```
{
 "cells": [
  {
   "cell_type": "markdown",
   "metadata": {},
   "source": [
    "# Assignment 6 - Streaming Text Analysis using Spark"
   ]
  },
  {
   "cell_type": "markdown",
   "metadata": {},
   "source": [
    "## Description:\n",
    "\n",
    "In this assignment, you will develop a Spark Streaming application to perform sentiment analysis. You will use a provided text file, where each line is considered a streaming input, to simulate real-time data processing.\n",
    "\n",
    "## Learning Outcome:\n",
    "\n",
    "By the end of this assignment, you will be able to:\n",
    "\n",
    "\n",
    "1. Set up a local Spark Streaming environment.\n",
    "1. Understand the basics of sentiment analysis.\n",
    "1. Implement a Spark Streaming application for real-time sentiment analysis.\n",
    "1. Analyze and interpret the results from streaming data."
   ]
  },
  {
   "cell_type": "markdown",
   "metadata": {},
   "source": [
    "## Important Notes:\n",
    "    \n",
    "- All programs are to be written using Python 3.\n",
    "- To get full marks, your code must be well-documented."
   ]
  },
  {
   "cell_type": "markdown",
   "metadata": {},
   "source": [
```

```
    "## Part 1: Sentiment Analysis\n",
    "### 0 - Folder Structure\n",
    "``` bash\n",
    "assignment6 (folder)\n",
    "    spark.py (file)\n",
    "    textfile_directory (folder)\n",
    "        textFile.txt (file)\n",
    "```\n",
    "\n",
    "### 1- Environment Setup\n",
    "Ensure you have a Spark environment set up locally. Refer to the
initial sections of the \"Lab4.ipynb\" for guidance on setting up your
environment.\n",
    "\n",
    "### 2- Data Source:\n",
    "A text file will be provided, containing sentences, each on a new
line. This file will simulate your streaming data source.\n",
    "\n",
    "### 3- Read from a text file as a source for Spark Streaming\n",
    "#### Locate Your Text File\n",
    "Ensure your text file (e.g., user_review.txt) is located in a
directory (textfile_directory) that your Spark application can access. Each
line in this file should represent a separate streaming entry.\n",
    " \n",
    "#### Create a StreamingContext\n",
    "First, you need to initialize a Spark StreamingContext. This is a key
entry point for all streaming functionalities.\n",
    "\n",
    "<br/>\n",
    "<br/>\n",
    "\n",
    "``` python\n",
    "# Spark Libraries\n",
    "from pyspark import SparkContext\n",
    "from pyspark.streaming import StreamingContext\n",
    "\n",
    "## Sentiment Analyzer Libraries\n",
    "import nltk\n",
    "from nltk.sentiment.vader import SentimentIntensityAnalyzer\n",
    "\n",
    "# Create a local StreamingContext with two working threads and batch
interval of 1 second\n",
    "sc = SparkContext(\"local[2]\", \"TextFileStreamDemo\")\n",
    "ssc = StreamingContext(sc, 1)\n",
    "\n",
    "# Download Natural Language Processing Library\n",
```

```
    "nltk.download('vader_lexicon')\n",
    "```\n",
    "\n",
    "<br/>\n",
    "<br/>\n",
    "\n",
    "#### Read from the Text File\n",
    "Use the `textFileStream` method to read data from the directory containing your text file. Note that this method reads new data from files added to the directory during the execution of the stream.\n",
    "\n",
    "<br/>\n",
    "<br/>\n",
    "\n",
    "``` python\n",
    "# Directory where new data files will be placed (for streaming)\n",
    "directory = \"./textfile_directory\"\n",
    "\n",
    "# Create a DStream that reads new text files added to `directory`\n",
    "lines = ssc.textFileStream(directory)\n",
    "```\n",
    "\n",
    "<br/>\n",
    "<br/>\n",
    "\n",
    "### 4- Process the Stream\n",
