Streaming Section

Process:

Luckily our process was quite straight-forward since we did most of the hard work in the section prior which include data exploration, feature engineering, model selection, model tuning...

However, since it's our first time trying Spark Structured Streaming we ran into many technical problems:

- **Parameters:**We discovered that the CrossValidation we did in the previous section doesn't achieve the same desired effect on new data, so in the beginning we performed CV while streaming and found out that each time we got the same parameters which are
  $$\text{maxDepth} = 20$$
  $$\text{numTres} = 5$$
  and these are the parameters that were used during the streaming.

- **Memory problems:** after running the code multiple times we encountered crashes when the training dataframe reaches ~1M entries, so we implemented a mechanism such that after we reach 1M entries we would take a **random** sample of the data and use it as training data, to then hopefully perserve the information from the data while also using memory efficiently. notice that we also used `df.unpersist()` to delete the remaining dataframe. This solution worked wonderfully.

- **Misc.** since this part of the project was possible only on the dedicated server we ran into many problems that weren't under our controls such as server failiure and congestion. Also by the nature of the tasks we are performing this task took a lot of time since we're working with great amounts of data.

The following code and plot provide details of a run we had done before the submittion, unfortunatly we didn't have time (not the ability) to read the 6M entries of data stored on the server, but we were able to get results of the first 5 iterations, where in each iteration a batch with a total of 500,000 entries were processed, predicted upon, trained upon and ready for the next iteration

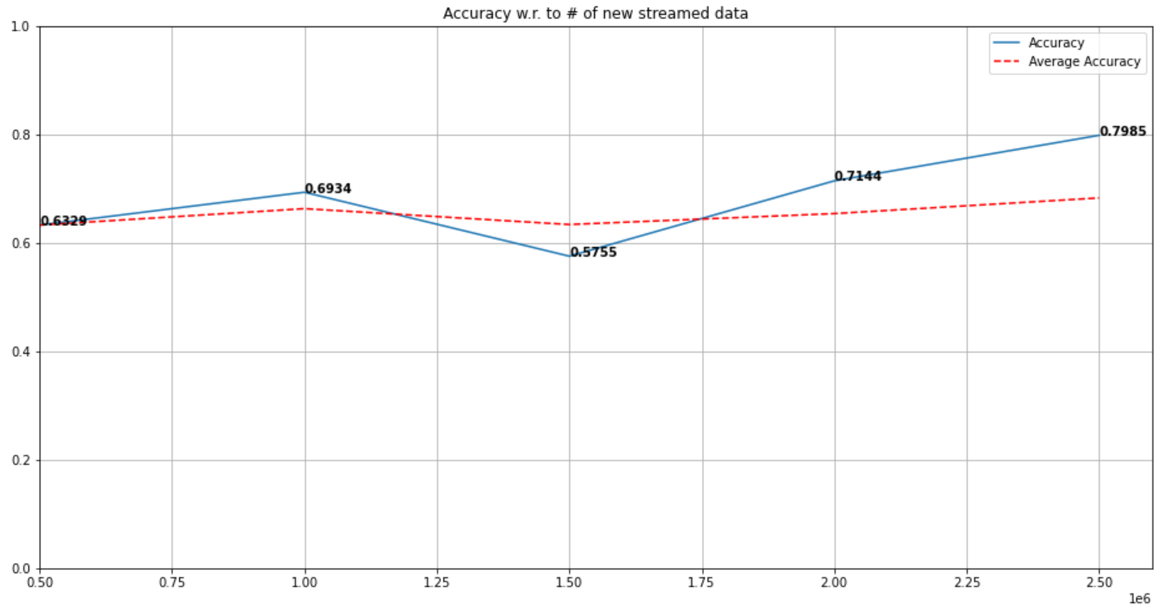
The run is provided here:

```
LogAggregationType: LOCAL
==========================================================================
====================================================
LogType:stdout
LogLastModifiedTime:Thu Sep 22 18:22:21 +0000 2022
LogLength:80782
LogContents:
after spark init
after norm
after functions
after read stream
inside process data
Current train size:  546083
This is batch number:
0
inside process data
New data in batch recieved  500000
New Data that has been predicted on since start :  500000
Accuracy for current batch = 0.6328518244072034
Current accuracy mean is  0.6328518244072034
Current train size after union 1046083
Model retrained!
Training data too big, scaling it down!
------------------------ITER DONE------------------

Current train size:  732586
This is batch number:
1
inside process data
New data in batch recieved  500000
New Data that has been predicted on since start :  1000000
Accuracy for current batch = 0.6934406981129726
Current accuracy mean is  0.663146261260088
Current train size after union 1232586
Model retrained!
Training data too big, scaling it down!
------------------------ITER DONE------------------

Current train size:  863013
This is batch number:
2
inside process data
New data in batch recieved  500000
New Data that has been predicted on since start :  1500000
Accuracy for current batch = 0.5754865520275354
Current accuracy mean is  0.6339263581825705
Current train size after union 1363013
Model retrained!
Training data too big, scaling it down!
------------------------ITER DONE------------------
```

```
Current train size:  954547
This is batch number:
3
inside process data
New data in batch recieved  500000
New Data that has been predicted on since start :  2000000
Accuracy for current batch = 0.714446674018459
Current accuracy mean is  0.65405643714154 26
Current train size after union 1454547
Model retrained!
Training data too big, scaling it down!
```

In [38]:

```
import matplotlib.pyplot as plt

batches = [500000, 1000000, 1500000, 2000000, 2500000]
acc = [0.6328518244072034, 0.6934406981129726, 0.5754865520275354, 0.714446674018459, 0.79845736257]
cum_acc = [0.6328518244072034, 0.663146261260088, 0.6339263581825705, 0.6540564371415426, 0.6829366222272342]

fig, ax = plt.subplots()
fig.set_size_inches((16,8))

plt.title("Accuracy w.r. to # of new streamed data")
plt.plot(batches, acc,label="Accuracy")
plt.plot(batches, cum_acc,"--r",label="Average Accuracy")
plt.ylim(0,1)
plt.xlim(500000)

for i_x, i_y in zip(batches, acc):
    plt.text(i_x, i_y, '{:.4f}'.format(i_y), weight='bold')

plt.legend()
plt.grid(True)
```


Accuracy w.r. to # of new streamed data
Accuracy
Average Accuracy
1e6