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).

|  |
| --- |
| *Top Words in Corpus: [('the', 26451056), ('of', 12507151), ('in', 10807932), ('and', 10758007), ('a', 7988396), ('to', 7681799), ('was', 4588136), ('is', 3746524), ('for', 3145951), ('as', 3066049)]* |

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

|  |
| --- |
| *Word Postions in our Feature Matrix. Last 20 words in 20k positions: [('sizeable', 19999), ('unlocked', 19998), ('dent', 19997), ('sadler', 19996), ('lockout', 19995), ('minerva', 19994), ('ub', 19993), ('ericsson', 19992), ('ami', 19991), ('erica', 19990), ('yarmouth', 19989), ('overt', 19988), ('csa', 19987), ('divert', 19986), ('melancholy', 19985), ('madurai', 19984), ('rida', 19983), ('corbin', 19982), ('estadio', 19981), ('ethnographic', 19980)]* |

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

Print out the results for the following queries:

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

|  |
| --- |
| *Output file:*  *('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)*  *Console output:*  *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))

|  |
| --- |
| *Output file:*  *('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)*  *('Lists\_of\_capitals', 1)*  *('Commons\_category\_link\_from\_Wikidata', 1)*  *Console Output:*  *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), ('Lists\_of\_capitals', 1), ('Commons\_category\_link\_from\_Wikidata', 1)]* |

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

|  |
| --- |
| *Output file:*  *('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)*  *('Articles\_with\_hCards', 1)*  *('Stanley\_Cup\_championship\_seasons', 1)*  *Console Output:*  *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), ('Articles\_with\_hCards', 1), ('Stanley\_Cup\_championship\_seasons', 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.

|  |
| --- |
| *Task 3.1: Summary Statistics*  *+-------+-----------------+*  *|summary| num\_categories|*  *+-------+-----------------+*  *| count| 25323285|*  *| mean|5.566655905819486|*  *| stddev|5.567163467038119|*  *| min| 1|*  *| max| 587|*  *+-------+-----------------+*  *Max: 587*  *Average: 5.566655905819486*  *Median: 4.0*  *StdDev: 5.567163467038041* |

**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.

|  |
| --- |
| *+--------------------+-------+*  *| category| count|*  *+--------------------+-------+*  *| "Noindexed\_pages"|2586023|*  *| "All\_stub\_articles"|2243344|*  *|"WikiProject\_Biog...|1650535|*  *|"Articles\_with\_sh...|1516225|*  *|"Redirects\_from\_m...|1489008|*  *|"Unprintworthy\_re...|1412217|*  *|"Coordinates\_on\_W...|1048235|*  *|"Biography\_articl...| 970853|*  *|"Stub-Class\_biogr...| 939454|*  *| "Living\_people"| 938708|*  *+--------------------+-------+* |

**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.

|  |
| --- |
| Expected Outcome:  Removing English stop words like "a," "the," "is," "are," "I," "you," etc., would likely improve the kNN results. Stop words are common in all kinds of texts and do not provide significant meaning when it comes to categorizing or classifying the text. Therefore, they can be considered as noise in the data.  Reasons:  Reduced Noise: Stop words do not contribute to the semantics of the text. Removing them reduces the noise in the data.  Improved Accuracy: By eliminating these common words, the kNN algorithm can focus on more meaningful words, which are more likely to be relevant for categorization.  Computational Efficiency: Removing stop words decreases the dimensionality of the data, making the kNN algorithm faster and more efficient.  Better Distance Metric: The cosine similarity or any other distance metric would be more meaningful as it would be computed based on relevant words.  However, the extent to which the results change would depend on the nature of the text data and what it is you're trying to classify or categorize. |

**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.

|  |
| --- |
| Expected Outcome:  Stemming words to their root form would also likely improve the kNN results but may not change them "heavily" unless the dataset has a lot of words that are variations of each other.  Reasons:  Vocabulary Consolidation: Stemming reduces inflected words to their root form. For example, "game," "gaming," "gamed," and "games" would all be reduced to "game." This can make the kNN algorithm more accurate as it treats different forms of the same word as one.  Improved Generalization: Stemming can make the model generalize better to new but similar texts.  Reduced Dimensionality: Like removing stop words, stemming also reduces the dimensionality of the feature space, making the algorithm more efficient.  Context Sensitivity: One downside could be that stemming might make some words lose their specific meanings, which could be important in certain contexts. For example, "running" and "ran" are stemmed to "run," but the former is present continuous while the latter is simple past. |

**Spark History Output:**

**Task 1:**

|  |
| --- |
| *out1and2spark* |

**Task 2:**

|  |
| --- |
| *out1and2spark* |

**Task 3:**

|  |
| --- |
| *out3spark* |