

Relatório MVP

**PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO DE JANEIRO (PUC-Rio)
PÓS-GRADUAÇÃO EM CIÊNCIA DE DADOS E ANALYTICS**

Disciplina: Sprint: Engenharia de Dados

Projeto: Depressão Estudantil

Por Beatriz Siqueira Marques da Silva

Professor: Victor Almeida

Rio de Janeiro

Abril/2025

1. INTRODUÇÃO

Este relatório tem por objetivo construir uma pipeline de dados na plataforma Databricks Community Edition para realizar o processo de busca, coleta, modelagem, carga e análise de dados sobre depressão estudantil. Os resultados apresentados servirão para obtenção de nota na Sprint de Engenharia de Dados.

O estudo tem como foco a investigação dos fatores que influenciam a depressão entre estudantes universitários, buscando compreender como características demográficas, hábitos de vida e condições de saúde estão relacionados ao desenvolvimento de sintomas depressivos.

2. OBJETIVO

O problema a ser resolvido é o de investigar os fatores que influenciam a depressão entre estudantes universitários e identificar perfis de risco para possíveis intervenções.

Questionário para uma pipeline de dados sobre a Depressão Estudantil:

1. Identificar fatores críticos que contribuem para a depressão estudantil
2. Qual o volume de estudantes com depressão?
3. Qual é o perfil demográfico dos usuários, maior volume entre homens ou mulheres?
4. Qual a faixa etária dos homens e mulheres com depressão?

2.1. Detalhamento

Será apresentado abaixo, o processo detalhado de desenvolvimento do pipeline de dados na nuvem, desde a busca inicial dos dados até a análise final. O detalhamento está dividido nas seguintes etapas principais: busca pelos dados, coleta, modelagem, carga e análise.

3. BUSCA PELOS DADOS

Para a busca de dados, foi utilizada a base de dados disponível no Kaggle: [Student Depression Dataset](#), contendo informações como:

- **ID:** Identificador único para cada estudante.
- **Age:** Idade do estudante.
- **Gender:** Gênero (por exemplo, Masculino, Feminino).
- **City:** Região geográfica.
- **CGPA:** Média de notas ou outras pontuações acadêmicas.

- **Sleep Duration:** Duração média de sono diária.
- **Profession:** Profissão.
- **Work Pressure:** Pressão no trabalho.
- **Academic Pressure:** Pressão acadêmica.
- **Study Satisfaction:** Satisfação com os estudos.
- **Job Satisfaction:** Satisfação com o trabalho.
- **Dietary Habits:** Hábitos alimentares.

Esta base de dados contém informações detalhadas sobre estudantes, com o objetivo de analisar, compreender e prever os níveis de depressão nesse público. Os dados incluem variáveis demográficas (como idade, gênero e cidade), desempenho acadêmico (como CGPA), hábitos de vida (como duração do sono e hábitos alimentares), fatores relacionados à pressão acadêmica e profissional, níveis de satisfação com estudo e trabalho, entre outros.

4. COLETA

A coleta de dados foi realizada diretamente do Kaggle, onde um arquivo CSV contendo informações relevantes foi baixado e inserido manualmente no Databricks.

Para carregar esse arquivo no ambiente do Databricks, foi utilizado o Apache Spark, que permite a leitura eficiente dos dados, garantindo flexibilidade na manipulação e transformação do dataset.

Abaixo está o código utilizado para o carregamento e a análise dos dados:

```
02:43 PM (4s) 6

# Localização e tipo do arquivo
file_location = "/FileStore/tables/Student_Depression_Dataset/Student_Depression_Dataset.csv"
file_type = "csv"

# Opções do CSV
infer_schema = "false"
first_row_is_header = "false"
delimiter = ","

df = spark.read.format(file_type) \
    .option("inferSchema", infer_schema) \
    .option("header", first_row_is_header) \
    .option("sep", delimiter) \
    .load(file_location)

display(df)

▶ (2) Spark Jobs
```

MVP_Student_Depression Python ▾
File Edit View Run Help Last edit was 9 minutes ago

▶ Run all My Cluster ▾ Share Publish

▶ (2) Spark Jobs

▶ df: pyspark.sql.dataframe.DataFrame = [c0: string, c1: string ... 16 more fields]

Table ▾ + 🔍 ⚙️ 📄

	A _c _c0	A _c _c1	A _c _c2	A _c _c3	A _c _c4	A _c _c5	A _c _c6	A _c _c7	A _c _c8
1	id	Gender	Age	City	Profession	Academic Pressu...	Work Pressure	CGPA	Stu...
2	2	Male	33.0	Visakhapatna...	Student	5.0	0.0	8.97	2.0
3	8	Female	24.0	Bangalore	Student	2.0	0.0	5.9	5.0
4	26	Male	31.0	Srinagar	Student	3.0	0.0	7.03	5.0
5	30	Female	28.0	Varanasi	Student	3.0	0.0	5.59	2.0
6	32	Female	25.0	Jaipur	Student	4.0	0.0	8.13	3.0
7	33	Male	29.0	Pune	Student	2.0	0.0	5.7	3.0
8	52	Male	30.0	Thane	Student	3.0	0.0	9.54	4.0
9	56	Female	30.0	Chennai	Student	2.0	0.0	8.04	4.0
10	59	Male	28.0	Nagpur	Student	3.0	0.0	9.79	1.0
11	62	Male	31.0	Nashik	Student	2.0	0.0	8.38	3.0
12	83	Male	24.0	Nagpur	Student	3.0	0.0	6.1	3.0
13	91	Male	33.0	Vadodara	Student	3.0	0.0	7.03	4.0
14	94	Male	27.0	Kalyan	Student	5.0	0.0	7.04	1.0
15	4								

