UE19CS322: BIG DATA PROJECT

MACHINE LEARNING USING SPARK STREAMING

NAME:	SRN:
Rishab K S	PES1UG19CS384
Pranav M R	PES1UG19CS340
K K Tarun Kumar	PES1UG19CS222
Rakeshgowda D S	PES1UG19CS376

Design Details:

- This project aims to understand and analyze machine learning tasks on large data streams. We have chosen the “Sentiment140” dataset.
- The CSV dataset is encoded into JSON and then it is streamed via TCP socket. Then the stream is received using the readstream() function with the source as a socket.
- After receiving a batch of data, it is Pre-Processed.
- After Pre-Processing we are building models for clustering and classification -
  Classification models – SGD Classifier, PassiveAggresive Classifier
  Clustering Model – Mini-batch K-Means
- The aforementioned models are trained on each batch of data.
- After training the models on the whole dataset we use them to predict the test batches.
- F1 score, precision, recall, and accuracy are calculated.

Surface Level Implementation Details:

- The Streaming Dataframe is converted to Static Dataframe using foreachBatch() operation which allows us to use any functions on the dataframe.
- Pre-Processing: The batch of data is filtered to remove special characters, digits and Twitter handles.

- For Classification, HashingVectorizer is used which maps string to feature integer index.
- We are using SGD Classifier and PassiveAggressive classifier from the sci-kit learn library for classification tasks as these classifiers support incremental learning. We perform incremental learning by using the partial_fit() function on the training dataframe.
- For clustering, TF-IDF Vectorizer converts string to a matrix of TF-IDF features.
- The feature decomposition is performed using TruncatedSVD to obtain the principle features.
- The output of the Truncated SVD is normalized and used to incrementally train the Mini-batch K-Means model on each batch of data.

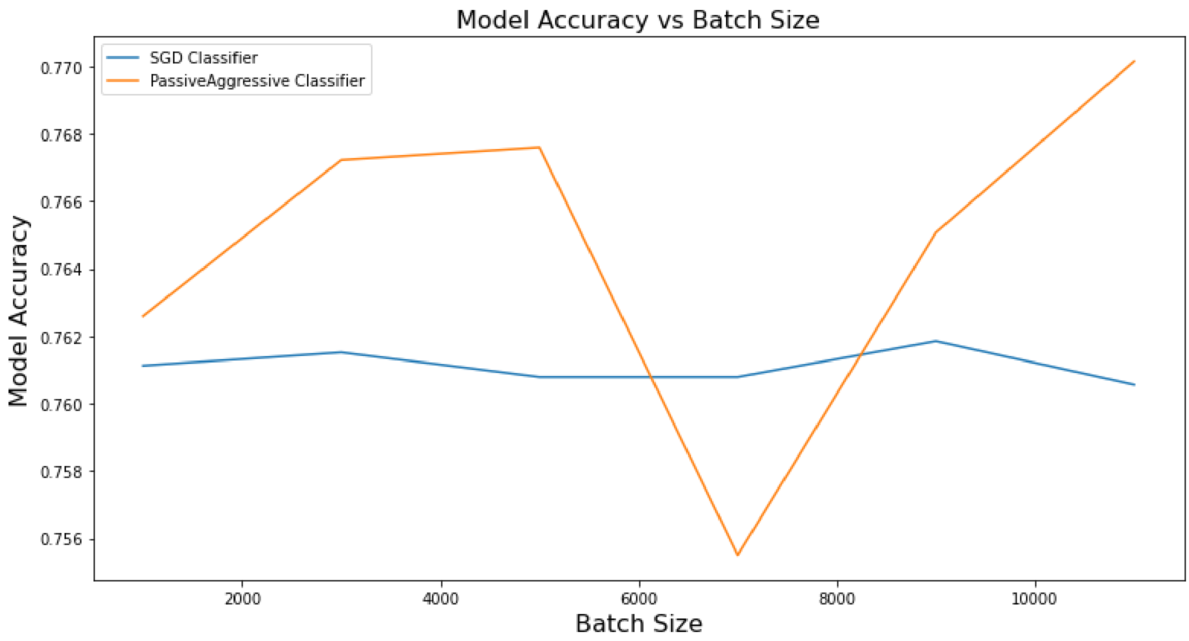
The Motive behind the Design Decisions:

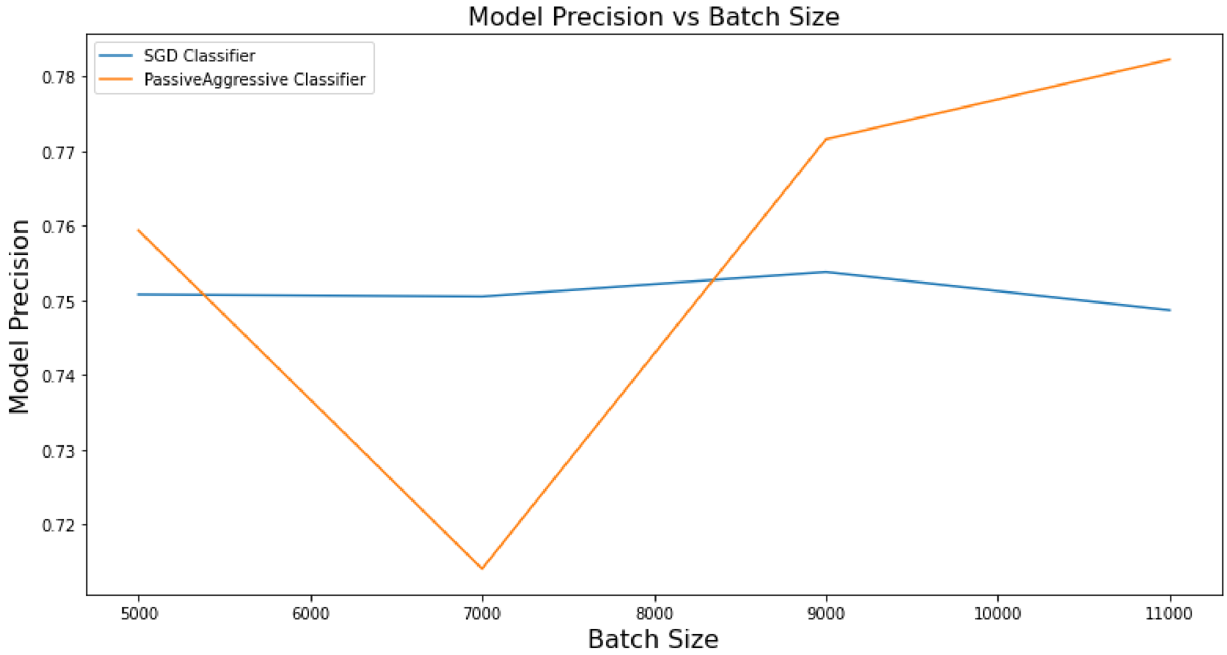
- SGD and PassiveAggressive models are used since they provide higher accuracy compared to Naïve Bayes learners.
- Mini-Batch K-means supports incremental clustering.
- Hashing Vectorizer is used since the complete dataset is unavailable.

