

LAPORAN
RENCANA TUGAS MANDIRI (RTM) ke-V
MATA KULIAH BIG DATA KELAS B



DISUSUN OLEH:
Aulia Nur Fitriani (21083010051)

DOSEN PENGAMPU:
Kartika Maulida Hindrayani, S.Kom., M.Kom.

PROGRAM STUDI SAINS DATA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS PEMBANGUNAN NASIONAL "VETERAN"
JAWA TIMUR
2023

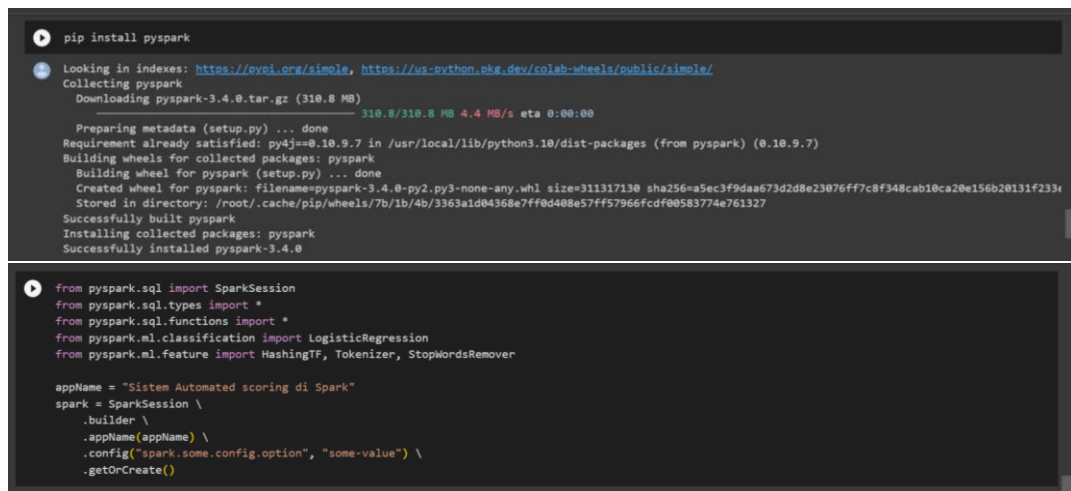
Tugas 5: Membuat Automated Scoring System Menggunakan PySpark

Automated Scoring System

Automated Scoring System (Sistem Penilaian Otomatis) adalah sistem yang menggunakan teknologi komputer dan algoritma untuk melakukan penilaian atau skoring secara otomatis terhadap jawaban atau respon siswa/mahasiswa dalam suatu ujian atau tugas tertentu.

Langkah-langkah

1. Install Modul PySpark terlebih dahulu dan Import Modul yang telah terinstall



```
pip install pyspark

Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Collecting pyspark
  Downloading pyspark-3.4.0.tar.gz (310.8 MB)
    310.8/310.8 MB 4.4 MB/s eta 0:00:00
  Preparing metadata (setup.py) ... done
Requirement already satisfied: py4j==0.10.9.7 in /usr/local/lib/python3.10/dist-packages (from pyspark) (0.10.9.7)
Building wheels for collected packages: pyspark
  Building wheel for pyspark (setup.py) ... done
  Created wheel for pyspark: filename=pyspark-3.4.0-py2.py3-none-any.whl size=311317130 sha256=a5ec3f9daa673d2d8e23076ff7c8f348cab10ca20e15eb20131f233e
  Stored in directory: /root/.cache/pip/wheels/7b/1b/4b/3363a1d04368e7ffed408e57ff57966fcdfe0583774e761327
Successfully built pyspark
Installing collected packages: pyspark
Successfully installed pyspark-3.4.0

from pyspark.sql import SparkSession
from pyspark.sql.types import *
from pyspark.sql.functions import *
from pyspark.ml.classification import LogisticRegression
from pyspark.ml.feature import HashingTF, Tokenizer, StopWordsRemover

appName = "Sistem Automated scoring di Spark"
spark = SparkSession \
    .builder \
    .appName(appName) \
    .config("spark.some.config.option", "some-value") \
    .getOrCreate()
```

Dalam PySpark:

