Lab 5

**Due Date: Please check the class schedule on Blackboard.**

**Machine Learning Using Spark MLlib with Python on Databricks**

## Tools:

1. [**Databricks Community**](https://community.cloud.databricks.com/) **Edition** (<https://community.cloud.databricks.com/>)
2. **Apache Spark** (<https://spark.apache.org/>)
3. **Spark MLlib** (<https://spark.apache.org/docs/latest/ml-guide.html>)
4. **PySpark (**<https://spark.apache.org/docs/2.4.0/api/python/pyspark.html>**)**
5. **PySpark SQL (**<https://spark.apache.org/docs/2.4.0/api/python/pyspark.sql.html>**)**

**Resources\***:

1. Dr. Liao’s tutorials, code snippets, and programming hints in the lab
2. **Databrick Spark MLlib tutorials/code examples:** [*Getting started with MLlib.\**](https://docs.databricks.com/_static/notebooks/getting-started/get-started-with-mllib-dbr7.html)
3. Dr. Liao’s lecture notes/demonstrations
4. Other sources on Internet

*\*Note that you must include* ***reference(s)*** *in your program when you refer others’ code.*

**Data/Code Examples/Lab Templates Location**: Class10/Lab5/\*

**Instructions**:

This lab is specially designed for you to learn **Spark MLlib** with Python on the Databricks platform **step by step**, including tools for data preprocessing, Machine Learning (ML) pipelines, several different machine learning algorithms, and corresponding ML applications.

This lab includes two tasks/**mini-projects**: 1) **market basket analysis and recommendation**, and 2) **binary classification using logistic regression**. All of these are very helpful for your final project. More specifically, in this lab, you will learn

* + How to **preprocess** data with Spark MLlib
  + How to use **machine learning pipelines** for optimization with Spark MLlib
  + how to conduct *advanced* **data analytics** with Spark MLlib
    - Association rules
    - Classification
    - Hyperparameter tuning and cross validation
    - Market basket analysis
    - Product recommendation
  + how to **embed SQL query** in Spark on Databricks

In this lab, **Databricks Runtime 7.3 ML or above** is required for creating a new **cluster** to run

**Spark MLlib** application programs.

# Task 1 (40 points + Extra Credit 20 points): Market Basket Analysis and Recommendation

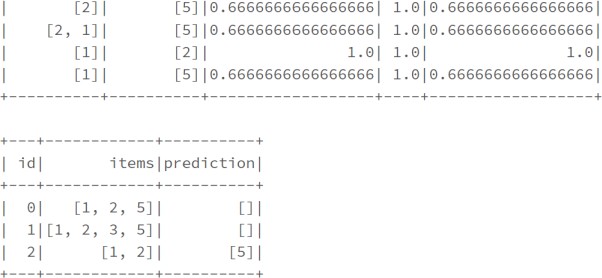
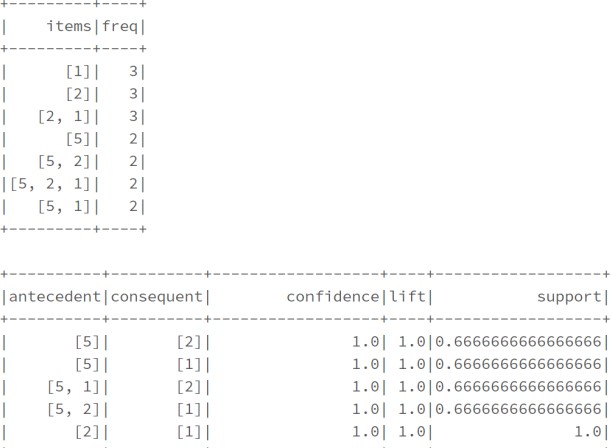
The goal of this task is to use Spark MLlib to build a model to generate association rules to quickly run the market basket analysis to uncover associations between different items, then further to provide recommendations for purchase on a distributed platform.

