Assignment 1

Aim: Write a program using Multivariate Analysis methods on selected Big Data.

Objective: To apply multivariate analysis techniques on a large dataset using PySpark.

Theory:
To analyze the relationships between multiple variables efficiently in a distributed environment.

Multivariate Analysis is a set of statistical techniques used to analyze data that involves multiple variables. It helps in understanding relationships, patterns, and dependencies among multiple features simultaneously. This analysis is widely used in fields like finance, marketing, and machine learning to extract insights from complex datasets.

Principal Component Analysis (PCA):
Principal Component Analysis (PCA) is a dimensionality reduction technique used to transform high-dimensional data into a lower-dimensional space while preserving as much variance as possible. It is widely used in machine learning, finance, image processing, and big data analytics for feature extraction, noise reduction, and visualization.

How PCA Works?

1. **Standardization of Data**
    - Normalize the data to have a mean of zero and unit variance.
2. **Compute the Covariance Matrix**
    - Understand relationships between different features.
3. **Eigen Decomposition**
    - Find eigenvalues and eigenvectors of the covariance matrix.
4. **Select Principal Components**
    - Choose top k eigenvectors (principal components) based on eigenvalues.
5. **Transform the Data**
    - Project data onto the new feature space with reduced dimensions.

Dataset Used:
Flights Dataset from Kaggle
The dataset consists of flight records, including departure delays and arrival delays. It contains the following key attributes:

- **DayofMonth:** The day of the month the flight was scheduled.
- **DayOfWeek:** The day of the week the flight was scheduled.
- **Carrier:** The airline carrier code.
- **OriginAirportID:** The airport ID of the origin.

- **DestAirportID:** The airport ID of the destination.
- **DepDelay:** Departure delay in minutes.
- **ArrDelay:** Arrival delay in minutes.

Model Selection & Hyperparameters:

- **PCA (n=2):** Reduced data to 2 dimensions for visualization and analysis.

Implementation:

1. **Initialize SparkSession** - Creates a Spark environment for distributed computing.
2. **Load Dataset** - Reads flight data from a CSV file.
3. **Feature Selection** - Selects relevant features for PCA.
4. **Standardization** - Scales feature values to ensure uniform importance.
5. **Apply PCA** - Reduces the dimensionality of data to 2 principal components.
6. **Display Results** - Shows transformed data with principal components.

Interpretation of Results:

- **PCA Output**: The 2D scatter plot of PC1 vs PC2 displayed diverse flight delay behaviors.
- **Distinct Delay Patterns**: Flights with significant delays formed distinct groups, highlighting differences in departure and arrival delays.
- **Feature Significance**: The variance in PCA components indicated the importance of departure and arrival delay features.

This modified assignment uses the flights.csv dataset instead of financial transactions while maintaining the original structure and methodology.

multivariate-analysis-1

March 18, 2025

[1]:
```
import kagglehub

# Download latest version
path = kagglehub.dataset_download("tylerx/flights-and-airports-data")

print("Path to dataset files:", path)
```

```
C:\Users\Harshal\OneDrive\Desktop\py_spark project\myenv\Lib\site-
packages\tqdm\auto.py:21: TqdmWarning: IProgress not found. Please update
jupyter and ipywidgets. See
https://ipywidgets.readthedocs.io/en/stable/user_install.html
  from .autonotebook import tqdm as notebook_tqdm

Warning: Looks like you're using an outdated `kagglehub` version (installed:
0.3.7), please consider upgrading to the latest version (0.3.10).
Path to dataset files:
C:\Users\Harshal\.cache\kagglehub\datasets\tylerx\flights-and-airports-
data\versions\1
```