- SparkSession adalah modul yang menginisialisasi dan mengatur sesi Spark, digunakan untuk membuat DataFrame, mengakses sumber data, dan menjalankan operasi Spark.
- LogisticRegression adalah modul yang membangun model regresi logistik menggunakan metode optimasi berbasis gradien.
- HashingTF adalah modul yang menghitung frekuensi kemunculan kata dalam dokumen dan memetakan kata-kata ke dalam representasi vektor menggunakan teknik hashing.
- Tokenizer adalah modul yang memecah teks menjadi kata-kata individu atau "token" untuk mempermudah pengolahan.
- StopWordsRemover adalah modul yang menghapus kata-kata umum atau sering muncul yang tidak memberikan informasi penting dalam analisis teks.

2. Import dan Read Data

```
[ ] import os
os.environ["PSPARK_SUBMIT_ARGS"] = "--master local[*] pyspark-shell"

[ ] import pandas as pd

essay = spark.read.csv("/content/training_data_essay.csv", inferSchema=True, header=True)
essay.show(truncate=False, n=5)
```

npm	nama_peserta	jawaban
0	Admin	Tidak, Hanya membutuhkan satu karena satu software sesuai dengan keahlian
0	Admin	Biaya dihitung berdasarkan waktu pengerjaan dan tingkat kesulitan
0	Admin	Hak cipta adalah hak eksklusif bagi pencipta atau penerima hak untuk mengumumkan atau memperbanyak ciptaannya atau memberi izin untuk
0	Admin	Dijelaskan kepada klien jika huruf terlalu besar maka kita tidak dapat memasukkan cukup banyak kata pada setiap baris agar pembaca le
0	Admin	1. Melindungi dan menjamin pekerja atas hak keselamatannya 2. Memlihara sumber produksi agar dapat digunakan secara aman dan efisien

only showing top 5 rows

Memanggil data dengan nama “training_data_essay” dengan library pandas dan dapat dilihat bahwa output nya adalah 5 data teratas.

```
#pilih data hanya dari kolom "SentimentText" dan kolom "Sentiment".
#pilih nilai di kolom "Sentiment" ke tipe integer dan mengganti nya dengan "label"
data = essay.select('npm', 'jawaban', 'soal', 'skor_per_soal')
data.show(5)
```

npm	jawaban	soal	skor_per_soal
0	Tidak, Hanya memb...	1	100.0
0	Biaya dihitung be...	2	100.0
0	Hak cipta adalah ...	3	100.0
0	Dijelaskan kepada...	4	100.0
0	1. Melindungi dan...	5	100.0

only showing top 5 rows

Dari berbagai kolom hanya dipilih kolom “npm”, “jawaban”, “soal”, dan “skor_per_soal”.

Dengan menampilkan 5 kolom

3. Metode Hashing

Metode hashing adalah teknik yang digunakan untuk mengonversi data menjadi nilai hash, yang biasanya berupa bilangan integer atau string yang pendek. Hashing adalah proses mengambil input data dengan panjang variabel dan mengubahnya menjadi nilai hash dengan panjang tetap.

```
from pyspark.sql.functions import hash, abs
data = essay.withColumn("HashValue", hash("jawaban"))
data.show()
```

npm	nama_peserta	jawaban	soal	skor_per_soal	HashValue
0	Admin	Tidak, Hanya memb...	1	100.0	-2059296905
0	Admin	Biaya dihitung be...	2	100.0	1183180174
0	Admin	Hak cipta adalah ...	3	100.0	1232762403
0	Admin	Dijelaskan kepada...	4	100.0	-2035408785
0	Admin	1. Melindungi dan...	5	100.0	1588395990
0	Admin	Ruang Komputer, P...	6	100.0	339970513
0	Admin	Aturlah posisi pe...	7	100.0	50850002
0	Admin	Posisi Kepala dan...	8	100.0	-945877996
0	Admin	1. Kecocokan soft...	9	100.0	1576366224
0	Admin	1. Fokus dan expo...	10	100.0	-1905649442
0	Admin	1. Peralatan yang...	11	100.0	550139146
0	Admin	1. Dibuat grafik ...	12	100.0	1727767227
1121020033	AP	tidak, cuma mengi...	1	52.7	1947733435
1121020033	AP	biaya dihitung be...	2	42.86	-1139063335
1121020033	AP	hak membuat merup...	3	42.16	122676417
1121020033	AP	dipaparkan pada k...	4	27.19	-1054163002
1121020033	AP	1. mencegah serta...	5	44.14	1990940339
1121020033	AP	ruang komputer, p...	6	100.0	1770907636
1121020033	AP	aturlah posisi fi...	7	57.68	-463479969
1121020033	AP	posisi kepala ser...	8	45.71	-412537011