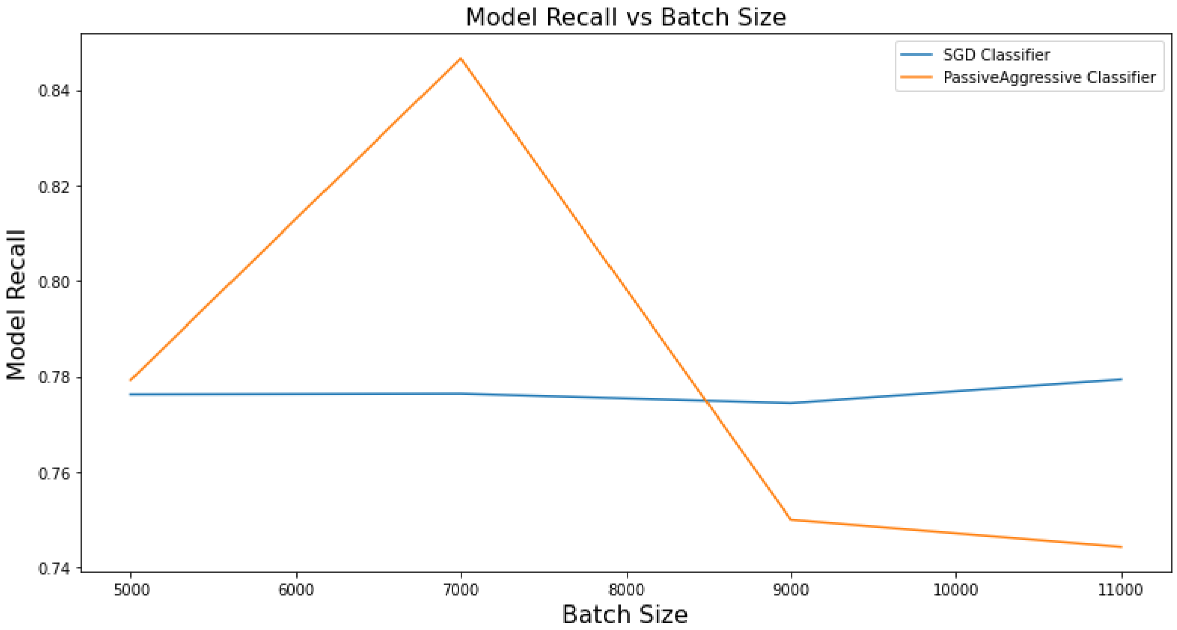
Knowledge acquired from the project:

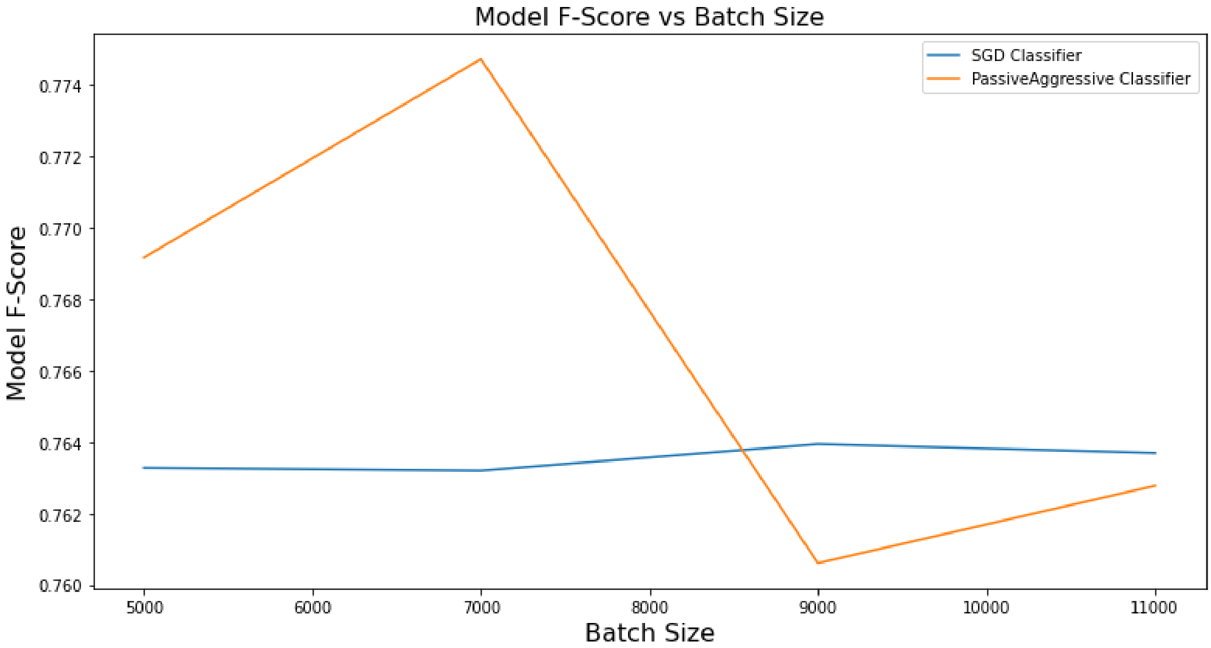
- Working with streaming data using pyspark
- Incremental Machine learning models
- Effect of varying batch sizes on model metrics