Please follow the following **step-by-step** instructions to write the PySpark code with Spark MLlib to complete Task 2:

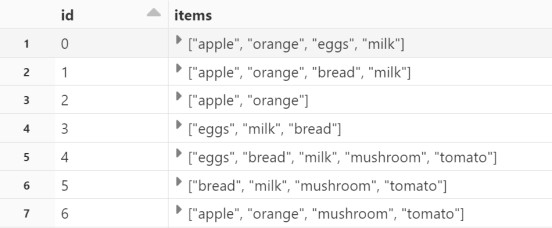
* 1. **(2 points)** Create a new notebook on Databricks, *AIT614-Sec#\_****Lab5.1****\_YourName.ipynb,* and write your full name and course# with section# on the notebook.
  2. **(3 points)** Please read [Spark MLlib Main Guide - **Frequent Pattern Mining**](https://spark.apache.org/docs/latest/ml-frequent-pattern-mining.html) and copy **FP-Growth**

Python code example into the notebook and run it.

*Hint: The outputs should be:*



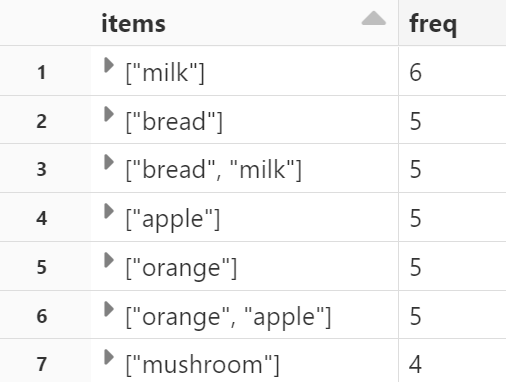
* 1. **(10 points) Create a training set**. Create your own dataset for training. For example, you can create different kinds of food items, and then display this dataset. The records/rows of the dataset should be >= **15**.

*Hint: For example, the output of some rows could be:*

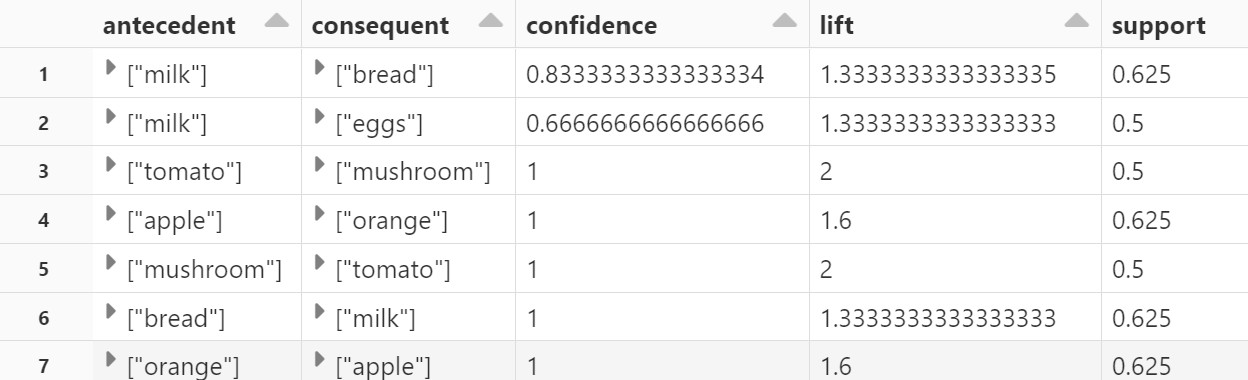
* 1. **(5 points) Use FP-Growth to build a model**. You can set different *minSupport* and *minConfidence*

values to test the model.

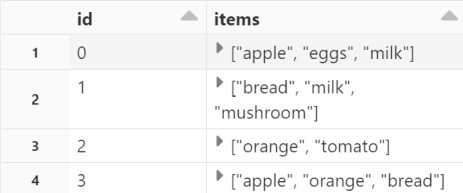
* 1. **(2 points)** Display frequent itemsets.

