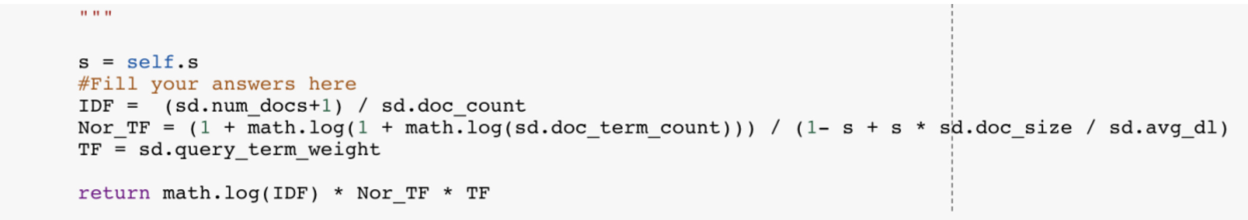
SI 650 Information Retrieval Assignment 2 part 1

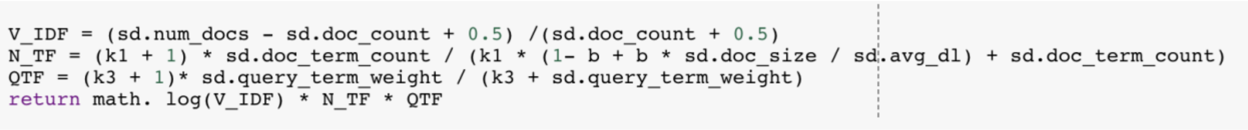
1. Built Retrieval Function

Pivoted function:



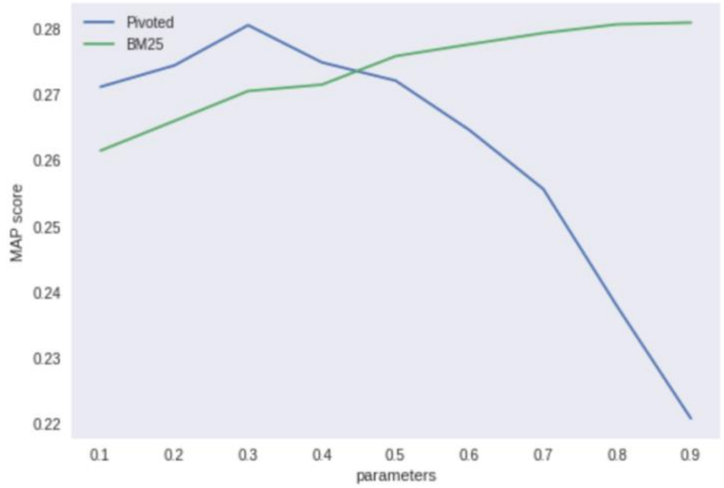
```
        """

        s = self.s
        #Fill your answers here
        IDF =  (sd.num_docs+1) / sd.doc_count
        Nor_TF = (1 + math.log(1 + math.log(sd.doc_term_count))) / (1- s + s * sd.doc_size / sd.avg_dl)
        TF = sd.query_term_weight

        return math.log(IDF) * Nor_TF * TF
```

BM25 function:

