**UE19CS322**

**BIG DATA PROJECT REPORT**

**TEAM ID : BD\_178\_333\_456\_900**

| **Hariharasudan S A** | **PES1UG19CS178** |
| --- | --- |
| **Prajna Udupa** | **PES1UG19CS333** |
| **Shilpitha R Shetty** | **PES1UG19CS456** |
| **Chinmayee Dindore** | **PES2UG19CS900** |

**PROJECT TITLE :**Machine learning with Spark Streaming

**Dataset Chosen** : sentiment

**DESIGN, IMPLEMENTATION DETAILS AND REASON BEHIND DESIGN DECISIONS FOR EACH UNIT :**

**Streaming :**

Since we are using pyspark, we import its class SparkSession to create a Spark Context.The server side here is the localhost and the client side is where we get the streaming data from the server batch-wise. As soon as we receive the batch from the socket , the data is preprocessed and text classification is performed . The StreamingContext object is created from the SparkContext object which is used to consume a stream of data in Spark . A SparkContext represents the connection to a spark cluster through which RDDs are created. Every 5 seconds a batch of data is streamed and later is preprocessed.

**Preprocessing:**

We preprocess the tweets so that we have the meaningful text of the tweet . In each batch, we applied some regex expressions to filter out the text. For example, the mentions, links, and special characters were removed in that way.

From the pyspark ML feature , we imported tokenizer , StopWordsRemover and HashingTF function to perform preprocessing of the texts in the tweet\_preprocessing function which returns the preprocessed tweets and it’s numeric weights.We also used SnowballStemmer from nltk for reducing the word to it’s base word . The final preprocessed tweets were then vectorized using the HashVectorizer provided by sklearn which in turn helps us building the model.

**Building the model and testing:**

Here we used 3 models for classification of the texts:

1. BernoulliNB model
2. Perceptron model
3. SGD classifier model

We split the data into training and testing data using the train\_test\_split method imported from sklearn.model\_selection.The reason why we use sklearn is that it provides us incremental learning models which MLib didn’t provide.

With a help of pickle module we saved the models batch- wise while training to a .sav file. Then we tested our models by loading them onto the test file.

We obtained substantial results from our models. The inferences have been provided in our github repository.

**Clustering :**

For clustering we used the MiniBatchKmeans which is used for incremental learning. We used the algorithm for each batch while training and also tested it out to find the clusters in our test file.

**TAKE AWAY FROM THE PROJECT:**

We were able to analyse the working of each of the classification model through the large batches of data via incremental learning.