A seguir será apresentado o código utilizado para salvar a tabela permanentemente no Databricks assegurando que os dados sejam armazenados de forma persistente e possam ser acessados em futuras análises ou consultas:

MVP_Student_Depression Python ▾
File Edit View Run Help Last edit was 10 minutes ago

▶ Run all My Cluster ▾ Share Publish

A seguir será apresentado o código utilizado para salvar a tabela permanentemente no Databricks assegurando que os dados sejam armazenados de forma persistente e possam ser acessados em futuras análises ou consultas:

```

# Definir o nome da tabela permanente
permanent_table_name = "Student_Depression_csv"

# Salvar o DataFrame como uma tabela permanente no catálogo do Spark
df.write.format("delta").saveAsTable(permanent_table_name)

```

▶ (6) Spark Jobs

Abaixo, o código para a consulta à tabela de dados **Students Depression**:

```

%sql
SELECT * FROM Student_Depression_csv

```

▶ (2) Spark Jobs

MVP_Student_Depression Python

File Edit View Run Help Last edit was 10 minutes ago

Run all My Cluster Share Publish

	A _c _c0	A _c _c1	A _c _c2	A _c _c3	A _c _c4	A _c _c5	A _c _c6	A _c _c7	A _c _c8
1	id	Gender	Age	City	Profession	Academic Pressu...	Work Pressure	CGPA	Stuc
2	2	Male	33.0	Visakhapatna...	Student	5.0	0.0	8.97	2.0
3	8	Female	24.0	Bangalore	Student	2.0	0.0	5.9	5.0
4	26	Male	31.0	Srinagar	Student	3.0	0.0	7.03	5.0
5	30	Female	28.0	Varanasi	Student	3.0	0.0	5.59	2.0
6	32	Female	25.0	Jaipur	Student	4.0	0.0	8.13	3.0
7	33	Male	29.0	Pune	Student	2.0	0.0	5.7	3.0
8	52	Male	30.0	Thane	Student	3.0	0.0	9.54	4.0
9	56	Female	30.0	Chennai	Student	2.0	0.0	8.04	4.0
10	59	Male	28.0	Nagpur	Student	3.0	0.0	9.79	1.0
11	62	Male	31.0	Nashik	Student	2.0	0.0	8.38	3.0
12	83	Male	24.0	Nagpur	Student	3.0	0.0	6.1	3.0
13	91	Male	33.0	Vadodara	Student	3.0	0.0	7.03	4.0
14	94	Male	27.0	Kalyan	Student	5.0	0.0	7.04	1.0
15									

10,000+ rows | Truncated data | 7.99s runtime Refreshed 8 hours ago

5. MODELAGEM

A seguir estão as informações que detalham a organização e estruturação dos dados, visando otimizar seu uso para análises, relatórios e tomadas de decisão.

Foi adotado um modelo de dados baseado no **Esquema Estrela (Star Schema)**, que permite melhor eficiência nas consultas analíticas, adequado para consultas rápidas. O modelo será composto por uma ****tabela de fatos**** centralizada que contém as medições principais, e uma **tabela de dimensões** associada às características de cada estudante.

5.1. Estrutura do Modelo

A estrutura do modelo é composta por 2 principais tabelas: Tabela de Fatos e Tabela de Dimensões.

5.1.1. Tabela de Fatos: "Student_Depression_Facts"

A **tabela de fatos** central é composta pelo Status de Depressão de cada estudante, que é a variável de interesse (target). A tabela também contém métricas relacionadas, como os níveis de satisfação e pressão que ajudam a contextualizar o status de depressão.

Abaixo os dados contidos na tabela de fatos:

- **ID** (identificador único)
- **Depression_Status** (Status de depressão: "Yes" ou "No")
- **CGPA** (Índice de aproveitamento acadêmico)
- **Sleep Duration** (Duração média de sono)
- **Work Pressure** (Pressão no trabalho)

- **Academic Pressure** (Pressão acadêmica)
- **Study Satisfaction** (Satisfação com os estudos)
- **Job Satisfaction** (Satisfação com o trabalho, se aplicável)
- **Dietary Habits** (Hábitos alimentares)

5.1.2. Tabelas de Dimensões: "Student_Dimensions"

A **tabela de dimensões** contém informações descritivas relacionadas aos estudantes e são usadas para contextualizar as métricas da tabela de fatos.

