Intermediary_Data_Processing

April 10, 2023

0.1 Intermediary Data Processing

This file is included as a necessary intermediary step to bridge the connection between data retrieval and model building due to technological barriers in accessing cloud software.

[1]:
```
import pandas as pd
import csv
import datetime
import yfinance as yf
import numpy as np
from vaderSentiment.vaderSentiment import SentimentIntensityAnalyzer
from bs4 import BeautifulSoup as bs
import requests
from pyspark.sql.functions import sum,max,min,mean,count
import datetime as dt
import pyspark
from pyspark.sql import SparkSession
import findspark
import yaml
from yaml.loader import SafeLoader
from os.path import abspath

warehouse_location = abspath('spark-warehouse')
with open('cfg.yml') as f:
    config = yaml.load(f, Loader = SafeLoader)

    #create spark connection
findspark.init()
spark = SparkSession.builder \
    .master(config['spark']['spark_master'])\
    .appName('retrieve')\
    .enableHiveSupport()\
    .config('spark.sql.warehouse.dir', warehouse_location)\
    .config(config['spark']['spark_jars'], config['spark']['spark_jars_path'])\
    .config('spark.cores.max', '2')\
    .config('spark.executor.cores', '2')\
    .getOrCreate()
spark.sparkContext.setLogLevel("WARN")
```

```
spark

#create database config details
url = config['postgres']['url']
properties = {
    'user': config['postgres']['user'],
    'password' : config['postgres']['user'],
    'url': url,
    'driver': config['postgres']['driver']

}
```

```
23/04/10 14:54:53 WARN Utils: Your hostname, cis6180 resolves to a loopback
address: 127.0.1.1; using 10.0.2.15 instead (on interface enp0s3)
23/04/10 14:54:53 WARN Utils: Set SPARK_LOCAL_IP if you need to bind to another
address
23/04/10 14:54:54 WARN NativeCodeLoader: Unable to load native-hadoop library
for your platform… using builtin-java classes where applicable

Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use
setLogLevel(newLevel).
```

1 Retrieve Data from Database and Write to Csv

[2]:
```
import pyspark.pandas as ps

#retrieve data from database
def return_data(ticker_list, from_date, to_date):
    sentiment = spark.read.format("jdbc")\
        .option("url", "jdbc:postgresql://localhost:5432/financials") \
        .option("driver", "org.postgresql.Driver").option("dbtable",␣
 ↪"sentiment") \
        .option("user", "adam").option("password", "green").load()
    finance = spark.read.format("jdbc")\
        .option("url", "jdbc:postgresql://localhost:5432/financials") \
        .option("driver", "org.postgresql.Driver").option("dbtable",␣
 ↪"company_data") \
        .option("user", "adam").option("password", "green").load()
    full_data = finance.join(sentiment, ['date', 'ticker'], 'leftouter').
 ↪fillna(0)
    df_list = []
    for ticker in ticker_list:
        working_data = full_data[full_data['ticker'] == ticker]
        working_data = working_data.sort('date', ascending = True).
 ↪filter((working_data.date >= from_date) & (working_data.date<= to_date)).
 ↪toPandas().set_index('date')
```

```
        working_data = working_data[~working_data.index.duplicated()]
        working_data.to_csv('data/'+ticker+'_dataframe.csv')
        df_list.append(working_data)
return_data(['MSFT', 'GOOG', 'AMZN', 'TSLA', 'NFLX'], "2016-01-01",␣
 ↪"2023-03-01")
```

```
WARNING:root:'PYARROW_IGNORE_TIMEZONE' environment variable was not set. It is
required to set this environment variable to '1' in both driver and executor
sides if you use pyarrow>=2.0.0. pandas-on-Spark will set it for you but it does
not work if there is a Spark context already launched.
```

[ ]:
```
spark.stop()
```