[2]:
```
import numpy as np
import pandas as pd
```

[3]:
```
import pyspark
from pyspark.sql import SparkSession
from pyspark.sql.functions import col, sum, when

from pyspark.ml.feature import StandardScaler,StringIndexer, VectorAssembler,␣
 ↪VectorIndexer, OneHotEncoder
from pyspark.ml.evaluation import MulticlassClassificationEvaluator
from pyspark.mllib.evaluation import MulticlassMetrics
from pyspark.ml import Pipeline
from pyspark.ml.classification import LogisticRegression,␣
 ↪DecisionTreeClassifier, RandomForestClassifier, GBTClassifier
```

[4]:
```
spark = SparkSession.builder.appName("flight").master("local[1]").config("spark.
 ↪sql.shuffle.partitions", "1").config("spark.driver.memory", "4g").
 ↪getOrCreate()
```

[5]:
```
data = spark.read.csv('C:/Users/Harshal/.cache/kagglehub/datasets/tylerx/
↪flights-and-airports-data/versions/1/flights.csv', inferSchema=True,␣
↪header=True)
data.show(10)
```

```
+----------+---------+-------+---------------+-------------+--------+--------+
|DayofMonth|DayOfWeek|Carrier|OriginAirportID|DestAirportID|DepDelay|ArrDelay|
+----------+---------+-------+---------------+-------------+--------+--------+
|        19|        5|     DL|          11433|        13303|      -3|       1|
|        19|        5|     DL|          14869|        12478|       0|      -8|
|        19|        5|     DL|          14057|        14869|      -4|     -15|
|        19|        5|     DL|          15016|        11433|      28|      24|
|        19|        5|     DL|          11193|        12892|      -6|     -11|
|        19|        5|     DL|          10397|        15016|      -1|     -19|
|        19|        5|     DL|          15016|        10397|       0|      -1|
|        19|        5|     DL|          10397|        14869|      15|      24|
|        19|        5|     DL|          10397|        10423|      33|      34|
|        19|        5|     DL|          11278|        10397|     323|     322|
+----------+---------+-------+---------------+-------------+--------+--------+
only showing top 10 rows
```

[6]:
```
data.printSchema()
```

```
root
 |-- DayofMonth: integer (nullable = true)
 |-- DayOfWeek: integer (nullable = true)
 |-- Carrier: string (nullable = true)
 |-- OriginAirportID: integer (nullable = true)
 |-- DestAirportID: integer (nullable = true)
 |-- DepDelay: integer (nullable = true)
 |-- ArrDelay: integer (nullable = true)
```

[7]:
```
data.columns
```

[7]:
```
['DayofMonth',
 'DayOfWeek',
 'Carrier',
 'OriginAirportID',
 'DestAirportID',
 'DepDelay',
 'ArrDelay']
```

[8]:
```
data = data.select("DayofMonth", "DayOfWeek", "Carrier",␣
↪"OriginAirportID","DestAirportID", "DepDelay", "ArrDelay", ((col("ArrDelay")␣
↪> 15).cast("Int").alias("Late")))
```

[9]:
```
data.show(10)
```

```
+----------+---------+-------+--------------+-------------+--------+--------+--
--+
|DayofMonth|DayOfWeek|Carrier|OriginAirportID|DestAirportID|DepDelay|ArrDelay|La
te|
+----------+---------+-------+--------------+-------------+--------+--------+--
--+
|        19|        5|     DL|         11433|        13303|      -3|       1|
0|
|        19|        5|     DL|         14869|        12478|       0|      -8|
0|
|        19|        5|     DL|         14057|        14869|      -4|     -15|
0|
|        19|        5|     DL|         15016|        11433|      28|      24|
1|
|        19|        5|     DL|         11193|        12892|      -6|     -11|
0|
|        19|        5|     DL|         10397|        15016|      -1|     -19|
0|
|        19|        5|     DL|         15016|        10397|       0|      -1|
0|
|        19|        5|     DL|         10397|        14869|      15|      24|
1|
|        19|        5|     DL|         10397|        10423|      33|      34|
1|
|        19|        5|     DL|         11278|        10397|     323|     322|
1|
+----------+---------+-------+--------------+-------------+--------+--------+--
--+
only showing top 10 rows
```

