This exam serves as the assessment for those students who cannot utilize the Hadoop system and/or Ambari GUI.

1. Which of the Spark components most interests you and why?

Spark Components :

* **Spark Core:** This is the base program for Spark. It is also known as *Spark 1.0*.
* **Spark Streaming:** Allows you to feed in real-time data and provide real-time output.
* **Spark SQL:** Write SQL queries and use SQLite functions in Spark. This is part of *Spark 2.0.*
* **MLLib:** A library of machine learning and data mining tools you can use in Spark. This is also part of *Spark 2.0*.
* **GraphX:** Allows you to create social network graphs and determine the degrees of separation in your data.

**MLLib looks like a lot of help.**

Today many companies focus on building customer-centric data products and services which need machine learning to build predictive insights, recommendations, and personalized results. Data scientists can solve these problems using popular languages like Python and R, but they spend a lot of time in building and supporting infrastructure for these languages. Spark has built-in support for doing machine learning and data science at a massive scale using the clusters.

MLlib is a low-level machine learning library. It can be called from Java, Scala and Python programming languages. It is simple to use, scalable and can be easily integrated with other tools and frameworks. MLlib eases the deployment and development of scalable machine learning pipelines. These are some of the important features and capabilities Spark MLLib offers:

* Linear regression, logistic regression
* Support Vector Machines
* Naive Bayes classifier
* K-Means clustering
* Decision trees
* Recommendations using Alternating Least Squares
* Basic statistics
* Chi-squared test, Pearsons or Spearman correlation, min, max, mean, variance
* Feature extraction
* Term Frequency/ Inverse Document Frequency useful for search

1. True or False? “Zeppelin is very similar in structure and function to Jupyter Notebook.”
   1. True
   2. False
2. How do the three types of Spark data storage differ from each other?

# Spark Data Storage

There are three main ways that data can be stored in Spark:

* Resilient Distributed Datasets (RDDs)
* DataSets
* DataFrames

RDDs are an artifact of Spark 1.0.

DataSets and DataFrames, are part of Spark 2.0's architecture.

 RDDs are a type of data storage that distribute the data across the Hadoop cluster, but they are somewhat slow.

DataSets are similar to RDDs, but they speed up the process, by utilizing more efficient memory representation in Spark.

A DataFrame is a subclass of a DataSet, and they are specifically meant for relational data. A DataFrame keeps ahold of rows and columns in Spark, which allows us to do more with our data.

1. How do you denote comments in Scala?
   1. #
   2. Change it to markdown
   3. //
   4. /#
2. What are the four hyperparameters for decision trees? Give both names and descriptions.

A decision tree has the following hyperparameters:

* **Maximum depth:** Limits the number of decisions you can make in a decision tree. Sometimes having too many can lead to overfitting of data.
* **Maximum bins:** Limits the number of decision rules the decision tree can have. A lot will probably make your decision tree more accurate, but could take up too much processing power.
* **Impurity measure:** *Purity* is how good you are at classifying accurately. If you have two categories, and each group only contains the appropriate data for that category, then you have complete purity. If you have some mix-up in there, then you have *impurity*. You want to have low impurity.
* **Minimum information gain:** If you include a decision rule that does not make the data more pure, then what good is it? Including a hyperparameter of minimum information gain allows you to only keep levels that will actually add to the accuracy of your data, and it can help ensure you don't overfit the model.