*Hint: For example, the output could be:*

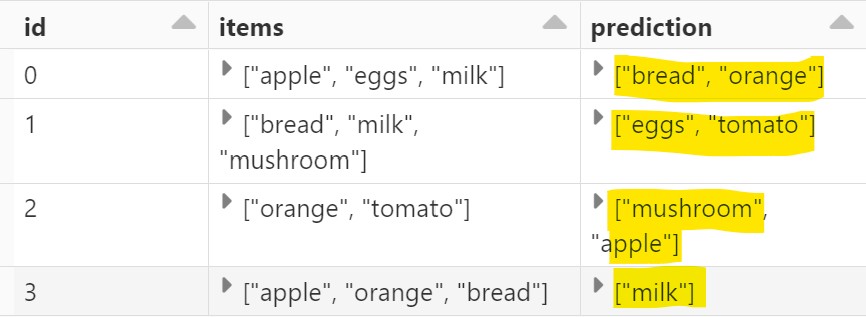
* 1. **(3 points)** Display generated association rules.

*Hint: For example, the output could be:*

* 1. **(5 points) Create a test set**. Create your own dataset for testing. The rows of the dataset should be >= **5**.

*Hint: For example, the output could be:*

## (5 points) Make recommendation/prediction.

*Hint: For example, the output could be:*

* 1. **(5 points)** List citations/references to appreciate other’s work in a professional manner.
  2. **(Extra Credit 20 points)** Load a real-world dataset and split into training and test sets then generate corresponding outputs.

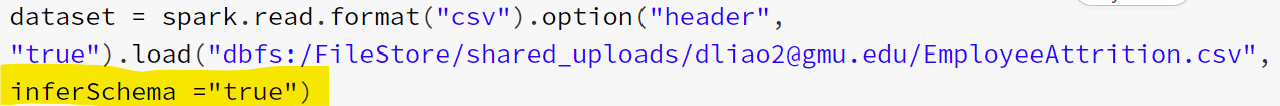
# Task 2 (60 points): Binary Classification using Logistic Regression

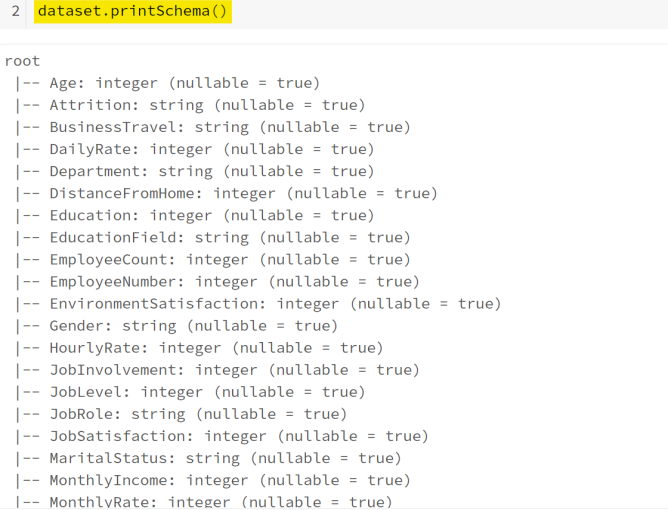
The goal of this task is to build a machine learning pipeline including a classification model that predicts the `Attrition` (Yes or No) from the features included in the dataset (income, work years, education level, marital status, job role, and so on), which we used in the Lab 3 and Lab 4.

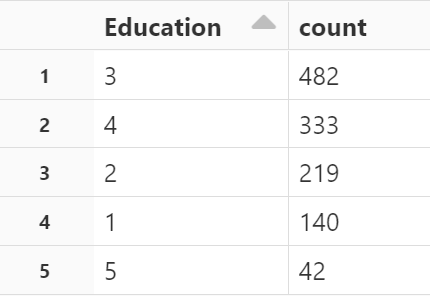
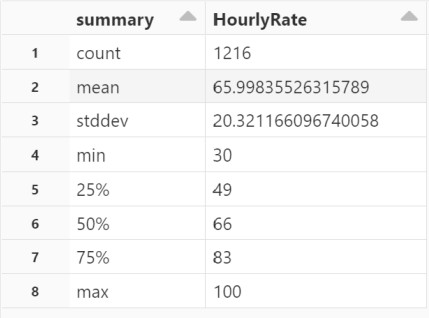
Please follow the below **step-by-step instructions** and Databrick Spark MLlib tutorials and code examples: [*Getting started with MLlib.\**](https://docs.databricks.com/_static/notebooks/getting-started/get-started-with-mllib-dbr7.html) to write the PySpark code with MLlib to complete the Task 1:

* 1. **(2 points)** Create a new notebook on Databricks, *AIT614-Sec#\_****Lab5.2****\_YourName.ipynb,* and write your full name and course# with section# on the notebook.
  2. **(5 points)** Please follow the Lab 4 instruction to read the data file *EmployeeAttrition.csv* into the notebook and display the dataset schema.

