READ ME

=====================================================================================

**Dataset:**

Dataset represents a series of students working on a multi-KC system where every step is an attempt to a question and every question has specific skills(KC) relevant to it. Some questions have single skill and some have multiple skills associated with it.

Student: Anonymous student ID

StepID: Numeric id for each unique step.

Correct: Binary indicator of whether the step was correct 1 = Yes 0 = No.

KC 1 - KC 27: Per-KC relevance columns. 1 if the Step is relevant to the KC 0 if not.

**Code Structure:**

There are two files of code, one contains the function definitions, and the other is the Main Code which should be run.

There is an \_\_init\_\_ .py file which works as an initializer

In this code, I have implemented the Bayesian Knowledge Tracing model to calculate probability that a student already knows a skill/knowledge component(KC) before solving the question to which that skill is relevant.

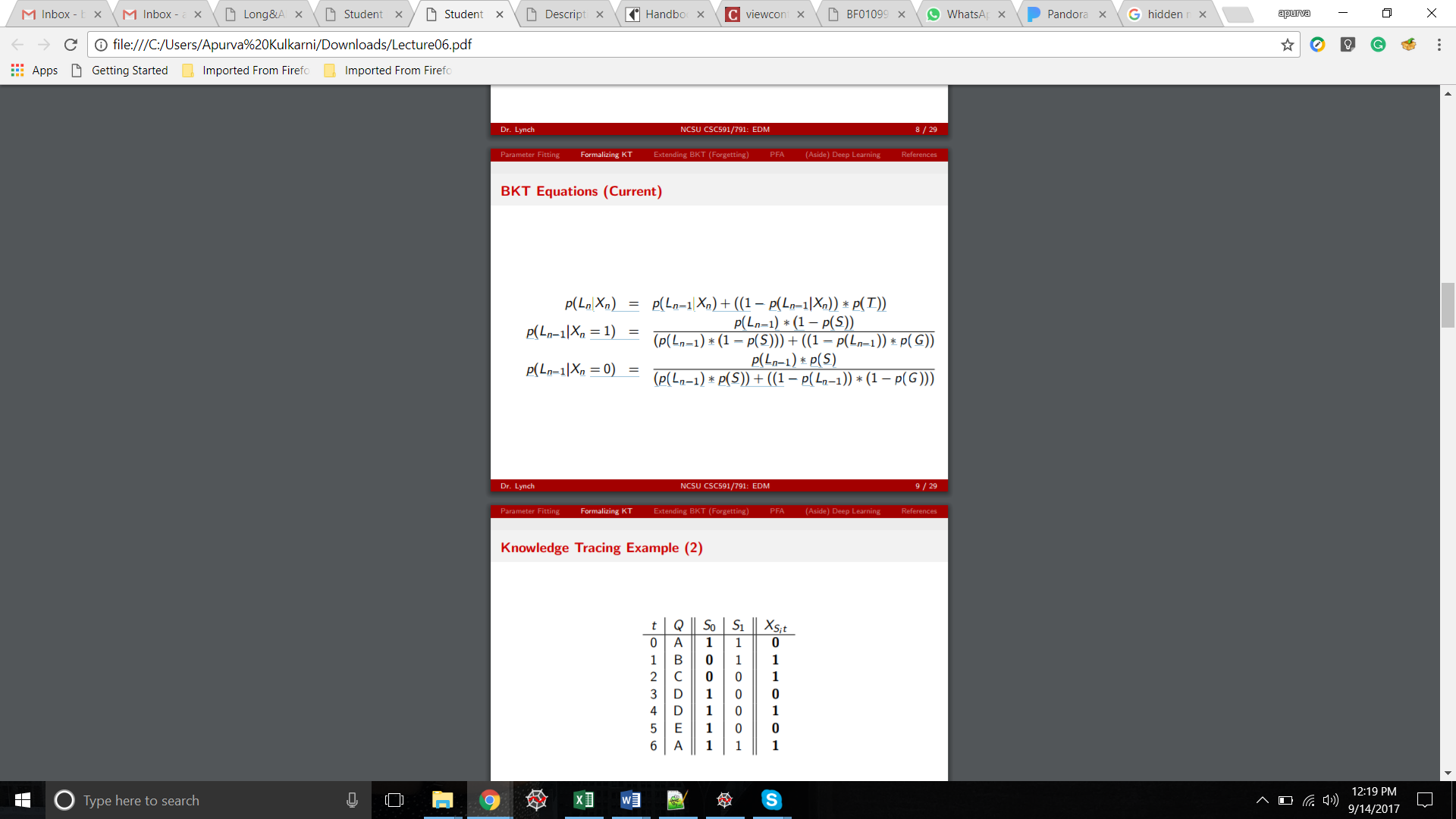
I have also calculated the probability that student will perform a correct action on the question by using the correct performance equation

The above equation only works for a single KC Step(ie when only one skill is relevant to one question) To calculate the probability that the student will succeed on a multi-KC step, a variation to the equation has also to be defined.

