CS777 – Week 2 Homework Submission Template

**!!!! PLEASE RENAME THIS DOCUMENT WITH YOUR NAME AND LASTNAME !!!!**

**Task 1 – Generate the Top 20K dictionary and Create the TF-IDF Array (4 Points)**

Get the top 20,000 words in a local array and sort them based on the frequency of words. In the end, produce an RDD that includes the docID as key and a NumPy array for the position of each word in the top 20K dictionary:

(docID, [dictionaryPos1, dictionaryPos2, dictionaryPos3...])

* In your code print out print allDocsAsNumpyArrays.take(3).

|  |
| --- |
| *[('45461112', array([0.08214804, 0.02815675, 0.01886792, ..., 0. , 0. ,*  *0. ])), ('35790646', array([0.07594937, 0. , 0.02531646, ..., 0. , 0. ,*  *0. ])), ('153150', array([0.09082969, 0.03231441, 0.02358079, ..., 0. , 0. ,*  *0. ]))]* |

* In your code print out print allDocsAsNumpyArraysTFidf.take(2):

|  |
| --- |
| *[('44056227', array([0.00850956, 0.01057571, 0.00474021, ..., 0. , 0. ,*  *0. ])), ('2293690', array([0.0079421 , 0.00531117, 0.00328539, ..., 0. , 0. ,*  *0. ]))]* |

**Task 2 – Implement the getPrediction function (8 Points)**

Print out the results for the following queries:

* print(getPrediction('Sport Basketball Volleyball Soccer', 10))

|  |
| --- |
| *Prediction for Sport Basketball Volleyball Soccer: [('All\_stub\_articles', 4), ('Disambiguation\_pages', 3), ('Disambiguation\_pages\_with\_short\_description', 3), ('All\_disambiguation\_pages', 3), ('All\_article\_disambiguation\_pages', 3), ('Articles\_with\_Turkish-language\_sources\_(tr)', 2), ('Articles\_containing\_Turkish-language\_text', 2), ('Living\_people', 2), ('Sportspeople\_from\_Newport\_Beach', 1), ('1971\_establishments\_in\_Turkey', 1)]* |

* print(getPrediction('What is the capital city of Australia?', 10))

|  |
| --- |
| *Prediction for What is the capital city of Australia?: [('Disambiguation\_pages\_with\_short\_description', 4), ('All\_disambiguation\_pages', 4), ('All\_article\_disambiguation\_pages', 4), ('Disambiguation\_pages', 3), ('All\_stub\_articles', 2), ('Regional\_capitals\_in\_Tanzania', 1), ('Unprintworthy\_redirects', 1), ("Articles\_with\_'species'\_microformats", 1), ('Commons\_category\_link\_from\_Wikidata', 1), ('Lists\_of\_capitals', 1)]* |

* print(getPrediction('How many goals Vancouver score last year?', 10))

|  |
| --- |
| *Prediction for How many goals Vancouver score last year?: [('Coordinates\_on\_Wikidata', 3), ('Webarchive\_template\_wayback\_links', 3), ('Disambiguation\_pages', 2), ('Disambiguation\_pages\_with\_short\_description', 2), ('All\_disambiguation\_pages', 2), ('Articles\_with\_short\_description', 2), ('All\_articles\_with\_unsourced\_statements', 2), ('All\_article\_disambiguation\_pages', 2), ('Canadian\_building\_and\_structure\_stubs', 1), ('Articles\_with\_unsourced\_statements\_from\_December\_2014', 1)]* |

**Task 3 – Using Dataframes (6 points)**

**Task 3.1**

Use Spark Dataframe to provide summary statistics (max, average, median, std) about the

number of Wikipedia categories that are used for Wikipedia pages. Print the results on the output

console, or store them on the cloud storage.

Hint: This question is all about Wikipedia categories.

|  |
| --- |
| *Max: 587, Average: 5.566655905819486, StdDev: 5.567163467038092, Median: 4* |

**Task 3.2**

Use Spark Dataframe to find the top 10 most used Wikipedia categories. Print the results on the output console, or store them on the cloud storage.

|  |
| --- |
|  |

**Task 4 – Removing Stop Words, do Stemming and redo task 2 (2 points)**

**Task 4.1 – Remove Stop Words (1 point)**

Describe if removing the English Stop words (most common words like ”a, the, is, are, i, you, ...”) would change the final kNN results here.

You do not need to implement this task, only discuss your expected outcome results.

|  |
| --- |
| *In my opinion, the results may change a lot. The noise will be reduced, and some context may loss as well. The top 20k words will change leading the different vectors and different KNN results* |

**Task 4.2 – Do English word stemming (1 point)**

We can stem the words [”game”,”gaming”,”gamed”,”games”] to their root word ”game”. Read more about stemming here <https://en.wikipedia.org/wiki/Stemming>

You do not need to implement this task, only discuss your expected outcome results.

|  |
| --- |
| *I believe the result will change but may not heavily. Stemming also reduce the noise and may lead a more general result which may different from original result.* |

**Spark History Output:**

**Task 1:**

|  |
| --- |
|  |

**A screenshot of a computer

Description automatically generated**

**Task 2:**

|  |
| --- |
| *I run task 1 and 2 together* |

**Task 3:**

|  |
| --- |
|  |