*Hint: Please add* ***inferSchema ="true"*** *in the file reading. The code snippets should be similar to:*

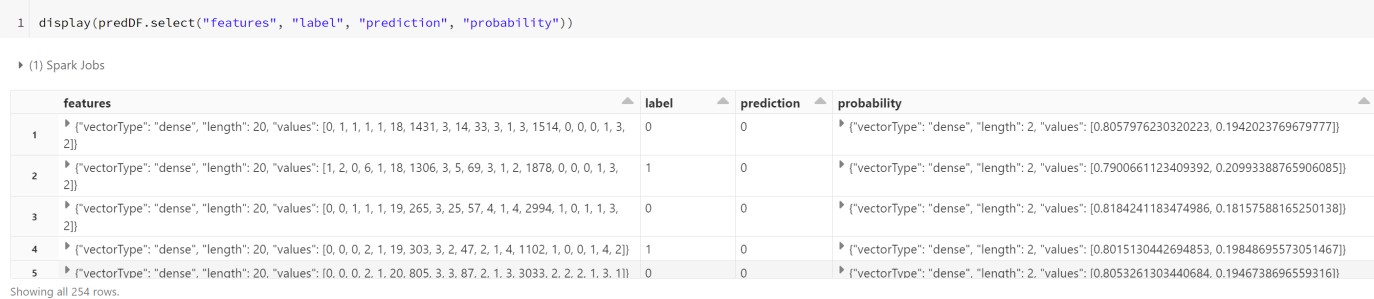




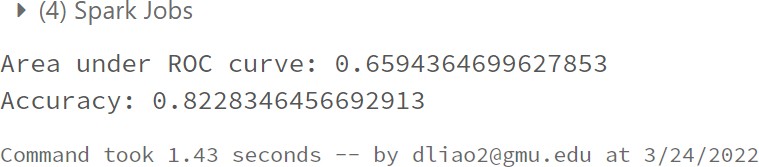
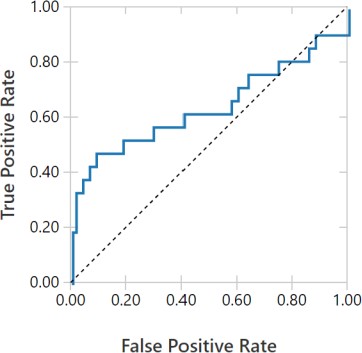
* 1. **(5 points)** Randomly split data into training and test sets. Display the distribution of the number of HourlyRate using *summary()* and Education using *groupBy()* in the training set. *Hint: The outputs are similar to:*

## (12 points) Feature preprocessing.

* + 1. (5 points) In the displayed schema, you can find the categorical features with string data type. Select five categorical features *"Department", "EducationField”, “Gender", "JobRole", and "MaritalStatus"* and convert to index vector using *stringIndexer()* only.
    2. (2 points) Set Attrition feature (Yes or No) as a label.
    3. (3 points) Select these numeric features: *"Age", "DailyRate", "Education", "DistanceFromHome", "HourlyRate", "JobInvolvement", "JobLevel", "JobSatisfaction", "MonthlyIncome", "YearsAtCompany", "YearsInCurrentRole", "YearsWithCurrManager", "NumCompaniesWorked", "PerformanceRating", "EnvironmentSatisfaction"*
    4. (2 points) Combine all feature columns into a single feature vector.
  1. **(2 points) Define the model**. Use a logistic regression model *LogisticRegression()*.
  2. **(8 points) Build the pipeline**. Define a pipeline using *pipeline()* and then apply the pipeline model to the test dataset. Note that in this step, it may take a few minutes to build. Then display the predictions from the model.