**Implementation of BKT Model:**

1. A function is defined, implementing the BKT Model

Equation used:



Where,

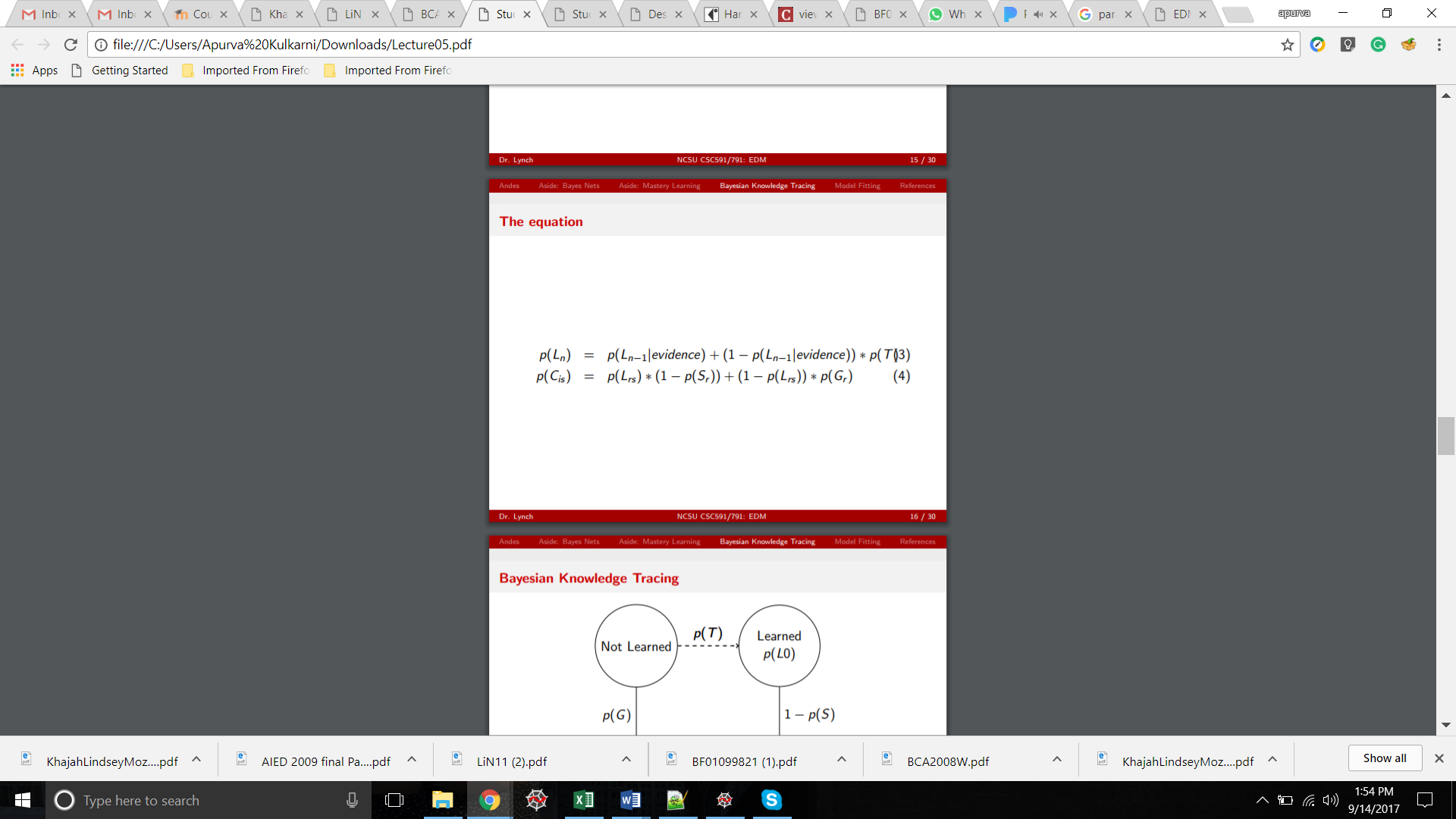
* P(Ln-1) Probability the skill is already known before this step in problem solving.
* p(L0) Probability the skill is already known before the first opportunity to use the skill in problem solving.
* p(T) Probability the skill will be learned at each opportunity to use the skill.
* p(G) Probability the student will guess correctly if the skill is not known.
* p(S) Probability the student will slip (make a mistake) if the skill is known.
* Xn Response to the question 1:correct, 0:incorrect

1. This model is trained by calling the above BKT function and passing the following KCs as parameter for each of the model:

KC\_1, KC\_27, KC\_24, KC\_14, KC\_22, KC\_20, KC\_21

This returns the probability that the appropriate KC is in the learned state for the student

1. The probability that student will perform a correct action is calculated using the correct performance equation:



1. The above correct performance equation only works for a single KC Step

To calculate the probability that the student will succeed on a multi-KC step, a variation to the equation has to be defined.