Analysis:

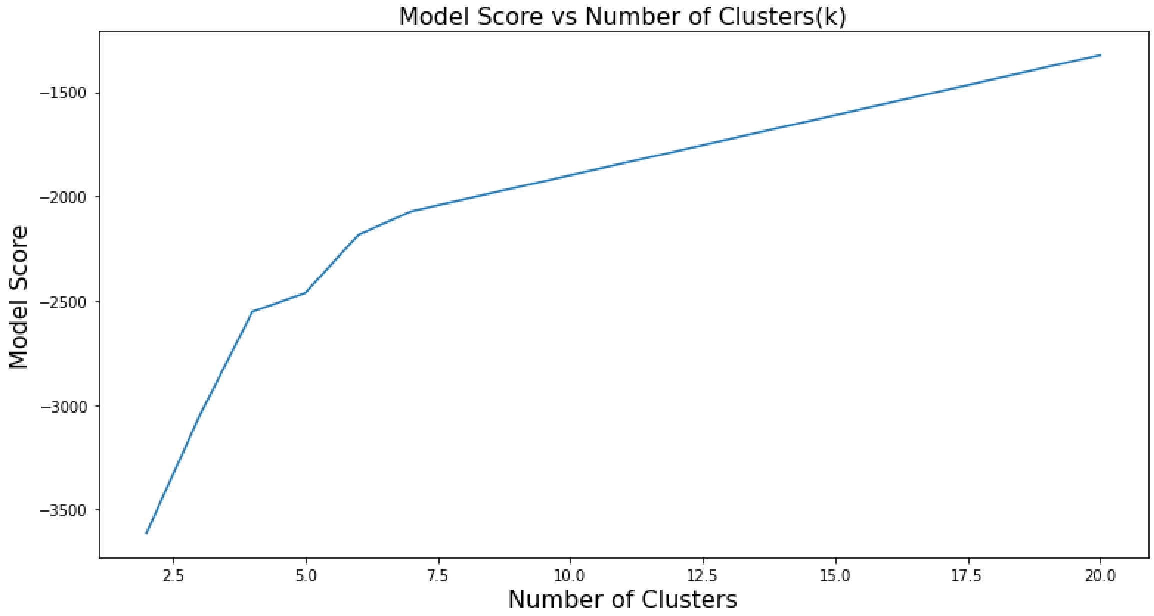

Model Accuracy vs Batch Size
SGD Classifier
PassiveAggressive Classifier
Model Accuracy
0.770
0.768
0.766
0.764
0.762
0.760
0.758
0.756
2000
4000
6000
8000
10000
Batch Size


Model Precision vs Batch Size
SGD Classifier
PassiveAggressive Classifier
Model Precision
0.78
0.77
0.76
0.75
0.74
0.73
0.72
5000
6000
7000
8000
9000
10000
11000
Batch Size





- Accuracy, Precision, Recall, and F1 Score have less variation for SGD Classifier for all batch sizes.
- For PassiveAggessive Classifier:
  - While Accuracy and Precision dip to an all-time low at batch size = 7000, they later increase as the batch sizes increase.
  - F1 Score and Recall peaks at batch_size = 7000 and later decrease as the batch sizes increase.



- Elbow Method is used to find the optimal K-Value (5-7).