Abaixo os dados contidos na tabela de dimensões:

- **ID** (identificador único)
- **Age** (Idade do estudante)
- **Gender** (Gênero do estudante: Male, Female, Other)
- **City** (Cidade ou região geográfica)

5.2. Catálogo de Dados - Student Depression Dataset

O **Catálogo de Dados** descreve os dados do **Student Depression Dataset** e seus respectivos domínios, garantindo clareza quanto aos valores possíveis para cada atributo.

Coluna	Descrição	Tipo de Dado	Domínio / Intervalo	Observações
ID	Identificador único do estudante	Numérico	Valor único por linha	Não deve conter valores nulos
Age	Idade do estudante	Numérico	15 a 30 anos	Verificar valores inconsistentes
Gender	Gênero do estudante	Categórico	Male, Female, Other	Normalizar capitalização
City	Cidade onde reside	Texto	Variado	Pode ser categorizado

Coluna	Descrição	Tipo de Dado	Domínio / Intervalo	Observações
CGPA	Índice acadêmico (nota média)	Numérico	0.0 a 4.0 ou 0 a 10	Verificar padrão de escala
Sleep Duration	Duração média de sono	Numérico	4 a 10 horas	Pode precisar padronização
Profession	Profissão exercida	Texto/Categórico	Estudante, Estagiário, CLT, etc.	Pode conter valores vazios
Work Pressure	Nível de pressão no trabalho	Categórico	Baixo, Médio, Alto	
Academic Pressure	Nível de pressão nos estudos	Categórico	Baixo, Médio, Alto	
Study Satisfaction	Satisfação com os estudos	Categórico	Baixo, Médio, Alto	
Job Satisfaction	Satisfação com o trabalho	Categórico	Baixo, Médio, Alto	
Dietary Habits	Hábitos alimentares	Categórico	Saudável, Irregular, Não Saudável	Pode ser subjetivo
Depression_Status	Status de depressão	Categórico	Yes, No	Variável-alvo (target)

6. PIPELINE DE ETL

A seguir será apresentado o processo completo de **ETL (Extração, Transformação e Carga)** realizado na plataforma **Databricks**, utilizando o dataset **Student Depression Dataset**. Esse processo tem como objetivo

estruturar e preparar os dados para análises posteriores, organizando-os de forma eficiente em um modelo apropriado para um Data Warehouse.

6.1. Extração

Nesta etapa, foi realizada a extração dos dados através da leitura do arquivo CSV contendo o Student Depression Dataset. No código, a extração é feita através do comando abaixo:

The screenshot displays a Databricks notebook titled "MVP_Student_Depression" with a Python editor. The code defines the file location and type, reads the CSV file into a Spark DataFrame, and displays the first few rows. Below the code, the output shows the first 14 rows of the DataFrame.

```
# EXTRAÇÃO
file_location = "/FileStore/tables/Student_Depression_Dataset/Student_Depression_Dataset.csv"
file_type = "csv"

df_raw = spark.read.format(file_type) \
    .option("header", "true") \
    .option("inferSchema", "true") \
    .option("sep", ",") \
    .load(file_location)

display(df_raw)
```

(3) Spark Jobs

df_raw: pyspark.sql.dataframe.DataFrame = [id: integer, Gender: string ... 16 more fields]

	id	Gender	Age	City	Profession	Academic Pressure	Work Pressure
1	2	Male	33	Visakhapatna...	Student		5
2	8	Female	24	Bangalore	Student		2
3	26	Male	31	Srinagar	Student		3
4	30	Female	28	Varanasi	Student		3
5	32	Female	25	Jaipur	Student		4
6	33	Male	29	Pune	Student		2
7	52	Male	30	Thane	Student		3
8	56	Female	30	Chennai	Student		2
9	59	Male	28	Nagpur	Student		3
10	62	Male	31	Nashik	Student		2
11	83	Male	24	Nagpur	Student		3
12	91	Male	33	Vadodara	Student		3
13	94	Male	27	Kalyan	Student		5
14	100	Female	19	Rajkot	Student		2

10,000+ rows | Truncated data | 10.67s runtime | Refreshed 7 hours ago

6.2. Transformação

A transformação é a etapa onde os dados são limpos, ajustados e preparados para análise. No código abaixo, a transformação inclui a normalização de valores, ajuste de categorias e criação de novas tabelas:

MVP_Student_Depression Python Last edit was 13 minutes ago Run all My Cluster Share Publish

File Edit View Run Help

(14) Spark Jobs

- df_cleaned: pyspark.sql.dataframe.DataFrame = [id: integer, Gender: string ... 16 more fields]
- df_fact: pyspark.sql.dataframe.DataFrame

6.2.1. Separação das Tabelas Fato e Dimensão do Dataset de Depressão Estudantil

A seguir, o código utilizado para a remoção de colunas que não devem estar na **tabela de fatos**:

```
# Remover colunas que não devem estar na tabela de fatos
df_fact = df_fact.drop("Age", "Gender", "City")
```

df_fact: pyspark.sql.dataframe.DataFrame

A seguir, o código utilizado para a remoção de colunas que não devem estar na **tabela de dimensões**:

MVP_Student_Depression Python Last edit was 14 minutes ago Run all My Cluster Share Publish

File Edit View Run Help

df_fact: pyspark.sql.dataframe.DataFrame

A seguir, o código utilizado para a remoção de colunas que não devem estar na **tabela de dimensões**:

```
# Remover colunas que não devem estar na tabela de dimensões
df_dimensions = df_dimensions.drop("CGPA", "Sleep_Duration", "Work_Pressure", "Academic_Pressure",
                                   "Study_Satisfaction", "Job_Satisfaction", "Dietary_Habits")
```

df_dimensions: pyspark.sql.dataframe.DataFrame = [ID: integer, Age: integer ... 2 more fields]

6.3. Carga

A carga dos dados é a etapa onde os dados transformados são salvos permanentemente em uma tabela no Databricks.

