Data is taken from kaggle and is available in HDFS as csv file at ‘*/user/datasets/B31PHD/phd_dataset.csv*’

*First row in the dataset is the header.  Treat fields with the value ‘NA’ as null values.*

**Dataset Description:**

| Column_Name | Description | Type |
|---|---|---|
| ROW_ID | Sequence number | Numeric |
| CASE_STATUS | Status associated with the last significant event or decision. Valid values include “Certified,” “Certified-Withdrawn,” Denied,” and “Withdrawn”. | String |
| EMPLOYER_NAME | Name of employer submitting labor condition application. | String |
| SOC_NAME | Occupational name associated with the SOC_CODE. SOC_CODE is the occupational code associated with the job being requested for temporary labor condition, as classified by the Standard Occupational Classification (SOC) System. | String |
| JOB_TITLE | Title of the job | String |
| FULL_TIME_POSITION | Y = Full Time Position; N = Part Time Position | String |
| PREVAILING_WAGE | Prevailing Wage for the job being requested for temporary labor condition. The wage is listed at annual scale in USD. The prevailing wage for a job position is defined as the average wage paid to similarly employed workers in the requested occupation in the area of intended employment. The prevailing wage is based on the employer’s minimum requirements for the position. | Numeric |
| YEAR | Year in which the H-1B visa petition was filed | Numeric |
| WORKSITE | City and State information of the foreign worker's intended area of employment | String |
| lon | longitude of the Worksite | Numeric |
| lat | latitude of the Worksite | Numeric |

Objective is to classify/predict the **CASE_STATUS** using the machine learning model in pyspark.

Perform the below steps:

1. Create a dataframe for above csv data.
   (First line in the dataset is header, comma is a part of data in few fields, fields are escaped with double-quote(“)).

2. Verify summary of the dataframe (how many rows and columns).

3. Derive the summary statistics.

4. Find the count of distinct values in each column.

5. List EMPLOYER_NAME and YEAR in the descending order of the Approved applications count (Approved applications are obtained using
   CASE_STATUS = 'CERTIFIED').

6. Observe the above results and list the EMPLOYER_NAME that had the maximum approved applications for each year for 2012, 2013, 2014, 2015 and 2016?

7. List the approved applications count in the descending order for the JOB_TITLE = "DATA SCIENTIST" and for each employer and year.

8. Find the null values count in each column.

9. Remove all the rows with null values (in any column/position).

10. Verify the null values count in each column.

11. List the count of applications in each status (CASE_STATUS) in the descending order of the year.

12. Find the mean PREVAILING_WAGE for each year for the approved applications.

13. Find the mean PREVAILING_WAGE for each year for the approved applications for each employer.

14. Find the approved applications count in each year for the full time positions in the descending order of the year.

15. Identify the different levels/labels/classes/categories in CASE_STATUS field and generate indexes, for each class/label/category starting from 0 in a new column named ‘label’.

    If you’re interested in deriving this using udf, below is the sample code.

```
from pyspark.sql.functions import udf

def catToValue(category):
     if   category == 'CERTIFIED': return int(0)
     elif category == 'CERTIFIED-WITHDRAWN': return int(1)
     ..
     ..
     else: return int(n)

 udfcatToValue = udf(catToValue, IntegerType())
 df = df.withColumn("label", udfcatToValue("CASE_STATUS"))
 df.show()
```

16. Dummy categorical variables (using String Indexer and One Hot Encoder).

17. Create a new column ‘features’ (feature vector for all the columns used for the model building) by combining the vectors created for the categorical variables and numerical features.

    Note: You may face “OutOfMemoryError: Java heap space” error,
    Invoke pyspark shell by specifing a larger RAM value.
    For example: pyspark --driver-memory 15g

18. Perform train and test splits.

19. Build at least two machine learning models.

20. Derive train and test accuracies for both the models.

Note:

1. Export/Note all your pyspark commands into a **text file(.txt only)** and upload to grader tool under **Big Data Submission** task with the file naming convention **B31PHD_<your_enrollment_id>_<first_name>_<last_name>.txt**
   Ex. B31PHD_1234_Abc_Xyz.txt