Multilabel classification with boosting methods

CS777 Big Data

Student: Chen Wu

Professor: Kia Teymourian

Data set

* A Wikipedia page dataset, including 18347 general wiki pages, 377 court case pages and 18911 advertisement pages.
* I union those data to a file, and then random split into training and testing set as a ratio of 0.8 to 0.2. Label distribution was preserved.
* Three labels: 0 denotes general file, 1 denotes advertisement, 2 denotes court file.
* Attributes are: TF-IDF array of top 20000 words.
* Source for course cases file is from CS777 homework, I the test set of the dataset
* Source for advertisement file: <https://www.kaggle.com/urbanbricks/wikipedia-promotional-articles?select=promotional.csv>
* The label distribution is imbalanced. Court case pages portion is much smaller than the other general pages and promotion files.

Research Question

* 1. How a binary model classifies multiple labels?
* 2. How word stemming and stop word removal improve performance of text classification?
* 3. Could mini batch gradient descent has the same performance as full batch gradient descent?

Learning Models

MLE:

Multilabel classification with hierarchical classification or one against others both.

Two python files, Mini batch gradient descent and full batch gradient descent.

Batch size: 100

Word stemming and stop words removal

SVM:

Predict multiple labels with mini batch gradient descent.

Batch size: 100

Word stemming and stop words removal

Loss and gradient divided by batch size, not the count of documents

Logistic regression:

Using logistic regression model from Spark Mllib library to compare the performance of my classification models.

Random forest:

Using random forest model from Spark Mllib library to compare performance with my classification models.

Linear regression:

Used to test mini batch gradient descent to compare full batch gradient descent.

Expectation

Assume my MLE and SVM model could predict multiple labels with mini batch gradient descent.

As the best scenario, my MLE and SVM models could predict three different labels with mini batch gradient to reduce computation.

Assess model

Evaluate models by testing dataset.

Measurement: F score and accuracy to measure the correctness of the model.

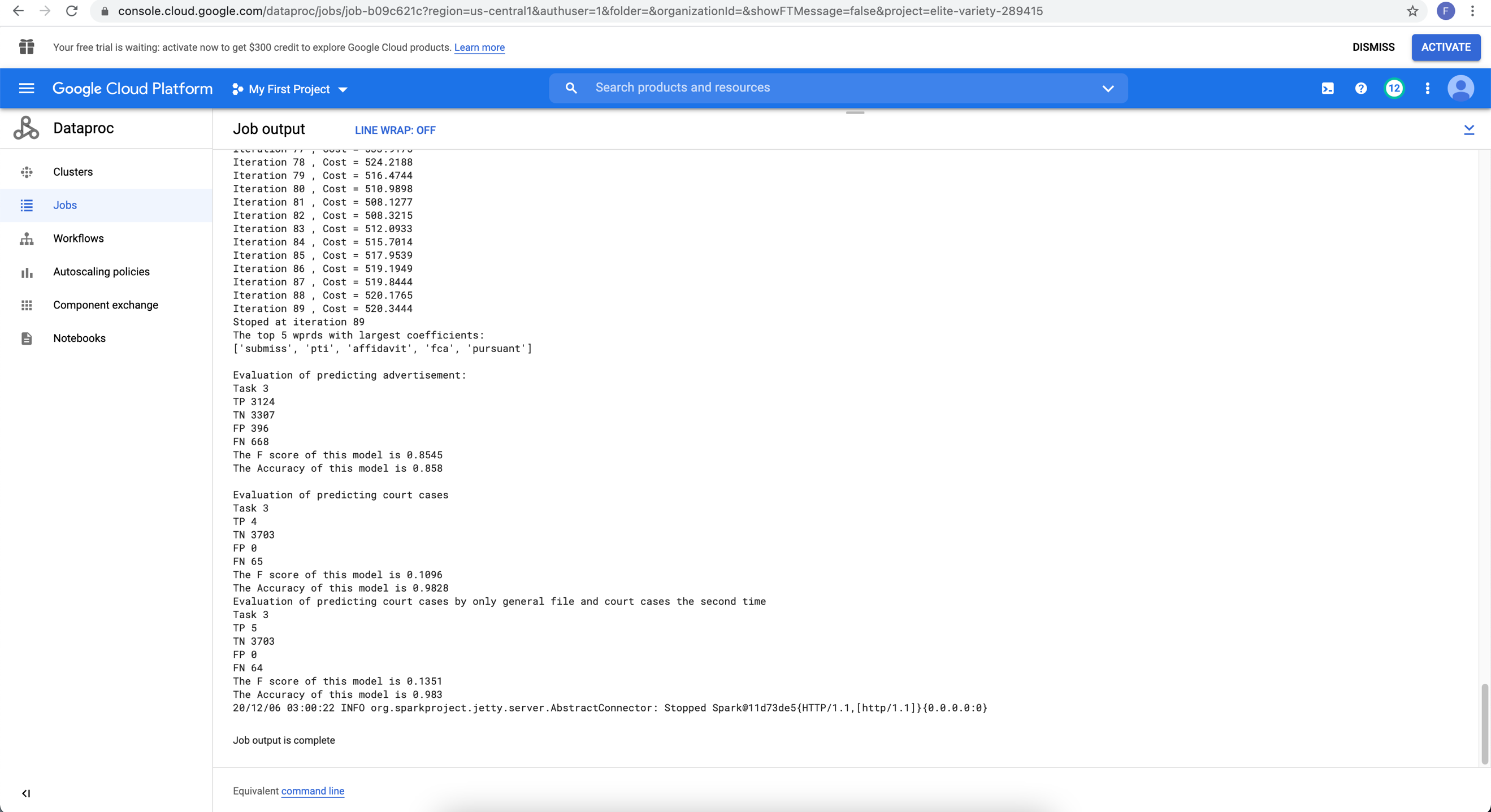
Expectation for measurements: The F score for predicting multiple labels should be above 0.8 and accuracy above 90%