[10]:
```
null_counts = {
    column: data.filter(col(column).isNull()).count() for column in data.columns
}

for column, count in null_counts.items():
    print(f"{column}: {count} null values")
```

```
DayofMonth: 0 null values
DayOfWeek: 0 null values
Carrier: 0 null values
OriginAirportID: 0 null values
DestAirportID: 0 null values
DepDelay: 0 null values
ArrDelay: 0 null values
Late: 0 null values
```

[11]:
```
strIdx = StringIndexer(inputCol = "Carrier", outputCol = "CarrierIdx")
str_encoder = OneHotEncoder(inputCol='CarrierIdx',outputCol='CarrierVec')
```

[12]:
```
output = strIdx.fit(data).transform(data)
```

[13]:
```
feature_cols = ['DayofMonth',
 'DayOfWeek',
 'CarrierIdx',
 'OriginAirportID',
 'DestAirportID',
 'ArrDelay',]
```

[14]:
```
assembler = VectorAssembler(inputCols=feature_cols, outputCol="features")
```

[15]:
```
output = assembler.transform(output)
```

[16]:
```
final_data = output.select("features","Late")
final_data.show(5)
```

```
+--------------------+----+
|            features|Late|
+--------------------+----+
|[19.0,5.0,1.0,114…|   0|
|[19.0,5.0,1.0,148…|   0|
|[19.0,5.0,1.0,140…|   0|
|[19.0,5.0,1.0,150…|   1|
|[19.0,5.0,1.0,111…|   0|
+--------------------+----+
only showing top 5 rows
```

[17]:
```
train_data,test_data = final_data.randomSplit([0.7, 0.3], seed = 1)
```

[18]:
```
train_data, valid_data = train_data.randomSplit([0.9, 0.1], seed = 1)
```

[22]:
```
scaler = StandardScaler(inputCol="features", outputCol="scaled_features")
 scaler_model = scaler.fit(train_data)
 train_data = scaler_model.transform(train_data)
 test_data = scaler_model.transform(test_data)
 valid_data = scaler_model.transform(valid_data)
```

[23]:
```
from pyspark.ml.feature import PCA

pca = PCA(k=2, inputCol="scaled_features", outputCol="pca_features")


pca_model = pca.fit(train_data)
```

```
train_pca = pca_model.transform(train_data)
test_pca = pca_model.transform(test_data)
valid_pca = pca_model.transform(valid_data)

# Show PCA output
train_pca.select("pca_features").show(truncate=False)
```

```
+----------------------------------------+
|pca_features                            |
+----------------------------------------+
|[-4.713894746708608,-3.387776582510908] |
|[-4.554491540208862,-3.2418025342565118]|
|[-3.450930879826006,-2.231212969418383] |
|[-4.960337058640134,-3.6107821473702155]|
|[-4.825457422371118,-3.4872656450011106]|
|[-4.825457422371118,-3.4872656450011106]|
|[-4.8131956372557525,-3.4760368720584647]|
|[-4.8131956372557525,-3.4760368720584647]|
|[-4.800933852140387,-3.4648080991158188]|
|[-4.751886711678927,-3.4198930073452356]|
|[-4.617007075409911,-3.2963765049761307]|
|[-4.757520619373999,-3.424850748306091] |
|[-4.7452588342586335,-3.413621975363445]|
|[-4.659426338451078,-3.335020564764924] |
|[-4.610379197989618,-3.29010547299434]  |
|[-5.119452269907024,-3.750800770870742] |
|[-4.960049063407277,-3.604826722616345] |
|[-4.678028005753881,-3.34656494493549]  |
|[-4.616719080177055,-3.2904210802222607]|
|[-3.5376819900249292,-2.3022890612694233]|
+----------------------------------------+
only showing top 20 rows
```

[ ]: