**Hi Everyone! This is team Risk Macaw.**

Our problem statement is :

There is a lot of focus on the regression testing efforts to be rightly spend on the critical areas of Ops Risk product development in GRC/MRM/Losses due to pace in development with parallel releases happening in Containerization and non-Containerization areas. Approximately 3 to 4 parallel releases and branches reach the production box in ECS and non ECS areas every month thereby making it extremely critical for the testers to focus on the hotspot modules during testing. As of today the testers are unable to identify the critical hotspots to focus on the right areas of testing. Similarly developers takes a lot of time to analyze the production defects and fix due to complexity in the code.​

1. To identify hotspots within the application which requires high degree of attention during regression testing.​
2. To identify defect patterns to give list of defects & solutions already occurred in past similar to the one in question for quick turnaround for developers.

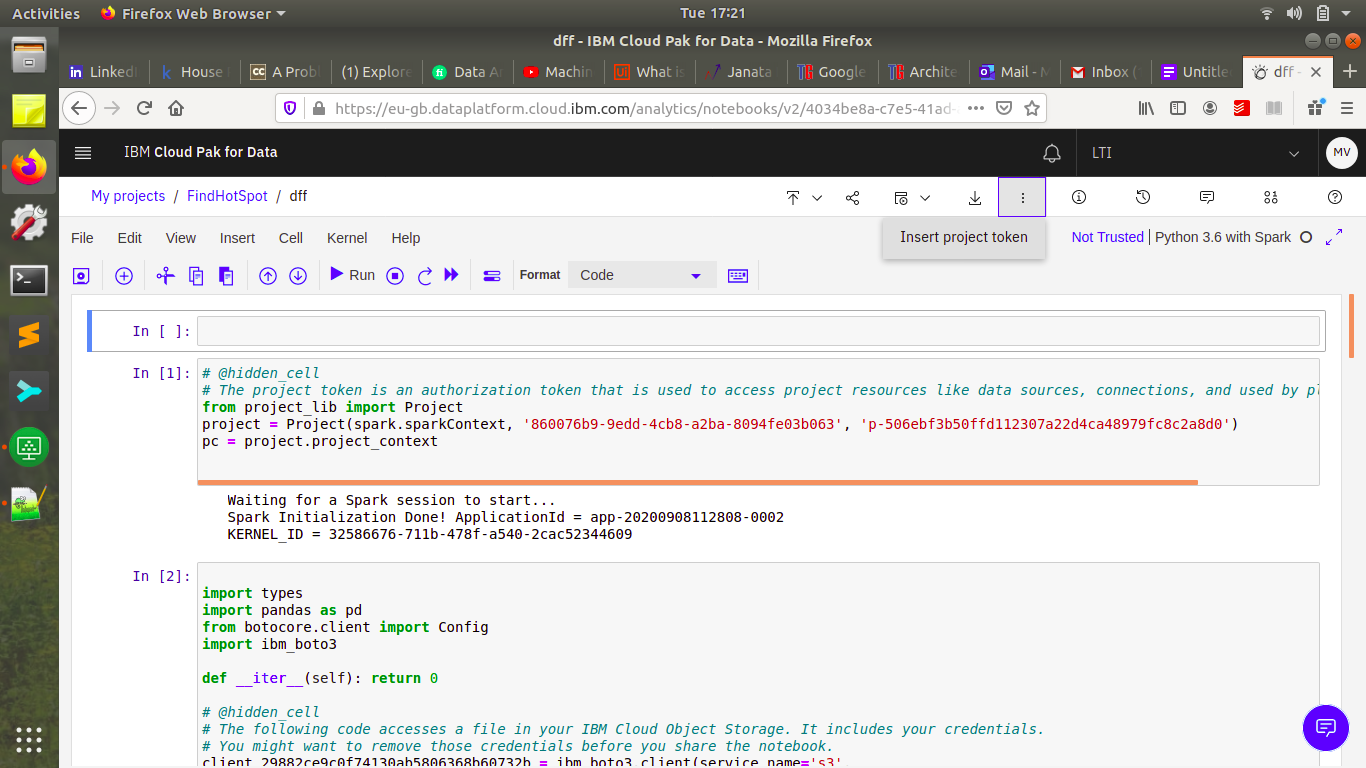
For these respective parts, we have 2 individual projects :