only showing top 20 rows

```
[ ] dt = data.select("soal", "skor_per_soal", "HashValue")
```

Kolom yang dipilih adalah “soal”, “skor_per_soal”, dan “hashvalue”.

4. Membagi Data Training dan Testing

Training dan testing adalah dua konsep penting dalam pembelajaran mesin (machine learning) yang digunakan untuk menguji dan mengevaluasi kinerja model yang telah dibuat. Pembagian data menjadi data latih (training data) dan data uji (testing data) adalah langkah penting dalam pengembangan model yang baik.

Dalam konteks pembagian data menjadi rasio 80:20, artinya 80 persen dari total data digunakan sebagai data latih, sedangkan 20 persen digunakan sebagai data uji. Tujuan dari pembagian ini adalah untuk menggunakan sebagian besar data untuk melatih model (data latih) dan mempertahankan sebagian kecil untuk menguji kinerja model (data uji).

```
#bagi data jadi 20% dan 80%
splits = data.randomSplit([0.8, 0.2])
train = splits[0].withColumnRenamed("skor_per_soal", "label")
test = splits[1].withColumnRenamed("skor_per_soal", "trueLabel")

#hitung jumlah data training dan testing
train_rows = train.count()
test_rows = test.count()
print ("Jumlah baris data training:", train_rows,
      ", jumlah baris data testing:", test_rows)

Jumlah baris data training: 88 , jumlah baris data testing: 32

train.show()
```

	npm nama_peserta	jawaban soal	label	HashValue
0	Admin 1. Fokus dan expo...	10 100.0	-1905649442	
0	Admin 1. Kecocokan soft...	9 100.0	1576366224	
0	Admin 1. Peralatan yang...	11 100.0	550139146	
0	Admin Aturlah posisi pe...	7 100.0	50850002	
0	Admin Biaya dihitung be...	2 100.0	1183180174	
0	Admin Dijelaskan kepada...	4 100.0	-2035408785	
0	Admin Posisi Kepala dan...	8 100.0	-945877996	
0	Admin Ruang Komputer, P...	6 100.0	339970513	
0	Admin Tidak, Hanya memb...	11 100.0	-2059296905	
1120020017	RDW aturlah posisi pe...	7 86.22	-1392782412	
1120020017	RDW biaya dihitung be...	2 84.52	-219318287	
1120020017	RDW dibuat grafik yan...	12 86.53	-902408772	
1120020017	RDW emperbanyak cipta...	3 47.67	944450734	
1120020017	RDW fokus dan exposur...	10 85.75	1988679646	
1120020017	RDW kecocokan softwar...	9 65.89	-1544668284	
1120020017	RDW memlihara sumber ...	5 88.3	40672765	
1120020017	RDW peralatan yang di...	11 80.09	-183887780	
1120020017	RDW posisi kepala dan...	8 89.17	1044171719	
1120020017	RDW posisi tubuh, pos...	6 90.37	-1702735646	
1120020017	RDW tidak, hanya memb...	1 100.0	-256638840	

```
#mendefinisikan model
from pyspark.ml.recommendation import ALS
from pyspark.ml.evaluation import RegressionEvaluator
als = ALS(maxIter=5, regParam=0.01, userCol='HashValue', itemCol='soal', ratingCol='label')

als = ALS(maxIter=5, regParam=0.01, userCol='HashValue',
          itemCol='soal', ratingCol='label')
#training model dengan fungsi ".fit()"
model = als.fit(train)
print("Model telah selesai ditraining!")

Model telah selesai ditraining!
```

