Machine learning with Spark Streaming:

Dhananjay Kumar:PES1UG19CS140
Manogna GV : PES1UG19CS158
Gautam Mayya : PES1UG19CS164
Vaibhav S : PES1UG19CS555

Sentiment analysis is one of the most important parts of Natural Language Processing. Sentiment analysis is the prediction of emotions in a word, sentence or corpus of documents. It is intended to serve as an application to understand the attitudes, opinions and emotions expressed within an online mention. The intention is to gain an overview of the wider public opinion behind certain topics. Precisely, it is a paradigm of categorizing conversations into positive, negative or neutral labels.

About the data set:
The data set we have chosen is **sentiment** and this contains more than 40000 tweets in the test case.

Tools Used:
Apache Spark: It is an open source lightning fast cluster computing platform to retrieve streaming data and forwarding to storage system like HDFS, Database Server. It is built on top of Map Reduce and can integrate well with other Apache software.

Scala: It is not only a High Level Functional but also supports Object Oriented Programming language model. This provides it an edge over Java which requires more code to be written for the same task as compared to Scala. The major success of Scala is that Apache Spark is itself implemented in Scala.

Designing:
The first step in the design is to preprocess the data and convert the streaming data into rdd and from there to convert to dataframes. Then we

have to process the stream and Data processing involves Tokenization which is the process of splitting the data into individual words called tokens. Tokens can be split using whitespace or punctuation characters.
A tweet acquired after data processing still has a portion of raw information in it which we may or may not find useful for our application. Thus, this data is further filtered by removing stop words, numbers and punctuations.

Normalization is a little more complex than tokenization. It entails condensing all forms of a word into a single representation of that word. Next we have to use the Naive Bayes classification which is nothing but applying Bayes rules for forming classification probabilities.

Next, Sentimental Analysis is done using custom algorithm by finding polarity. For discovering the polarity, we used a simple algorithm of counting positive and negative words in data.

Take away from the project:
Through this project, we will be able to successfully process streaming data and convert it into data frames. Also we can recognize which one of the machine learning models is the best and most accurate for the data set.