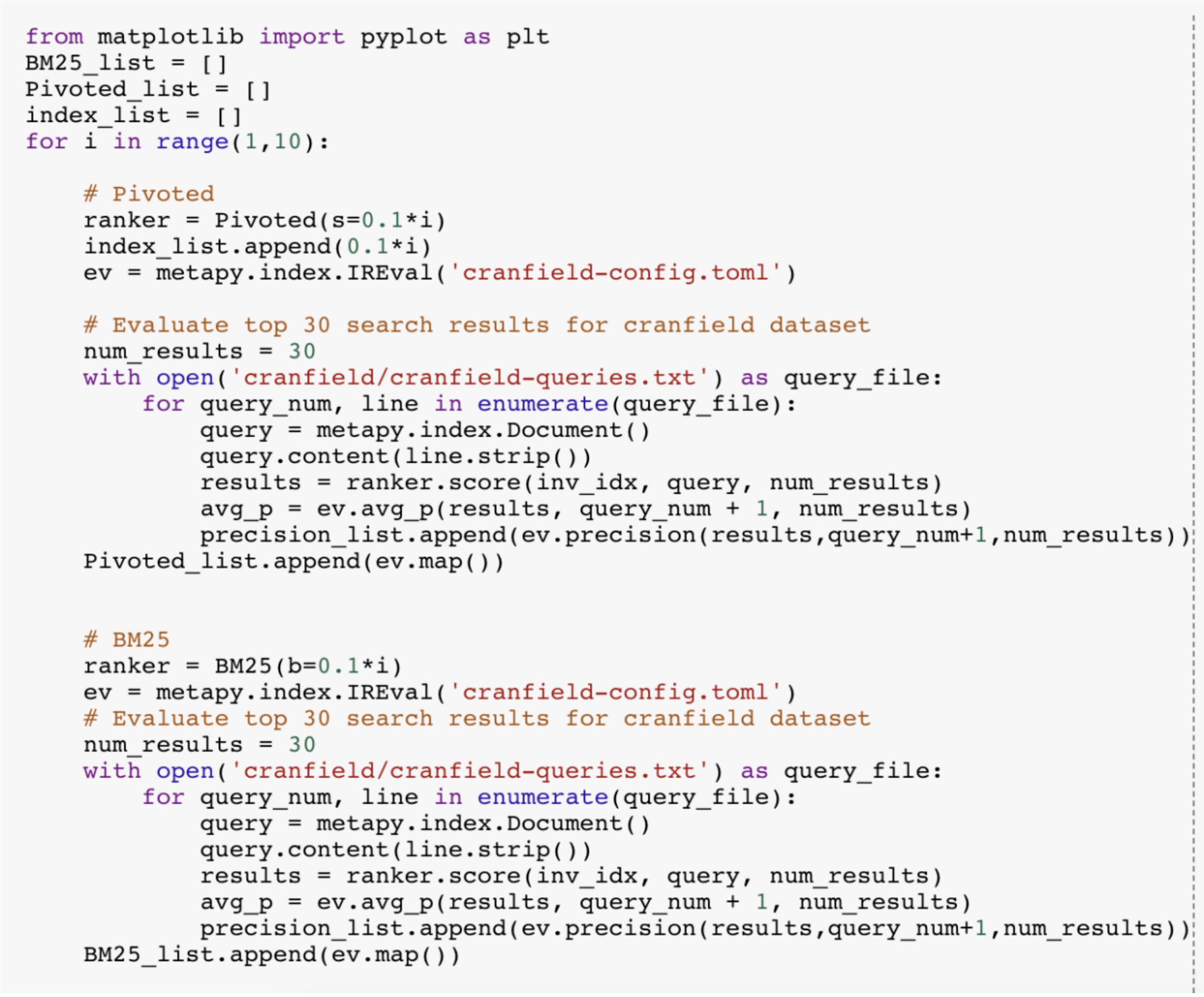


```
V_IDF = (sd.num_docs - sd.doc_count + 0.5) /(sd.doc_count + 0.5)
N_TF = (k1 + 1) * sd.doc_term_count / (k1 * (1- b + b * sd.doc_size / sd.avg_dl) + sd.doc_term_count)
QTF = (k3 + 1)* sd.query_term_weight / (k3 + sd.query_term_weight)
return math. log(V_IDF) * N_TF * QTF
```

2. Evaluate your Retrieval Function



```
Pivoted: [0.271180483845591, 0.2744279237181812, 0.2805417429677985,
0.27488304406886577, 0.27211184481732426, 0.2646063060185587,
0.2556696270439405, 0.23785328576891762, 0.22081696949293717]
The Best parameter is s = 0.3 value = 0.2805417429677985
BM25: [0.261491760735152, 0.265995430371294, 0.27054272369703847,
0.2715105751692196, 0.2758441537423579, 0.27763057856646,
0.2793448148044489, 0.28066853779697154, 0.28092929667443534]
The best parameter is b = 0.9 value = 0.28092929667443534
```



```
from matplotlib import pyplot as plt
BM25_list = []
Pivoted_list = []
index_list = []
for i in range(1,10):

    # Pivoted
    ranker = Pivoted(s=0.1*i)
    index_list.append(0.1*i)
    ev = metapy.index.IREval('cranfield-config.toml')

    # Evaluate top 30 search results for cranfield dataset
    num_results = 30
    with open('cranfield/cranfield-queries.txt') as query_file:
        for query_num, line in enumerate(query_file):
            query = metapy.index.Document()
            query.content(line.strip())
            results = ranker.score(inv_idx, query, num_results)
            avg_p = ev.avg_p(results, query_num + 1, num_results)
            precision_list.append(ev.precision(results,query_num+1,num_results))
    Pivoted_list.append(ev.map())


    # BM25
    ranker = BM25(b=0.1*i)
    ev = metapy.index.IREval('cranfield-config.toml')
    # Evaluate top 30 search results for cranfield dataset
    num_results = 30
    with open('cranfield/cranfield-queries.txt') as query_file:
        for query_num, line in enumerate(query_file):
            query = metapy.index.Document()
            query.content(line.strip())
            results = ranker.score(inv_idx, query, num_results)
            avg_p = ev.avg_p(results, query_num + 1, num_results)
            precision_list.append(ev.precision(results,query_num+1,num_results))
    BM25_list.append(ev.map())
```