Model Evaluation

MLE:

Full batch

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| classification method | F score advertisement | Accuracy advertisement | F score court | Accuracy court |
| Hierarchical | 0.85 | 85.8% | 0.11 | 98.28% |
| One against other | 0.85 | 85.8% | 0.13 | 98.3% |



Average F score and accuracy by adding both TP, TN, FP, FN up:

Hierarchical:

TP 3128

TN 7010

FP 396

FN 733

The F score of this model is 0.8471

The Accuracy of this model is 0.8998

One against other:

TP 3129

TN 7010

FP 396

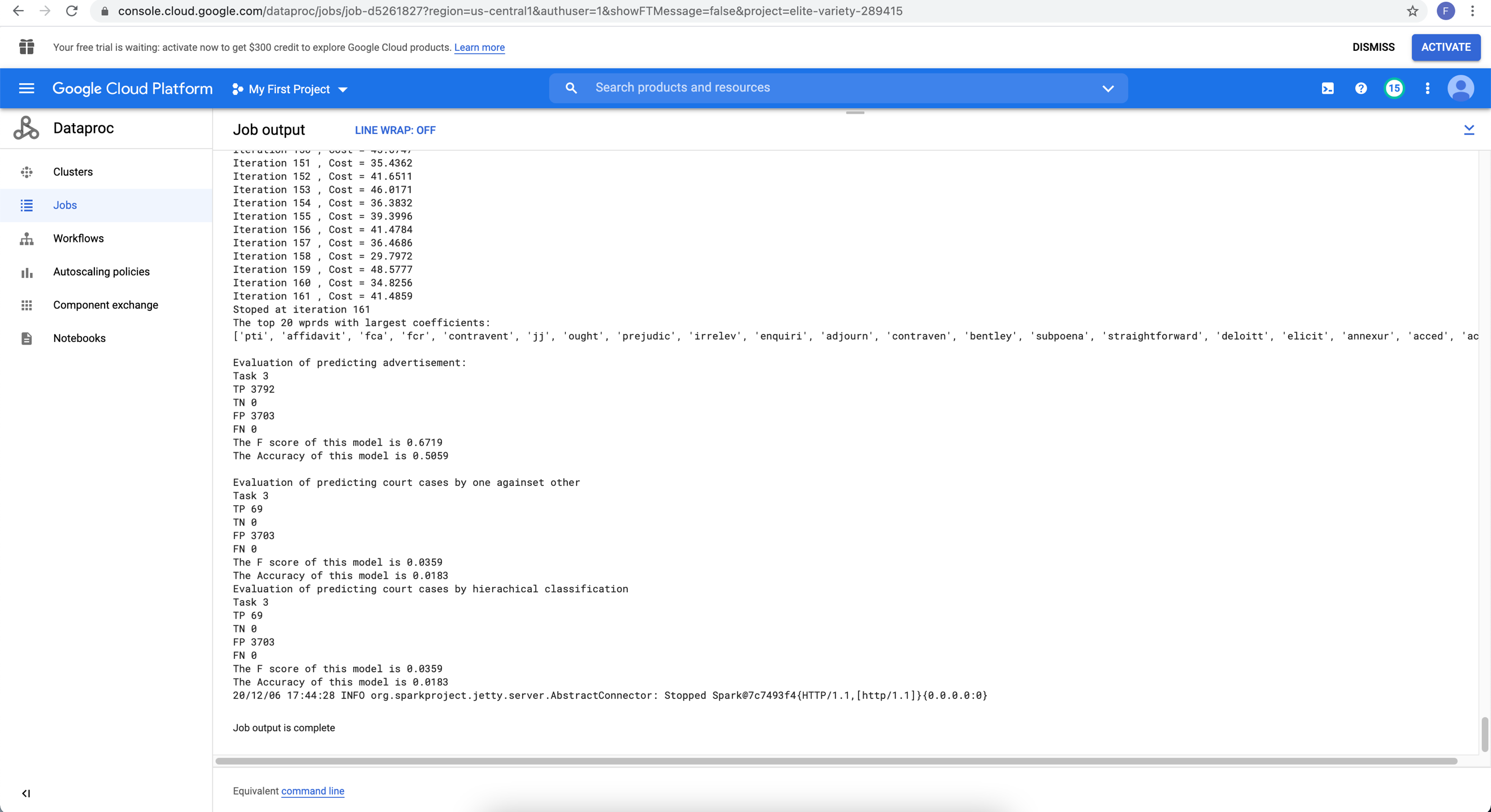
