I have successfully implemented code that generates random variables following two distinct Beta distributions, specifically Beta(0.01, 1) and Beta(0.01, 2). Among the work for image and text, I choose text. However, due to the constraints in applying the SpanBERT machine learning model for my task, I decided to take a different route. I selected a custom natural language processing approach to fulfill the task of discerning relationships between employees within my dataset.

For Beta(0.01, 1),

First run dataset.py by

python dataset.py,

文本

描述已自动生成

Then run main.py by

图表, 条形图, 直方图

描述已自动生成python main.py "SELECT \* FROM beta WHERE 1 ORACLE LIMIT 10000 USING proxy\_estimates [PRECISION] TARGET 0.95 WITH PROBABILITY 0.95"

or

图表, 条形图

描述已自动生成python main.py "SELECT \* FROM beta WHERE 1 ORACLE LIMIT 10000 USING proxy\_estimates [RECALL] TARGET 0.95 WITH PROBABILITY 0.95"

Same thing for Beta(0.01, 2),

First run dataset.py by

python dataset.py,

文本

描述已自动生成

Then run main.py by

图表, 条形图, 直方图

描述已自动生成python main.py "SELECT \* FROM beta WHERE 1 ORACLE LIMIT 1000 USING proxy\_estimates [PRECISION] TARGET 0.95 WITH PROBABILITY 0.95"

or

图表, 条形图

描述已自动生成python main.py "SELECT \* FROM beta WHERE 1 ORACLE LIMIT 1000 USING proxy\_estimates [RECALL] TARGET 0.95 WITH PROBABILITY 0.95"

Beta(0.01, 1) True values Probabilities 0.5% A(x) = Beta(0.01, 1) and O(x) = Bernoulli(A(x))  
Beta(0.01, 2) True values Probabilities 1% We use the same procedure as directly above but with Beta(0.01, 2)

**We can clearly see that the change of β value can improve the precision of the algorithm. For failure probability δ, if the precision target is 95%, both of my Beta1 and Beta2 experiment results give an error rate of 1/40, with 2.5% being within the confidence intervals.**

For text,

I assumed that A and B are representative of employees who may possess mutual feelings of either liking or disliking each other. These relationships are expressed as "A verb B", where A and B are randomly selected, distinct employees, and the verb is chosen from an extensive set indicating affinity or antipathy. Affectionate relations are conveyed using terms such as "like" or "are good friends", earning the "oracle\_labels" column a value of 1. Conversely, if the verb implies discord, such as "hate", "dislike", or "feel tired of", the "oracle\_labels" column receives a value of 0. Subsequently, a machine learning model is employed to deduce from the sentences themselves whether A likes B or not. The complexity of the sentences is further enhanced by the inclusion of adverbials or clauses, posing an additional challenge to the machine learning model.

First also run dataset.py by

python dataset.py,

**文本

描述已自动生成**

图表, 条形图, 直方图

描述已自动生成python main.py "SELECT \* FROM text WHERE 1 ORACLE LIMIT 1000 USING proxy\_estimates [PRECISION] TARGET 0.95 WITH PROBABILITY 0.95"