UE19CS322 - Big Data
Spark Streaming for Machine Learning
Sentimental Analysis

Team members:

Manish Reddy M **PES1UG19CS262**
Sethupathy RV **PES1UG19CS443**
Swaroop Bhat **PES1UG19CS532**
Gaurav J **PES1UG19CS599**

Design Details:

- To implement sentiment analysis we modeled the input sentences in which we removed the stop words using *RegexTokenizer*.
- The cleaned sentences were converted into a vector of numbers using word2vec. The label column was converted into a domain set of (0,1).
- A pipeline was utilized to perform the preprocessing where the dataset was fit and then transformed
- To implement streaming use a batch size of 1000.
- This input dataset is used while training to train 4 models namely
    1. Logistic regression model
    2. Stochastic gradient descent model
    3. Multilayer perceptron model
    4. K-means clustering model
- We initialize the models using global variables where we execute incremental learning in a loop.
- For each Resilient data distribution we update the weights of the particular model.
- This is achieved using the partial_fit function where the models learn in batches of batch size 500.
- For each model the accuracy, Recall, Precision, F1-score, Confusion Matrix is determined on the training set.

- Once all the RDD's are streamed the model is then saved as a pickle file. Now we stream the testing dataset.
- Using the saved models we then predict the required parameters. The mean accuracy is calculated across all batches.

Implementation Details:

- Data wrangling(transformations) to get the streamed data to a dataframe.
- As the Message is a string we need to convert it to a vector to allow the model to be fit.
- Batch sizes were tested and 1000 was found to be a good fit
- Term frequency and inverse document frequency is used to get a relative importance of each word.
- We tuned the hyperparameters to find the best possible accuracy
- We used both supervised and unsupervised classification algorithms and we found the supervised performed much better.

Reason behind the design decision:

- Models were chosen in a way they are compatible with incremental learning.
- As it is streaming data, for training the model it was necessary we had to implement an incremental model since it is impractical to train a model for every Dstream and the context would be lost between subsequent iterations.
- Models once it's done training had to be pickled because testing was done through a different file and a different stream.

Take Away from the Project:

- Big data is the future of large processing and this project provided a foray on large scale application.
- It was fascinating to gain exposure to actual tech that most tech giants use under the hood.
- As most machine learning algorithms are done incrementally we learnt how data is broken up and the state is saved between multiple iterations.
- We learnt classification and clustering algorithms and how they are used in batches.