FN 732

The F score of this model is 0.8473

The Accuracy of this model is 0.8999

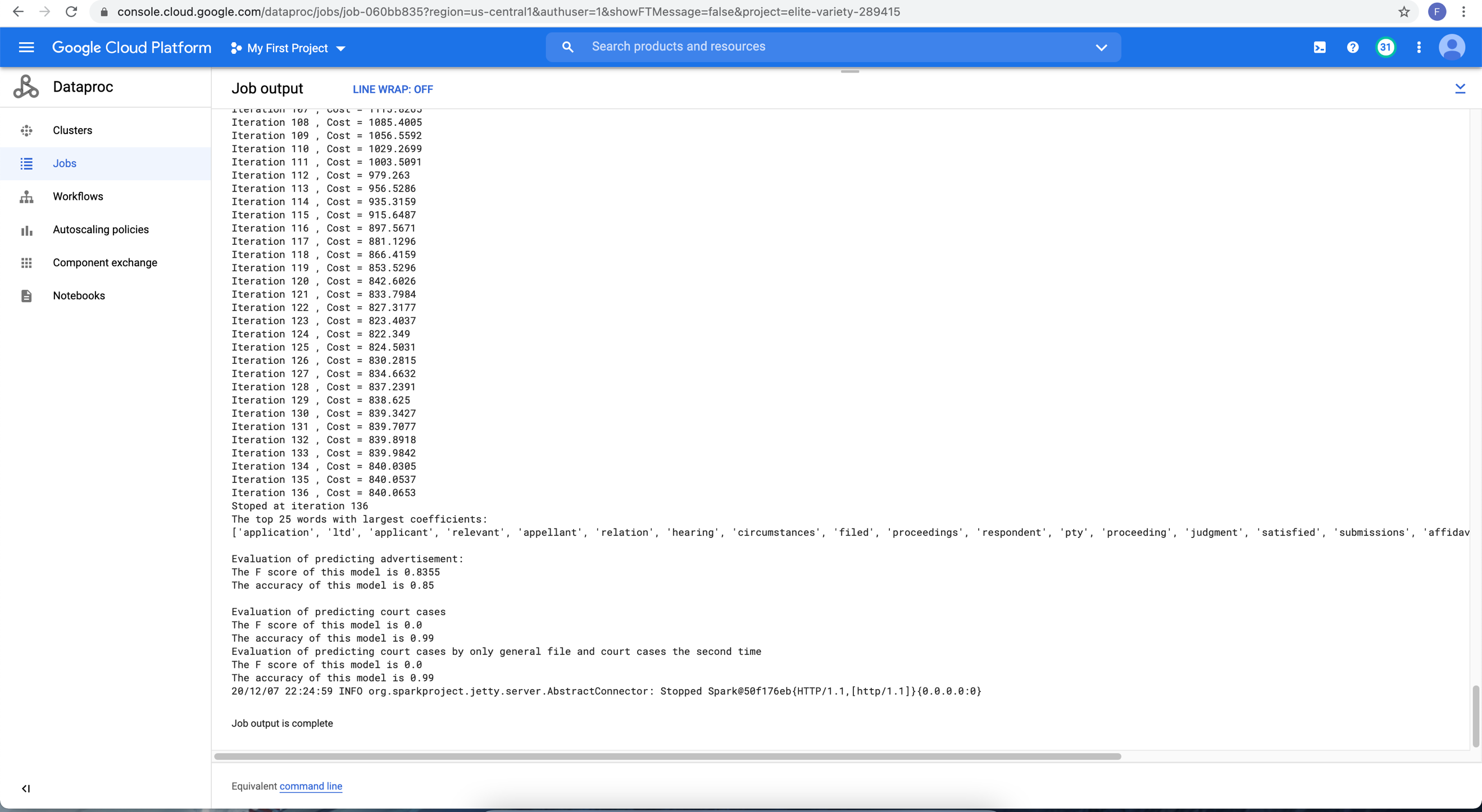
Mini batch

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| classification method | F score advertisement | Accuracy advertisement | F score court | Accuracy court |
| Hierarchical | 0.67 | 50.59% | 0.04 | 1.83% |
| One against other | 0.67 | 50.59% | 0.04 | 1.83% |



We can see an almost same result between hierarchical and one against other methods. Mini batch gradient descent doesn’t work for MLE.

