [1]. Although several machine learning tools are being developed to deal real-time data processing, most of the traditional tools processes data in batches and struggle handling large-scale data.

Therefore, a cloud computing tools such as Spark, can provide the option to process such large-scale data as well as real-time computation in less time. Such tools can help large-scale data preprocessing, cleaning and aggregational computations.

Project Overview

For this project, we will implement a multiclass classification version of the Support Vector Machine (SVM) model to perform sentiment analysis of the user responses in social media (twitter) which has multiple classes. We know Apache Spark MLlib has two linear classification model: SVM and Logistic Regression (LR) and although LR can do multiclass classification, the MLlib SVM can only do binary classification. Therefore, the goal of this project is to develop a multiclass SVM classifier in a distributed fashion and also compare the model with LR classifier model.

Dataset

For the project, initial plan was to use the “Sentiment140 dataset with 1.6 million tweets” dataset from Kaggle [2]. This dataset provides user responses to different products, brands, or topics through user tweets on the social media platform-Twitter. The dataset was collected using the Twitter API and contained around 1,60,000 tweets. This dataset contains 6 attributes where two of them are polarity score (categorical) and text (tweets) we are mostly interested in. But while working on the project with this dataset, we found that the dataset only have 2 classes although the description says 3 classes.

New dataset: Later, we used the ‘Twitter US Airline Sentiment[3]’ that contains more than 14000 tweets data samples classified into 3 types: positive, negative, neutral. It contains 15 attributes but we have only used the sentiment class and the text (tweets) for our project.

Project Motivation

In a world of social-media and businesses, where people are expressing their emotional states, sentiment analysis plays a big role. It is an NLP application that helps to understand human behavior through text and speech. For example, using sentiment analysis tools, business organizations can learn about the emotional and behavioral state of customer using machine learning models which can enhance customer relationship and service.

And in the age of continuous data generation, cloud computing tools like Spark can help processing this large-scale real-world data with simple API, fast and distributed processing.

Task Involved:
The project work was divided into several steps which were briefly described below:

1. **Preparing data:** Real-life data specially social media data contains redundant information. Therefore, we needed to understand the data and have removed unwanted columns from the dataset. Given, the categorical target variable, we changed it to label encoded integers. [1: negative, 2: Neutral, 3: positive]. We only selected the 'text' column as our input feature. But social-media texts contain different formatted and irrelevant texts and characters. Therefore, we pre-processed the texts to remove special characters, alphanumeric characters, lower-cased every words etc.

2. **Feature transformation:** With our only feature, 'text' column in the dataset we then transformed each sentence into a new feature embedding. Because, the machine does not understand text and we needed to transform each sentence into something the model understand. We converted the text into vector embeddings. Namely, we have used something similar to 'bag of words' method. In out embedding each word is assigned with an unique number. We then split each word of the sentence with the help of tokenizer and assign the unique value to each word. But several row in the dataset has different length of words since the sentences are not equal. Therefore, we padded the sentences to have equal feature size.

3. **Model Development:** We know the SVM model works similar to logistic regression(LR) when the kernel is linear. The core difference is instead of assuming a probabilistic model like LR, we find a particular optimal separating hyperplane. We define "optimality" in the context of the support vectors and the SVM tries to maximize the margin between two classes.
   The idea behind multiclass SVM is similar to binary classification where in binary classification it is easy to get the marginal difference between two classes but in multiple classification the goal is to select a class that has a bigger marginal difference compared to other classes. This can be achieved by computing a score for each class, and the correct class will have high score then incorrect classes.

Experiment:

To experiment our project, we first implemented the model in our local machine where we have installed the PySpark library to create a local spark environment. We also check the script with the AWS spark machine but limited to read the data locally or from AWS S3 bucket since we faced problem copying the dataset into HDFS cluster.

We prepared the data using python libraries (e.g. sklearn, nltk, re) in the local machine using pyspark and regular python libraries. As mentioned above, we have used the airlines tweets dataset for the experiment since it has multiple target class. We also checked the performance with the sentiment140 dataset where we used 2000 samples.

For training the model, we split the data into train-test split where 70% used as traing and 30% for testing. We have used spark.DataFrame for reading the .csv file and then converted the data into spark RDD so then we can perform all the calculation in the distributed format.

Since our dataset is almost equally balanced, we have used accuracy to evaluate our model performance. For explanation of important features (i.e. important words contributing to the sentiment) we have calculated the top-3 important features from the model weights since we used an linear SVM model. From the top-3 features we then calculated the corresponding words that drives the prediction to have higher accuracy.

Results:
Although, we have tried initially with TF-IDF features extraction procedure, our linear model had around 35% accuracy for the Airlines_tweet dataset and 23% accuracy for the sentiment140 dataset.

The feature embedding mentioned above method has performed better for both dataset compared to other feature transformation. And our model has out-performed the sklearn Logistic regression model.

Our Model (SVM) Testing Accuracy: 72.83%
Logistic Regression Testing Accuracy: 53.93%

As for, feature explanation, our model found that out of 11129 unique words from the dataset, 6057 word has high feature importance to predict a sentiment. Below table shows some example words that have high importance.

```
['site', 'able', 'eat', 'anything', 'next', 'fail', 'miss', 'worry', 'together', 'very', 'can', 'get',
'any', 'cold', 'air', 'vents', 'noair', 'worstflightever', 'roasted', 'middle', 'red', 'eye',
'noob', 'maneuver', 'just', 'cool', 'birthday', 'add', 'elevate', 'entered', 'name', 'during',
'booking', 'problems', 'hours', 'operation', 'club', 'posted', 'online', 'current', 'left',
'expensive', 'headphones', 'one', 'answering', 'number', 'awaiting', 'return', 'phone',
'call', 'prefer', 'service', 'option', 'news', 'start', 'flights', 'hawaii', 'end', 'year', 'via',
```

Discussion:
In this project, we have developed a score-based method multi-class SVM classifier using Spark and Python. We used Spark DataFrame, RDD and IndexedMatrix to perform our calculation in distributed manner and compared our model performance with sklearn Logistic Regression model. Few challenges working with spark RDD we have faced is the limitation of

computational degrees of freedom. Also, data-preprocessing using pysaprk.ml libraries fails in local spark environment if the data generates large scale intermediate data.

References

1. Zhai, Y., Ong, Y. S., & Tsang, I. W. (2014). The emerging” big dimensionality”. IEEE Computational Intelligence Magazine, 9(3), 14-26.
2. Dataset https://www.kaggle.com/datasets/kazanova/sentiment140
3. Dataset https://www.kaggle.com/datasets/crowdflower/twitter-airline-sentiment