Classifiers using PySpark

The goal of the assignment is to build classifiers on “Adult Census Income” dataset using PySpark. The dataset will be given to you. It contains demographic information about individuals, such as age, education, occupation, marital status, and more. The target variable in this dataset is the income level of individuals, categorized into two classes: whether an individual earns more than $50,000 per year or not. You are expected to build a binary classifier on the dataset.

Do the following tasks on the dataset

Preprocessing steps:

1. Preprocess the dataset suitable to build the models
2. If you notice that feature selection may help to build a better model, implement appropriate feature selection methods.

Pipeline building steps:

3. For the categorical columns, use StringIndexer and OneHotEncoder to create pipeline stages
4. Convert the label column using StringIndexer. Add it to the pipeline stage
5. Concatenate the categorical and numerical columns using VectorAssembler
6. Apply StandardScaler as a stage to the pipeline
7. Transform the dataset using the pipeline

Handling imbalance:

8. Note that the dataset is imbalanced. Use your favorite strategy to balance the dataset. You may convert to Pandas Dataframe to avail the existing packages.

Build classifiers and evaluate

9. Split it into train and test split
10. Implement the following three classifiers on the dataset: DecisionTreeClassifier, RandomForestClassifier and a classifier of your own choice. Adjust the hyperparameters to maximize performance.
11. Evaluate the classifiers on the test set using the metrics: F1, Accuracy and areaUnderROC (For F1 and Accuracy, you may use MulticlassClassificationEvaluator)

When the Python file is run: it should print the following

1. Stages in the pipeline (don't worry that the names will be cryptic)
2. Normalized counts of labels in both train and test split
3. Classifier-name, F1, Accuracy and areaUnderROC (for all three classifiers)

Important Pointers

1. You are expected to submit *.py file instead of Jupyter notebook.
2. You should submit a report describing the data preprocessing steps, feature selection method (if you implemented), imbalance handling strategy (if you implemented), screenshots showing the results obtained after running your Python file. Additionally, you can mention any important observations made during the assignment.
3. Here is the allocation of points per task

10% for a clean coding style

60% for the correctness of the implementation

10% for readable comments

20% for report