MLE without stop words removal and stemming

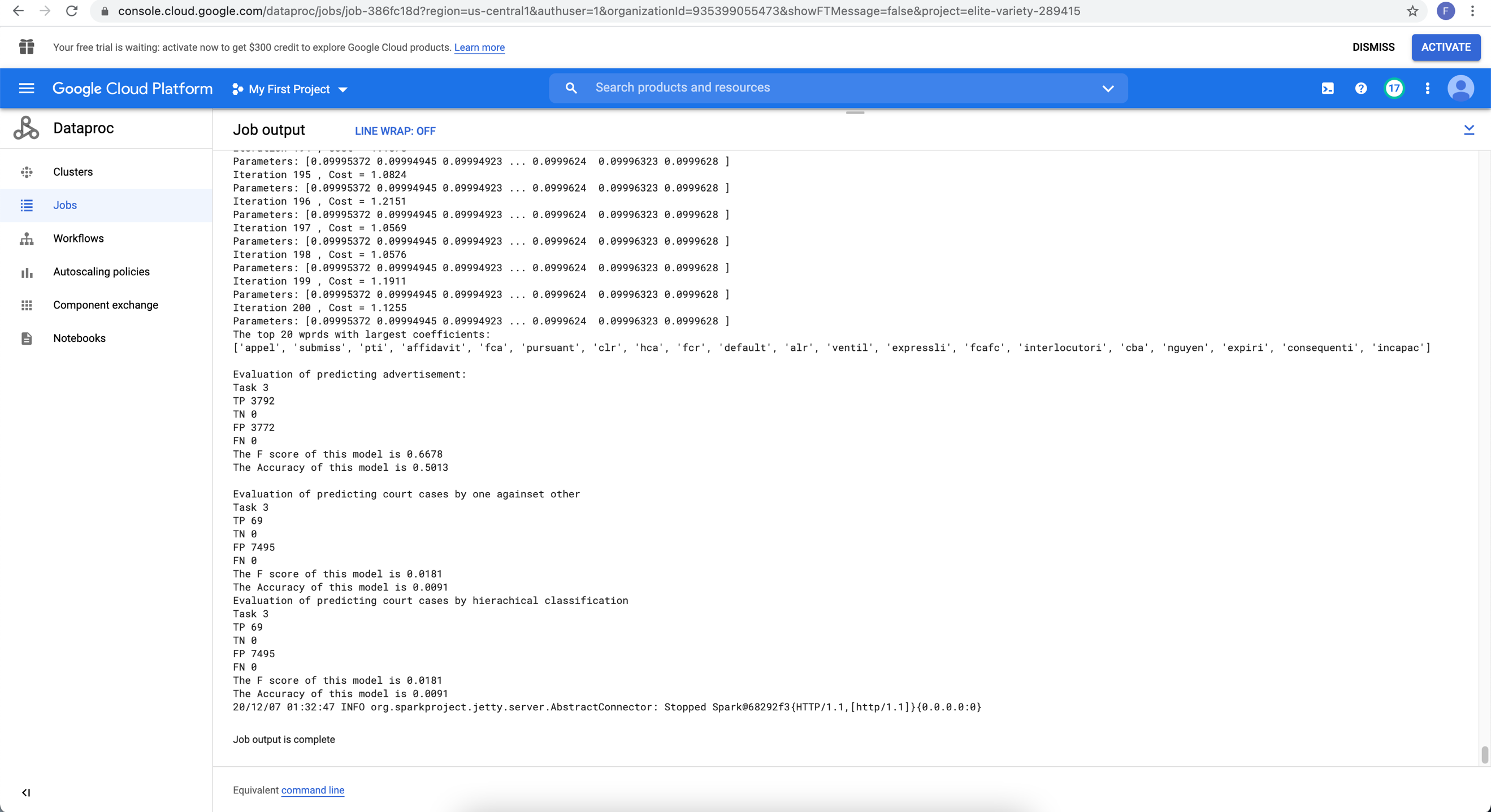


It doesn’t predict court cases without text preprocessing.

SVM:

Mini batch gradient descent for multilabel prediction:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| classification method | F score advertisement | Accuracy advertisement | F score court | Accuracy court |
| SVM | 0.67 | 50.13% | 0.02 | 0.1% |