1. FindHotSpot

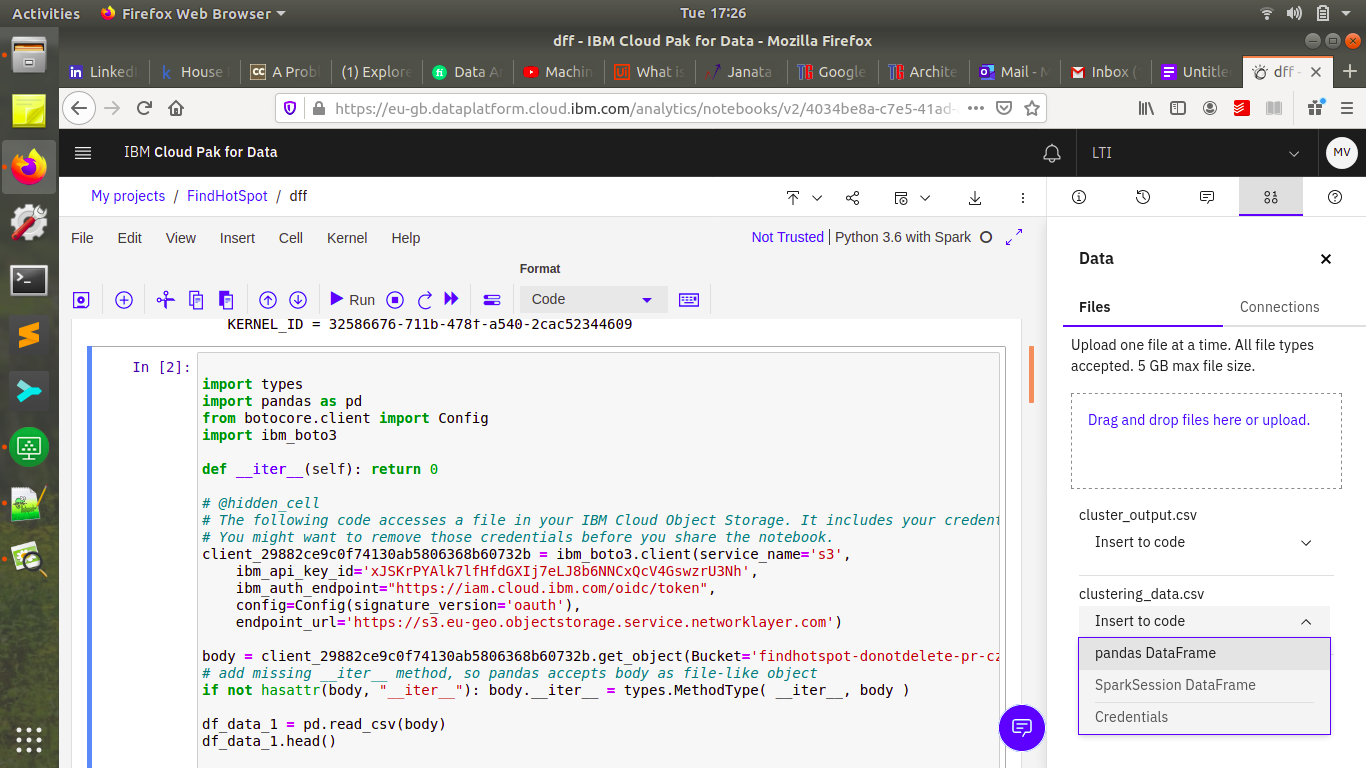
2. TrainClassifier

PART A

1. Under My Projects tab , select 'FindHotSpot'.
2. Once into the 'FindHotSpot'Project' , navigate to 'Assets Tab'. Under Assets Tab , we will find data assets, notebooks and dashboards.
3. In data assets , all the data files are mentioned.
4. In notebooks, we have a 'dff' file, which we are using for the prediction.
5. To run the project , open the ‘dff’ notebook in edit mode.
6. You will see a Jupyter notebook. Now add a new cell on top and now select ‘Insert new Token’ as shown in the figure.



1. Now add a new cell in the third line and navigate to side panel and select data. In data select the input file ‘clustering\_data.csv’ and ‘insert to code as pandas dataframe’. See the image for reference



1. Now after this, navigate to kernel tab and select ‘Restart and Run All’. This will execute the notebook. Now after this navigate to ‘Assets Tab’
2. Under 'Assets Tab' , now we can see two csv files, one is the output file ‘cluster\_output.csv’ .
3. To check the output, scroll down, we can find the dashboard, select the dashboard to visualize the output.

PART B

1. Under My Projects tab , select 'TrainClassifier'.
2. Once into the 'TrainClassifier'Project' , navigate to 'Assets Tab'. Under Assets Tab , we will find data assets and notebooks.
3. In data assets , all the data files are mentioned. 'test\_data.csv' is the input file.
4. In notebooks, we have 'TrainClassifier' file and 'PredictDefectResolution', which we are using for the prediction.
5. Select ‘TrainClassifier’ notebook and follow Steps 6 and 7 from PART A. Here the input file is ‘test\_data.csv’ and after this, execute the notebook.
6. Under 'Assets Tab' , now we can see two csv files, and pickle files.
7. Now, select notebook ‘PredictDefectResolution’ and follow steps 6 and 7. Here the input files are : category\_id.csv , defect\_summary.csv , svcclassifier.pickle and tfidf.pickle.
8. After this, execute this notebook.
9. To check the output, select the output file : 'predicted\_out.csv'. We can see all the resolutions for the problem.