**Implementation of Multi-Expert Model:**

1. A model has to be designed that calculates the probability that a student succeeds on a Step with multiple KCs relevant to it.
2. As per the Specific Multiplication Rule in Probability,

P(A and B) = P(A) \* P(B)

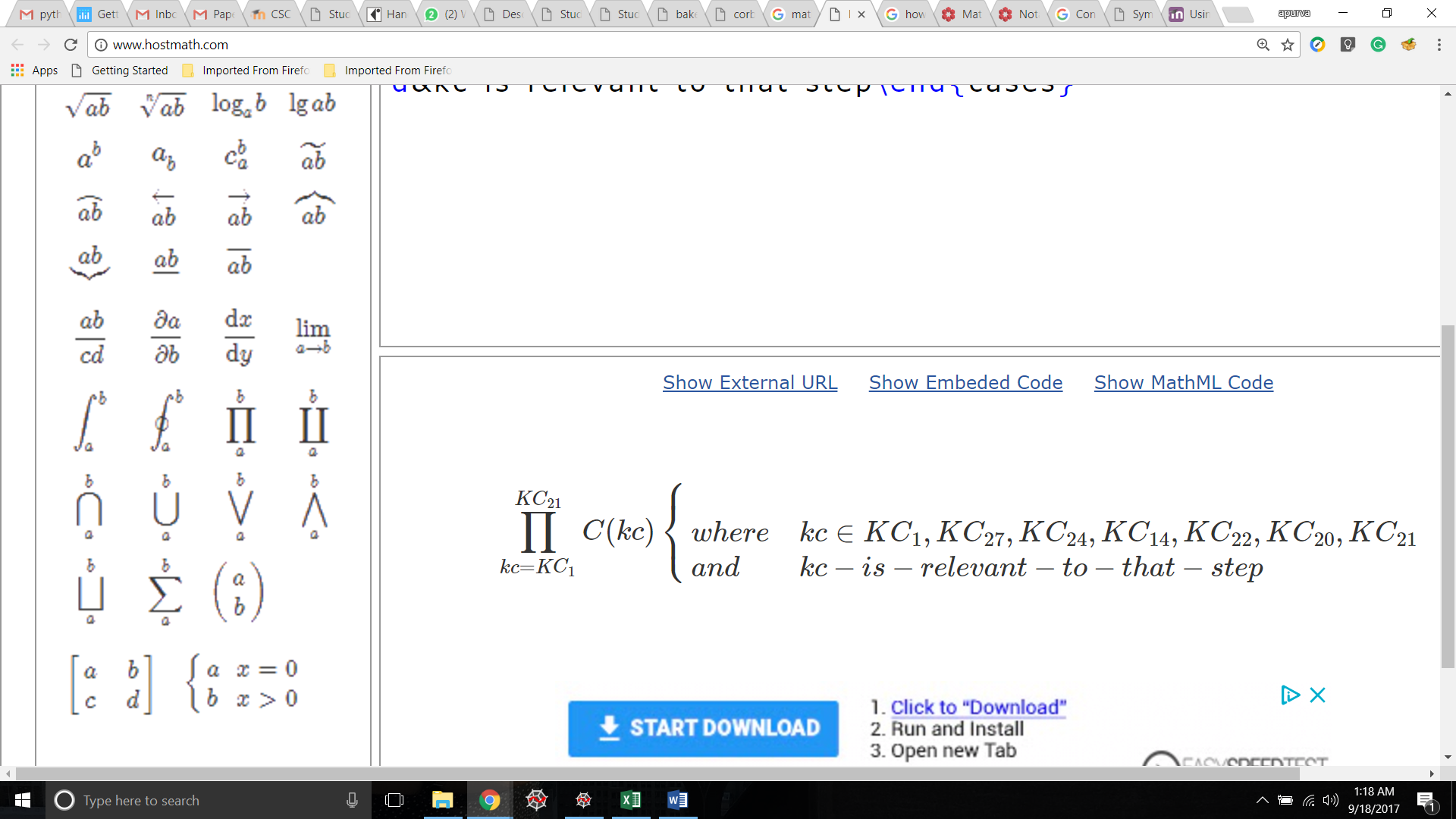
A and B have to be independent events.

1. Relevance to one KC doesn’t depend on other KCs, hence the above rule can be applied in MultiKC step as follows:

P (C\_KC1 and C\_KC27 and so on….) = C\_KC1 \* C\_KC27 \* (product of all relevant KCs)

That is: the product of individual correctness of all the relevant KC

1. The above rule can be represented in following equation:



**Citations for parameter settings:**

* As specified in “More Accurate Student Modeling Through Contextual Estimation of Slip and Guess Probabilities in Bayesian Knowledge Tracing Ryan S.J.d. Albert T. Corbett Vincent Aleven”, the guess parameter is bounded to be between 0 and 0.3, and the slip parameter is bounded to be between 0 and 0.1
* And as specified in “Learning Bayesian Knowledge Tracing Parameters with a Knowledge Heuristic and Empirical Probabilities William J. Hawkins, Neil T. Heffernan, Ryan Baker”, the value of initial learning is assumed to be 0.5.
* Hence, I have assumed the following parameter settings:
  + - P(L0) = 0.5
    - P(T) = 0.7
    - P(G) = 0.3
    - P(S) = 0.1