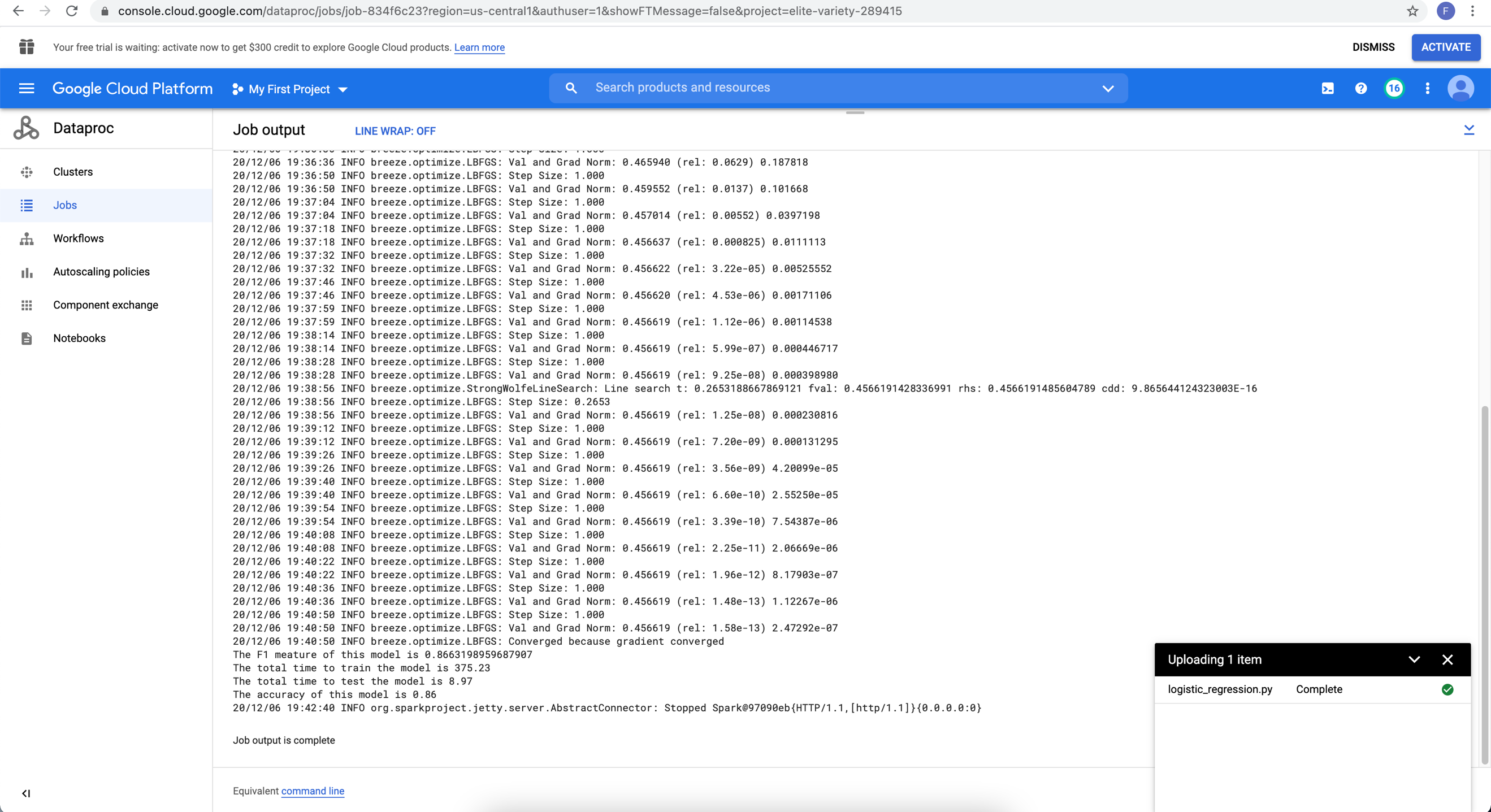
The SVM model has been failed to predict multiple labels with mini batch gradient. It’s not able to predict any label. The F score and accuracy for predicting advertisement don’t make sense because the label ratio of promotion is 50%.

Logistic regression by Spark library:

Full batch with 3 labels

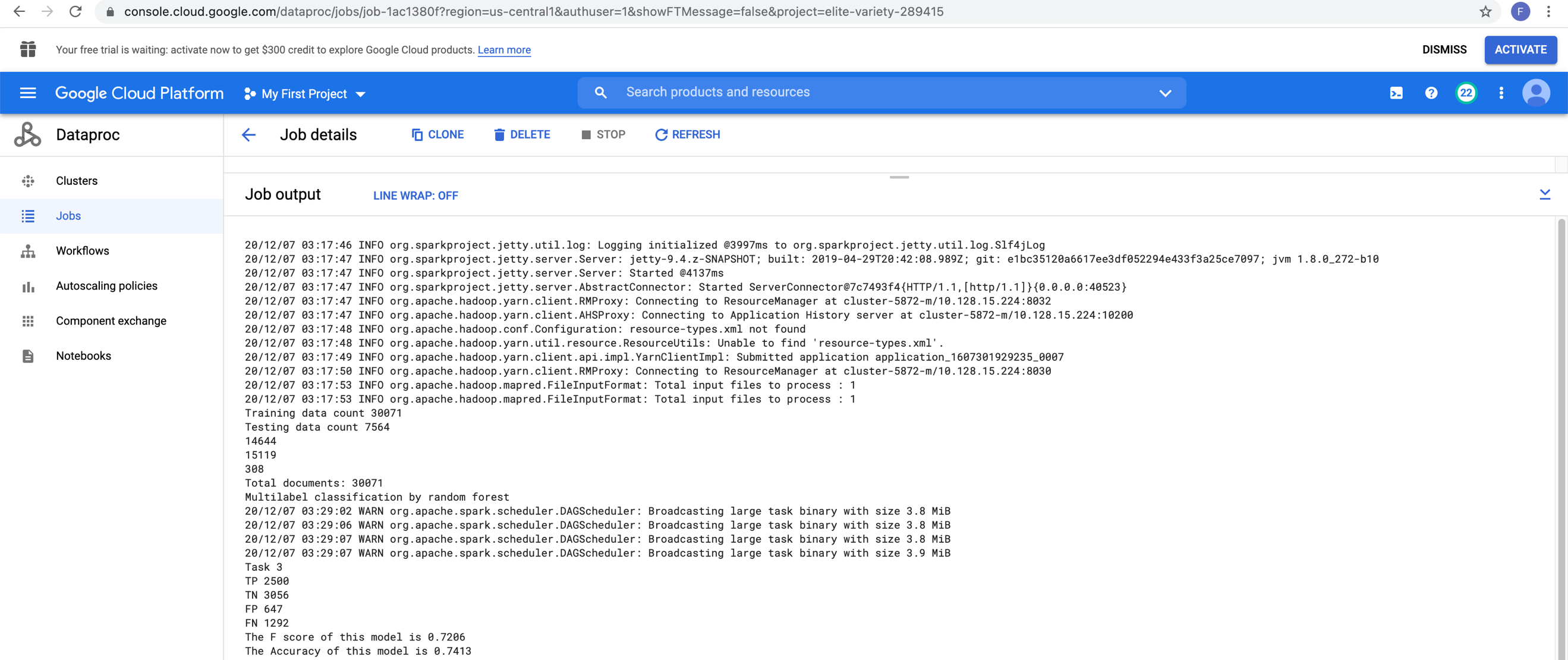
Spark logistic regression model train data and get parameters matrix using one against other approach. It does better than my models by an average F score for hierarchical and one against other.

F score 0.86, accuracy 86%.



Compare to algorithms which is naturally not binary, such as random forest:

We can see an overall F score 0.72 and 74% accuracy. Is not better than logistic regression.



Unlike binary algorithms, some algorithms naturally support multiple label classification, such as tree.

Random forest: F score 0.72 , accuracy 74%.

Logistic regression: F score 0.86, accuracy 86%.