5. Metode ALS (Alternating Least Squares)

Metode ALS (Alternating Least Squares) adalah algoritma yang digunakan untuk memecahkan masalah rekomendasi berdasarkan matriks faktorisasi. Tujuan utama dari metode ALS adalah mengisi nilai yang hilang atau tidak diketahui dalam matriks user-item dengan perkiraan nilai yang akurat.

```

prediction = model.transform(test)
prediction.show()

```

	npem	nama_peserta	jawaban	soal	trueLabel	HashValue	prediction
1121020032		ZA	dibuat grafik yan...	12	86.53	-902409772	86.529945
0		Admin	1. Dibuat grafik ...	12	100.0	1727767227	NaN
1121020024		IDP	ruang komputer, p...	6	100.0	1770907636	99.99995
1121020024		IDP	hak cipta adalah ...	3	83.43	-1876419705	83.42995
1121020032		ZA	hak cipta adalah ...	3	91.71	770340049	91.70994
0		Admin	Hak cipta adalah ...	3	100.0	1232762403	NaN
1121020036		DAR	hak cipta adalah ...	3	76.06	1698979133	NaN
1121020024		IDP	melindungi dan me...	5	74.73	1470717550	NaN
0		Admin	1. Melindungi dan...	5	100.0	1588395990	NaN
1121020033		AP	dipaparkan pada k...	4	27.19	-1054163002	NaN
1120020017		RDW	dijelaskan kepada...	4	72.06	-683533012	72.059975
1121020035		BS	dijelaskan kepada...	4	78.52	-551735710	NaN
1121020032		ZA	posisi kepala dan...	8	89.17	1044171719	89.16995
1121020024		IDP	posisi kepala dan...	8	100.0	1782344444	99.999954
1121020033		AP	aturlah posisi fi...	7	57.68	-463479969	NaN
1121020035		BS	aturlah posisi pe...	7	59.87	-437364728	NaN
1121020024		IDP	aturlah posisi pe...	7	100.0	1123192925	NaN
1121020024		IDP	fokus dan exposur...	10	50.71	1698953930	NaN
1121020033		AP	1. perlengkapan y...	11	41.99	-2137389145	NaN
1121020032		ZA	peralatan yang di...	11	83.22	-275485461	NaN

only showing top 20 rows

```

[ ] from pyspark.ml.evaluation import RegressionEvaluator

evaluator = RegressionEvaluator(
    labelCol="trueLabel", predictionCol="prediction", metricName="rmse")
rmse = evaluator.evaluate(prediction)
print ("Root Mean square Error (RMSE):", rmse)

Root Mean square Error (RMSE): nan

[ ] prediction.count()
a = prediction.count()
print("jumlah baris sebelum di hapus data kosong: ", a)
cleanPred = prediction.dropna(how="any", subset=["prediction"])
b = cleanPred.count()
print("jumlah baris setelah di hapus data kosong: ", b)
print("jumlah baris data kosong: ", a-b)

jumlah baris sebelum di hapus data kosong: 32
jumlah baris setelah di hapus data kosong: 14
jumlah baris data kosong: 18

[ ] #evaluasi mse
evaluator = RegressionEvaluator(metricName="mse", labelCol="trueLabel", predictionCol="prediction")
mse = evaluator.evaluate(prediction)
print("Mean Squares Error = " + str(mse))

Mean Squares Error = nan

```

- Mean Square Error (MSE): MSE mengukur rata-rata dari kuadrat selisih antara nilai prediksi dan nilai sebenarnya. Setiap selisih di kuadrat agar menjadi positif dan untuk memberikan bobot yang lebih besar pada kesalahan yang lebih besar. MSE memberikan gambaran tentang sejauh mana variabilitas data dapat dijelaskan oleh model. Semakin kecil nilai MSE, semakin baik model dapat memprediksi nilai sebenarnya.
- Root Mean Square Error (RMSE): RMSE adalah akar kuadrat dari MSE. Hal ini dilakukan untuk mengembalikan nilai ke skala aslinya. RMSE digunakan untuk memberikan interpretasi yang lebih intuitif tentang kesalahan model, karena memiliki satuan yang sama dengan variabel target atau nilai sebenarnya. Semakin kecil nilai RMSE, semakin akurat model dalam memprediksi nilai sebenarnya.