6.3.1. Tabela de Fatos Ajustada

Abaixo, código para salvar a **tabela de fatos** com as colunas ajustadas:

MVP_Student_Depression Python Last edit was 15 minutes ago Run all My Cluster

File Edit View Run Help

```
# CARGA
df_fact.write.mode("overwrite").format("delta").saveAsTable("Student_Depression_Facts_Dataset")
```

(6) Spark Jobs

Após a execução das etapas de ETL, foi realizada a consulta à **tabela de fatos**, conforme o código abaixo:

MVP_Student_Depression Python

File Edit View Run Help Last edit was 14 minutes ago

Run all My Cluster Share Publish

(2) Spark Jobs

_sqldf: pyspark.sql.dataframe.DataFrame

Table

	id	Profession	Academic_Pressure	Work_Pressure	CGPA	Study_Satisfaction
1	2	Student	5.0	0.0	8.97	2.0
2	8	Student	2.0	0.0	5.9	5.0
3	26	Student	3.0	0.0	7.03	5.0
4	30	Student	3.0	0.0	5.59	2.0
5	32	Student	4.0	0.0	8.13	3.0
6	33	Student	2.0	0.0	5.7	3.0
7	52	Student	3.0	0.0	9.54	4.0
8	56	Student	2.0	0.0	8.04	4.0
9	59	Student	3.0	0.0	9.79	1.0
10	62	Student	2.0	0.0	8.38	3.0
11	83	Student	3.0	0.0	6.1	3.0
12	91	Student	3.0	0.0	7.03	4.0
13	94	Student	5.0	0.0	7.04	1.0
14	100	Student	2.0	0.0	8.52	4.0

6.3.2. Tabela de Dimensão Ajustada

Abaixo, código para salvar a **tabela de dimensão** com as colunas ajustadas:

```

# CARGA
df_dimensions.write.mode("overwrite").format("delta").saveAsTable("Student_Dimensions_Dataset")

```

(6) Spark Jobs

Após a execução das etapas de ETL, foi realizada a consulta à **tabela de dimensão**, conforme o código abaixo:

MVP_Student_Depression Python

File Edit View Run Help Last edit was 16 minutes ago

Run all My Cluster Share Publish

(2) Spark Jobs

_sqldf: pyspark.sql.dataframe.DataFrame = [ID: integer, Age: integer ... 2 more fields]

Table

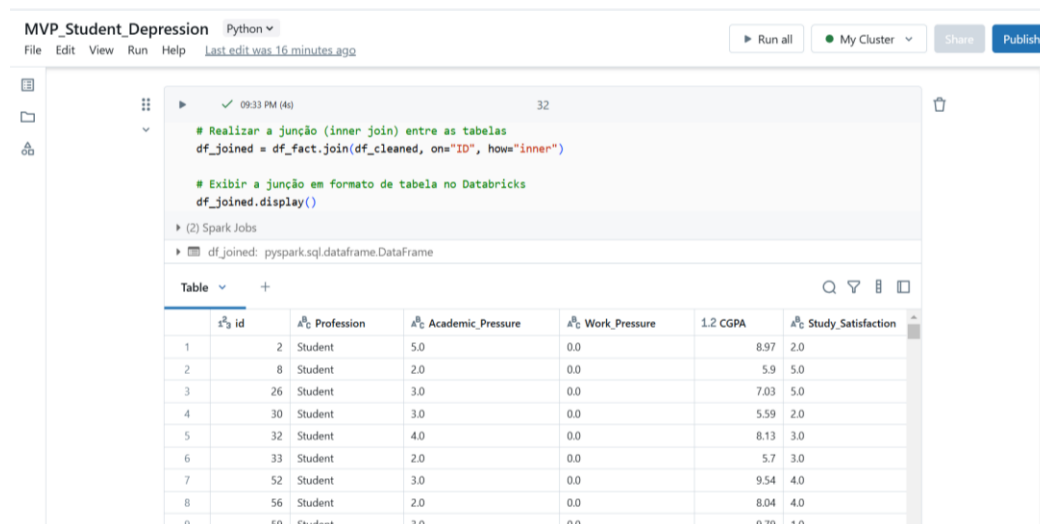
	ID	Age	Gender	City
1	2	33	Male	Visakhapatna...
2	8	24	Female	Bangalore
3	26	31	Male	Srinagar
4	30	28	Female	Varanasi
5	32	25	Female	Jaipur
6	33	29	Male	Pune
7	52	30	Male	Thane
8	56	30	Female	Chennai

6.4. Junção das Tabelas de Fatos e Dimensões com Base na Coluna ID

Este processo realiza a junção (inner join) entre as tabelas de fatos e dimensões utilizando a coluna ID como chave de ligação. Ao fazer isso, garantimos que os dados de ambas as tabelas sejam combinados de maneira eficiente, permitindo uma análise mais completa e precisa. A junção assegura que apenas os registros

correspondentes de ambas as tabelas sejam incluídos no resultado final, mantendo a integridade dos dados e facilitando a análise conjunta das informações de status de depressão com os atributos demográficos e comportamentais dos estudantes.

Segue abaixo o código para realizar a junção das duas tabelas:



The screenshot shows a Databricks notebook interface. At the top, the notebook is titled 'MVP_Student_Depression' and is in 'Python' mode. Below the title bar, there are buttons for 'Run all', 'My Cluster', 'Share', and 'Publish'. The main area contains a code cell with the following Python code:

```
# Realizar a junção (inner join) entre as tabelas
df_joined = df_fact.join(df_cleaned, on="ID", how="inner")

# Exibir a junção em formato de tabela no Databricks
df_joined.display()
```

Below the code cell, the output is displayed as a table. The table has 7 columns: 'id', 'Profession', 'Academic_Pressure', 'Work_Pressure', 'CGPA', and 'Study_Satisfaction'. The table contains 9 rows of data.

id	Profession	Academic_Pressure	Work_Pressure	CGPA	Study_Satisfaction
1	Student	5.0	0.0	8.97	2.0
2	Student	2.0	0.0	5.9	5.0
3	Student	3.0	0.0	7.03	5.0
4	Student	3.0	0.0	5.59	2.0
5	Student	4.0	0.0	8.13	3.0
6	Student	2.0	0.0	5.7	3.0
7	Student	3.0	0.0	9.54	4.0
8	Student	2.0	0.0	8.04	4.0
9	Student	3.0	0.0	9.79	1.0

7. ANÁLISE

Os dados foram preparados para análise detalhada, onde serão explorados os fatores que podem influenciar o bem-estar e a saúde mental dos estudantes.

Neste processo, examinaremos as correlações entre variáveis como desempenho acadêmico, satisfação com os estudos e hábitos alimentares, buscando identificar possíveis relações com o status de depressão entre os participantes.

7.1. Identificar fatores críticos que contribuem para a depressão estudantil

Este item tem como objetivo identificar os principais fatores que contribuem para o desenvolvimento da depressão entre os estudantes. Através da análise de dados relacionados ao desempenho acadêmico, pressão no trabalho, hábitos alimentares e outros fatores, buscamos entender as correlações e identificar as condições que podem agravar a saúde mental dos alunos, oferecendo insights importantes para a implementação de medidas de apoio e prevenção.

7.1.1. Depressão e Desempenho Acadêmico

Abaixo, código com o volume de estudantes com depressão e seu desempenho acadêmico:

MVP_Student_Depression Python

File Edit View Run Help Last edit was 17 minutes ago

Run all My Cluster Share Publish

Abaixo, código com o volume de estudantes com depressão e seu desempenho acadêmico:

```
# Definindo os critérios para alto e baixo desempenho acadêmico
high_performance_threshold = 8.0
low_performance_threshold = 5.0

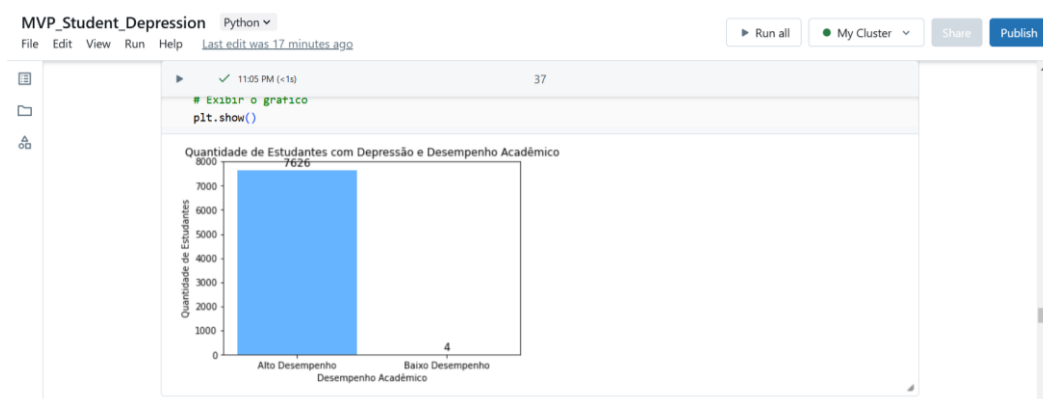
# Contar o número de alunos com alto desempenho acadêmico (CGPA > 8.0) e depressão
high_performance_with_depression = df_fact.filter((df_fact['CGPA'] > high_performance_threshold) & (df_fact['Depression_Status'] == 1)).count()

# Contar o número de alunos com baixo desempenho acadêmico (CGPA <= 5.0) e depressão
low_performance_with_depression = df_fact.filter((df_fact['CGPA'] <= low_performance_threshold) & (df_fact['Depression_Status'] == 1)).count()

print(f"Quantidade de estudantes com alto desempenho acadêmico (CGPA > 8.0) e depressão: {high_performance_with_depression}")
print(f"Quantidade de estudantes com baixo desempenho acadêmico (CGPA <= 5.0) e depressão: {low_performance_with_depression}")
```

(4) Spark Jobs

Quantidade de estudantes com alto desempenho acadêmico (CGPA > 8.0) e depressão: 7626
Quantidade de estudantes com baixo desempenho acadêmico (CGPA <= 5.0) e depressão: 4



Com base nos dados apresentados, podemos observar:

Alto Desempenho Acadêmico (CGPA > 8.0) e Depressão: A quantidade de estudantes nesta categoria foi de 7.626. Um número considerável podendo indicar que, apesar de apresentarem alto desempenho acadêmico, uma parte significativa desses estudantes enfrenta desafios relacionados à depressão.

Baixo Desempenho Acadêmico (CGPA <= 5.0) e Depressão: Apenas 4 estudantes se encaixaram nesta categoria. Esse número muito pequeno pode ser devido à natureza do filtro de desempenho acadêmico baixo (CGPA <= 5.0), que restringe ainda mais a amostra, e ao fato de a depressão ser um fator relevante em qualquer faixa de desempenho, mas mais difícil de se observar com baixos resultados acadêmicos.

A relação entre o desempenho acadêmico e a depressão pode ser mais complexa do que apenas um simples vínculo entre baixo desempenho e problemas psicológicos. No caso de "Alto Desempenho Acadêmico e Depressão", pode-se argumentar que muitos desses estudantes estão lidando com uma pressão extrema para manter boas notas ou expectativas altas, o que pode levar a um estresse significativo e, em alguns casos, à depressão.

7.1.2. Depressão e Pressão Acadêmica

Abaixo, código com o volume de estudantes **com depressão e que sofrem pressão acadêmica**:

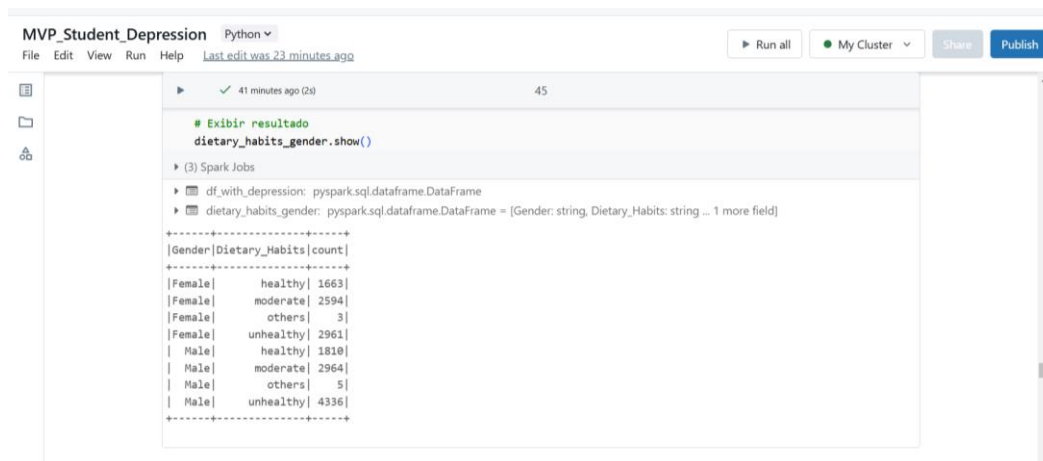


O gráfico apresentado ilustra que o número de 932 estudantes com depressão está associado àqueles que estão enfrentando pressão acadêmica.

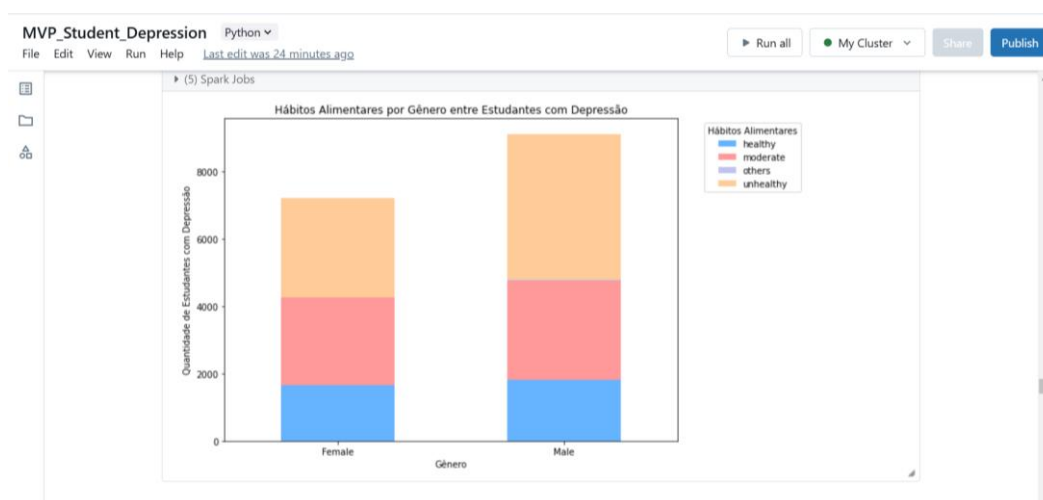
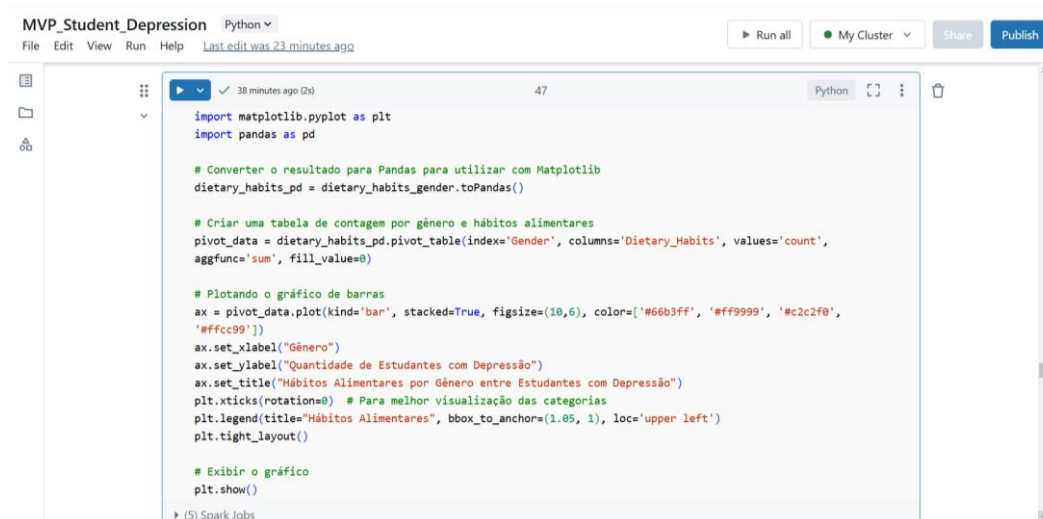
7.1.3. Depressão e Hábitos Alimentares

Abaixo, código com o volume de estudantes **com depressão e seus hábitos alimentares**:





Abaixo, gráfico com o volume de estudantes com depressão e seus hábitos alimentares:



A análise dos dados mostra a distribuição dos hábitos alimentares entre homens e mulheres com base nas categorias de alimentação (saudável, moderada, outros e não saudável). A partir dos números apresentados, podemos observar:

Mulheres:

- **Saudável:** 1663 mulheres têm hábitos alimentares considerados saudáveis.
- **Moderada:** 2594 mulheres apresentam hábitos alimentares moderados.
- **Outros:** 3 mulheres têm hábitos alimentares classificados como "outros".
- **Não saudável:** 2961 mulheres têm hábitos alimentares não saudáveis, o que representa a maior quantidade entre as categorias.

Homens:

- **Saudável:** 1810 homens têm hábitos alimentares saudáveis.
- **Moderada:** 2964 homens apresentam hábitos alimentares moderados.
- **Outros:** 5 homens têm hábitos alimentares classificados como "outros".
- **Não saudável:** 4336 homens têm hábitos alimentares não saudáveis, sendo a maior quantidade entre as categorias também para os homens.

A categoria **não saudável** é a mais predominante tanto para homens quanto para mulheres, o que pode ser um indicativo relevante de comportamentos alimentares preocupantes entre os estudantes.

A quantidade de mulheres com **hábitos saudáveis** é inferior à de homens, mas o número de mulheres com **hábitos não saudáveis** está bem próximo ao de homens, o que pode indicar que ambos os gêneros apresentam padrões alimentares semelhantes nesse aspecto.

7.2. Qual o volume de estudantes com depressão?

Abaixo, código com o volume de estudantes com depressão:

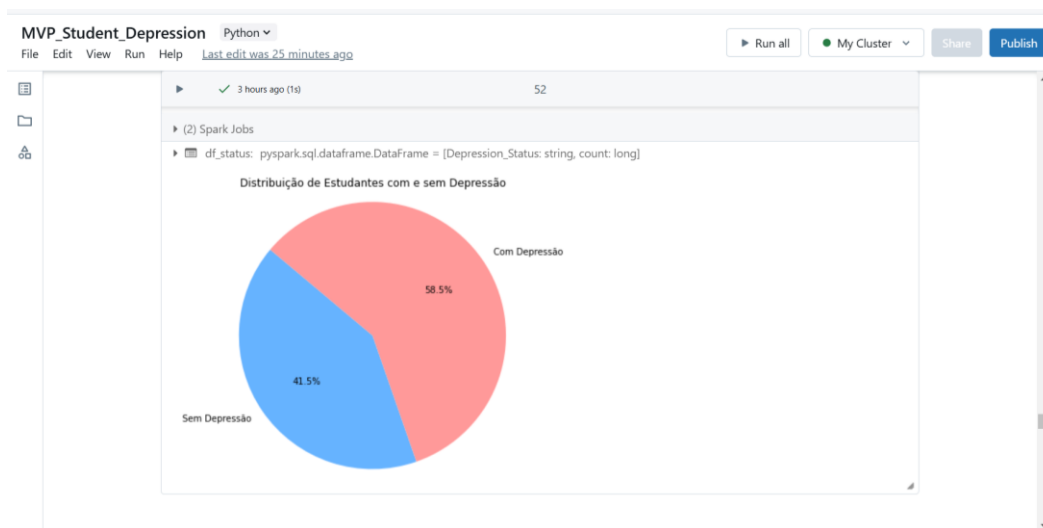


```
MVP_Student_Depression Python
File Edit View Run Help Last edit was 24 minutes ago
Run all My Cluster Share Publish

# Contagem do número de estudantes com depressão
df_fact.groupby("Depression_Status").count().show()

(2) Spark Jobs
+-----+
|Depression_Status|count|
+-----+
|0|11565|
|1|16336|
+-----+
```