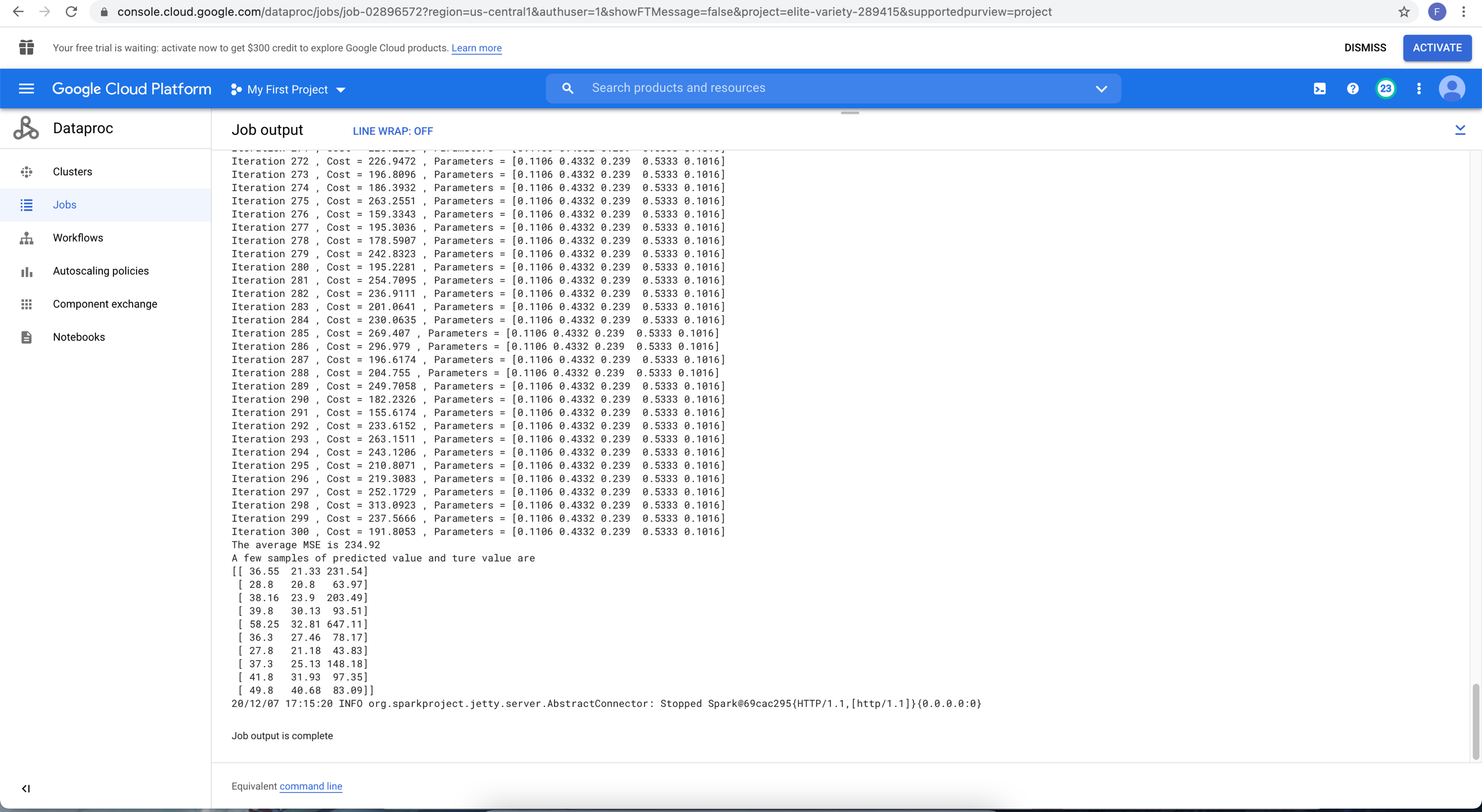
MLE: F score 0.8471, accuracy 90%. the best!

Random forest is not better than logistic regression. It’s not better than my models and Spark logistic regression library.

Briefing:

Seems mini batch gradient doesn’t work or my approach is wrong. Let’s test if mini batch gradient descent works for linear models.

Linear regression with mini batch gradient descent for predicting total amount paid to taxi:



We can see the squared error is too big and predicted total amount is not close to the true value. Let’s compare full batch linear regression:



We can see a much smaller average error compared to mini batch gradient. Full batch gradient descent works much better than mini batch.

Why mini batch does not work well?

* Loss varies by every batch. Batch loss is not in a small range, it can be much smaller than last time or much bigger. So, it doesn’t follow a pattern but reply on variance and label distribution.
* Label distribution is imbalanced. It escalates the variation of loss for each iteration. If the loss is 10 this time, last time might be 0 or 100. Unlike full batch gradient descent has a maximum loss.
* We have to adjust parameters for the batch size. It tried replacing total count by batch and scale up the dot product by batch size, nothing worked.

What I learned for this project

* 1. Mini batch gradient descent does not performance as good as using full training set data. Or there is a math equation to control loss but I didn’t find it. Mini batch gradient works for unsupervised learning better than supervised learning.
* 2. Hierarchical multilabel classification and one against other approach showed no big different on my model. The reason could be I used three distinct documents types, they are horizontally separable.
* 3. Mathematical thinking is more logistic, to prove a theory may requirement many opposite evidence.

Conclusion

* 1. Binary algorithms could learn a multilabel dataset by making a parameter matrix. Which could be hierarchical or one against other. Model can detect labels of test dataset and predict labels individually.
* 2. Text preprocessing by stemming and stop words removal could increase the accuracy compared to using all words. (0 for court without preprocessing)
* 3. Mini batch gradient descent does not have a performance as good as full batch. It has a worse performance than using all data.