    "In the field of computational linguistics and natural language processing, sentiment analysis aims to determine the attitude of a speaker, writer, or other subject with respect to some topic or the overall contextual polarity or emotional reaction to a document, interaction, or event. A basic task in sentiment analysis is classifying the polarity of a given text at the document, sentence, or feature/aspect level—whether the expressed opinion in a document, a sentence or an entity feature/aspect is positive, negative, or neutral. For example, consider the following three text inputs:\n",
    "\n",
    "“I love ice cream a lot”\n",
    "\n",
    "“I dislike ice cream a lot”\n",
    "\n",
    "“ice cream is made from milk”\n",
    "\n",
    "One would expect that the polarity of the first is (rather) positive, of the second is (rather) negative and of the third is (rather) neutral. The word “rather” is used here to express subjectivity, since humans not always agree about the polarity of a sentence. We rely on an
```

```
out-of-the-shelf library to perform sentiment analysis. \n",
    "\n",
    "Implement the logic for processing each line in the file. This is where you would add your sentiment analysis code. The output of your code should include the percentage of positive, negative and neutral sentences, respectively. \n",
    "\n",
    "<br/>\n",
    "<br/>\n",
    "\n",
    "``` python\n",
    "# THIS FUNCTION PRINTS EACH LINE\n",
    "lines.pprint()\n",
    "\n",
    "\n",
    "\n",
    "\n",
    "# THESE FUNCTIONS PERFORM SENTIMENT ANALYSIS\n",
    "# Initialize the SentimentIntensityAnalyzer from NLTK\n",
    "sid = SentimentIntensityAnalyzer()\n",
    "\n",
    "# Perform sentiment analysis on each sentence\n",
    "def analyze_sentiment(sentence):\n",
    "    score = sid.polarity_scores(sentence)\n",
    "    return  score, sentence\n",
    "\n",
    "# Apply sentiment analysis function to each sentence in the stream\n",
    "sentiment_analysis = lines.map(analyze_sentiment)\n",
    "\n",
    "# Print sentiment analysis for each batch interval\n",
    "sentiment_analysis.pprint()\n",
    "\n",
    "```\n",
    "\n",
    "<br/>\n",
    "<br/>\n",
    "\n",
    "### 5- Start the Stream\n",
    "Start the Spark Streaming process and await termination.\n",
    "\n",
    "<br/>\n",
    "<br/>\n",
    "\n",
    "``` python\n",
    "# Start the computation\n",
    "ssc.start()\n",
```

```
    "\n",
    "# Wait for the computation to terminate\n",
    "ssc.awaitTermination()\n",
    "```\n",
    "\n",
    "<br/>\n",
    "<br/>\n",
    "\n",
    "### 6- Activate the Stream\n",
    "Once your python script is running you may notice that it will not be streaming your textfile. To simulate a file being added to your monitored folder please do the following:\n",
    "1) Open your text file\n",
    "2) Add or delete a couple characters\n",
    "3) Save the file - as soon as you save it watch your terminal for updates\n",
    "\n",
    "### 7- Expected Warnings\n",
    "It is normal if you see the following warnings in your terminal. These will not effect your program:\n",
    "```bash\n",
    "Setting default log level to \"WARN\".\n",
    "To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).\n",
    "24/03/21 00:38:28 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable\n",

"/usr/local/lib/python3.11/site-packages/pyspark/streaming/context.py:72: FutureWarning: DStream is deprecated as of Spark 3.4.0. Migrate to Structured Streaming.\n",
    "  warnings.warn(\n",
    "\n",
    "```\n"
   ]
  },
  {
   "cell_type": "markdown",
   "metadata": {},
   "source": [
    "*Note: If you are using Macbook with M1/M2/M3 chip, there might be some problems running spark streaming application. Please consider using Macbook with Intel chip or other devices*"
   ]
  },
  {
```

```
   "cell_type": "markdown",
   "metadata": {},
   "source": [
    "## Part 2 : Other Textual Analytics \n",
    "\n",
    "Repeat steps 1-6 to perform other real-time analysis tasks such
as:\n",
    "- Word count: Count the occurrence of each word in the stream.\n",
    "- Top 10 words: Retrieve the top 10 most frequent words in the
stream."
