Part b

Question 4

Hyper-Parameters of ALS-

- ***numBlocks***
- ***rank***
- ***maxIter***
- ***regParam***
- ***implicitPrefs***
- ***alpha***
- ***nonnegative***

Hyper-Parameters used for tuning-

Cmd 21

```
parameters = (ParamGridBuilder()
              .addGrid(als.regParam, [0.01, 0.5, 1])
              .addGrid(als.maxIter, [5, 10, 20])
              .addGrid(als.rank, [5, 10, 20])
              .build())
```

Command took 0.07 seconds -- by vishal.kanna@mail.utoronto.ca at 6/16/2021, 11:00:51 AM on firstCluster

Cmd 22

1. Reg param- the regularization hyper-parameter in ALS- used in the l2 regularization that prevents the model from overfitting. This hyper-parameter is scaled for every feature.

2. Max iter- the maximum number of learning iterations to run. Max iteration that is too low can lead to a very poor model as the model is not able to converge.

3. Rank- the number of latent or hidden factors in the model. These are factors that are not known to us but found by the algorithm as the model learns the data. This hyper-parameter can be tuned so that the model can have a greater number of hidden features making more complex connects between the user and the item.

Hyper-Parameters of the model after tuning-

```
1  cv = CrossValidator(estimator=als, estimatorParamMaps=parameters, evaluator=eval, numFolds=2)
```

Command took 0.07 seconds -- by vishal.kanna@mail.utoronto.ca at 6/18/2021, 11:06:10 AM on firstCluster

Cmd 22

```
1  model = cv.fit(training)
```

▸ (9) Spark Jobs

Command took 27.21 minutes -- by vishal.kanna@mail.utoronto.ca at 6/18/2021, 11:06:21 AM on firstCluster

Cmd 23

```
1  best_model = model.bestModel
```

Command took 0.03 seconds -- by vishal.kanna@mail.utoronto.ca at 6/18/2021, 11:40:29 AM on firstCluster

Cmd 24

```
1  print("**Best Model**")
2  # Print "Rank"
3  print("  Rank:", best_model._java_obj.parent().getRank())
4  # Print "MaxIter"
5  print("  MaxIter:", best_model._java_obj.parent().getMaxIter())
6  # Print "RegParam"
7  print("  RegParam:", best_model._java_obj.parent().getRegParam())
```

```
**Best Model**
  Rank: 10
  MaxIter: 10
  RegParam: 0.1
```

Command took 0.04 seconds -- by vishal.kanna@mail.utoronto.ca at 6/18/2021, 11:40:34 AM on firstCluster

Evaluating the best model with 80/20 split-

Cmd 25

```
1  final_predictions = best_model.transform(test)
```

▸ final_predictions: pyspark.sql.dataframe.DataFrame = [movieId: integer, rating: integer ... 2 more fields]

Command took 0.05 seconds -- by vishal.kanna@mail.utoronto.ca at 6/18/2021, 11:40:44 AM on firstCluster

Cmd 26

```
1  RMSE = eval.evaluate(final_predictions)
2  print(RMSE)
```

▸ (5) Spark Jobs

```
1.1031829802818578
```

Command took 3.99 seconds -- by vishal.kanna@mail.utoronto.ca at 6/18/2021, 11:40:48 AM on firstCluster

We can see the RMSE has improved from 1.150 from our prior question to 1.103 with our tuned hyper-parameters.