*Hint: Note that only FOUR stages in the pipeline. The output is similar to:*

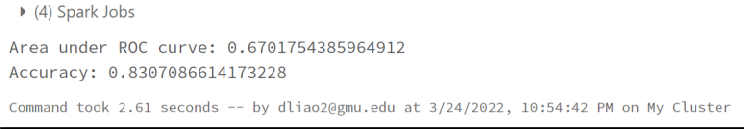
* 1. **(3 points) Evaluate the model**. Plot the ROC curve and print Area under Roc curve and Accuracy. *Hint: The outputs are similar to:*



* 1. **(3 points) Hyperparameter tuning and cross validation.** Use *ParamGridBuilder()* and

*CrossValidator()* to tune the model.

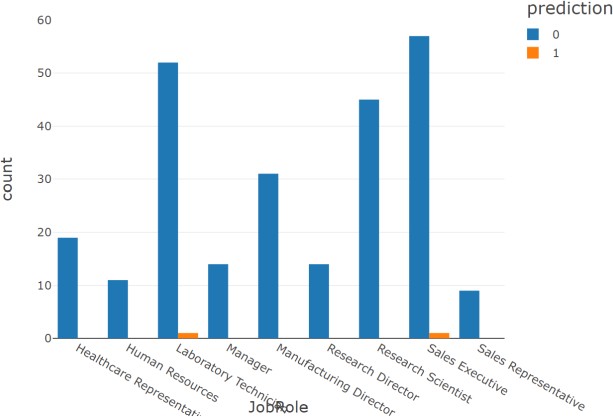
## (3 points) Make predictions and evaluate model performance.

*Hint: The output is similar to:*

## (12 points) Use SQL commands.

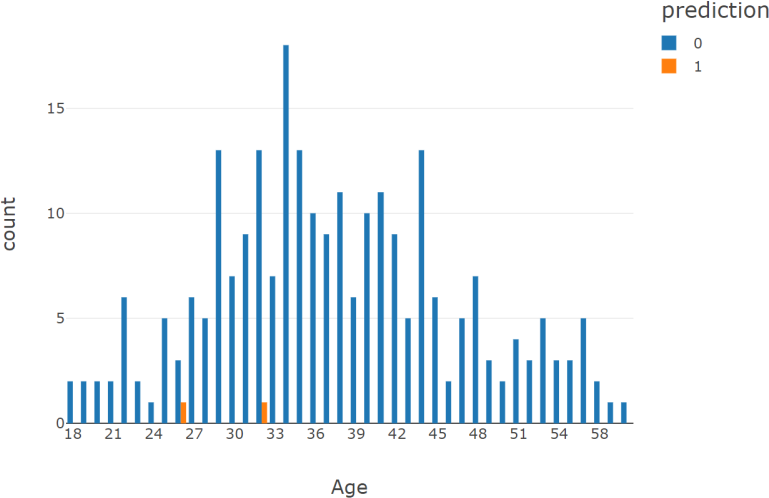
* + 1. (2 points) create a temporary view of the predictions dataset.
    2. (5 points) Display predictions grouped by Age and JobRole in a bar chart.

*Hint: The output is similar to:*



* + 1. (5 points) Display prediction grouped by Age and prediction in a bar chart.

*Hint: The output is similar to:*



* 1. **(5 points)** List citations/references to appreciate other’s work in a professional manner.

# SUBMISSION

## Run ALL Cells:

In Databricks Notebook, **Run All**. Please make sure all of you code has run and printed out the results.

## Expore to IPython Notebook and HTML:

Go to **File**-> **Export** -> **IPython Notebook** and **HTML**, and save your work into **IPython Notebook** and HTML files.

## Submit to Blackboard:

* + Follow the code template to write your work and output four files: **“AIT614- Sec#\_Lab5.1\_YourNames.ipynb”**, **“AIT614-Sec#\_Lab5.1\_YourNames.HTML”, “AIT614- Sec#\_Lab5.2\_YourNames.ipynb”,** and **“AIT614-Sec#\_Lab5.2\_YourNames.HTML”**.
  + Zip all the notebook and HTML files to **ONE zipped** file since blackboard does not allow you to submit HTML file separately.
  + Go to **Blackboard/Course Content/Class10/Assignments/Lab 5 Submission** to submit **ONE zipped file**.