   ]
  },
  {
   "cell_type": "markdown",
   "metadata": {},
   "source": [
    "# Deliverables\n",
    "- The python scripts for \"word count\" and \"top 10 words\" (include
both)\n",
    "- Show the results of running your code on the provided text file.\n",
    "- A documentation/report which includes the performance of each
analysis (time taken to complete each task), the architecture of your
stream processing system, and answers to the following questions:\n",
    "   - Is the performance real-time?\n",
    "   - How this architecture (of the entire spark streaming process)
would change if you receive streaming data from an external source with a
high injection rate?"
   ]
  },
  {
   "cell_type": "markdown",
   "metadata": {},
   "source": [
    "# Appendix – Useful Online Resources and Tutorials"
   ]
  },
  {
   "cell_type": "markdown",
   "metadata": {},
   "source": [
    "Spark Streaming Programming Guide\n",

"http://spark.apache.org/docs/latest/streaming-programming-guide.html\n",
    "\n",
    "Python Streaming Examples\n",
```

```
"https://github.com/apache/spark/tree/master/examples/src/main/python/strea
ming\n",
    "\n",
    "An easy-to-use Python library for accessing the Twitter API\n",
    "http://www.tweepy.org/\n",
    "\n",
    "Tutorial on Medium: Apache Spark at a Glance\n",

"https://medium.com/cloudnesil/apache-spark-at-a-glance-7088b9fe5ef5\n",
    "\n",
    "Apache Spark General Tutorial\n",
    "https://www.toptal.com/spark/introduction-to-apache-spark\n",
    "\n",
    "Apache Spark Streaming Tutorial: Identifying Trending Twitter
Hashtags\n",
    "https://www.toptal.com/apache/apache-spark-streaming-twitter\n",
    "\n",
    "Apache Spark Streaming with Twitter (and Python)\n",

"https://www.linkedin.com/pulse/apache-spark-streaming-twitter-python-laure
nt-weichberger/\n",
    "\n",
    "Twitter-Sentiment-Analysis-Using-Spark-Streaming-And-Kafka\n",

"https://github.com/sridharswamy/Twitter-Sentiment-Analysis-Using-Spark-Str
eaming-And-Kafka\n",
    "\n",
    "Twitter Sentiment Analysis using Python\n",

"https://www.geeksforgeeks.org/twitter-sentiment-analysis-using-python/\n",
    "\n",
    "Sentiment Analysis on Reddit News Headlines with Python’s Natural
Language Toolkit (NLTK)\n",

"https://www.learndatasci.com/tutorials/sentiment-analysis-reddit-headlines
-pythons-nltk\n",
    "\n",
    "Potential fix for Spark Streaming Application on Macbook with M1/M2/M3
chip\n",

"https://betterprogramming.pub/change-one-line-of-code-to-make-your-spark-j
obs-work-again-12f492bc2b07\n"
   ]
  },
  {
   "cell_type": "markdown",
```

```
   "metadata": {},
   "source": [
    "# Good Luck!"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": null,
   "metadata": {},
   "outputs": [],
   "source": []
  }
 ],
 "metadata": {
  "kernelspec": {
   "display_name": "Python 3 (ipykernel)",
   "language": "python",
   "name": "python3"
  },
  "language_info": {
   "codemirror_mode": {
    "name": "ipython",
    "version": 3
   },
   "file_extension": ".py",
   "mimetype": "text/x-python",
   "name": "python",
   "nbconvert_exporter": "python",
   "pygments_lexer": "ipython3",
   "version": "3.9.13"
  }
 },
 "nbformat": 4,
 "nbformat_minor": 2
}
```