Abaixo, gráfico contendo a distribuição de estudantes **com** e **sem** depressão:



O resultado mostra a contagem de estudantes **com** e **sem** depressão no conjunto de dados:

Sem depressão (0): 11.565 estudantes

Com depressão (1): 16.336 estudantes

Ou seja, aproximadamente **58,5%** dos estudantes apresentam sinais de depressão, enquanto **41,5%** não apresentam. Esse dado já traz um alerta importante sobre a necessidade de ações voltadas à saúde mental no ambiente estudantil.

7.3. Qual é o perfil demográfico dos usuários, maior volume entre meninos ou meninas?

Abaixo, código com a contagem com maior número de casos de depressão entre os estudantes, segmentado por gênero:

MVP_Student_Depression Python

File Edit View Run Help Last edit was 26 minutes ago

Run all My Cluster Share Publish

2 hours ago (1s) 55

```
# Realizar a junção das tabelas de fatos e dimensões com base na coluna ID
df_with_dimensions = df_fact.alias("fact").join(df_cleaned.alias("dim"), on="ID", how="inner")

# Filtrar as pessoas com depressão (Depression_Status == 1)
df_depressed = df_with_dimensions.filter(F.col("fact.Depression_Status") == 1)

# Contar pessoas com depressão por gênero
df_depressed_gender_count = df_depressed.groupBy("dim.Gender").count()

# Exibir o resultado
df_depressed_gender_count.show()
```

(3) Spark Jobs

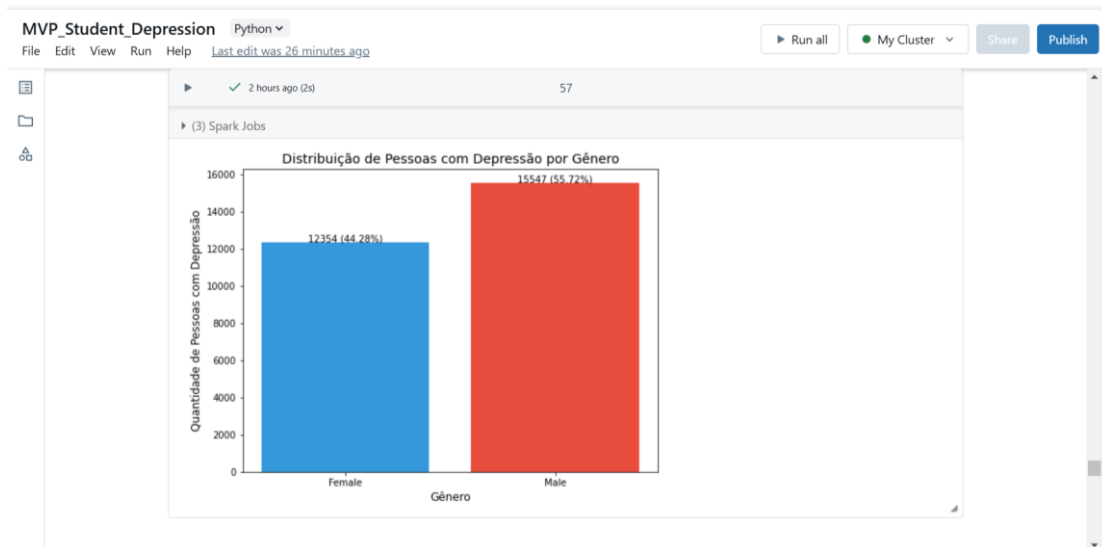
df_depressed: pyspark.sql.dataframe.DataFrame

df_depressed_gender_count: pyspark.sql.dataframe.DataFrame = [Gender: string, count: long]

df_with_dimensions: pyspark.sql.dataframe.DataFrame

```
+-----+
|Gender|count|
+-----+
|Female| 7221|
|Male| 9115|
+-----+
```


Abaixo, gráfico contendo a distribuição de estudantes dos gêneros feminino e masculino com depressão:



A análise dos dados apresentados, onde temos a contagem de pessoas com depressão por gênero, mostra o seguinte:

Mulheres (Female): 7.221 casos de depressão.

Homens (Male): 9.115 casos de depressão.

A análise dos percentuais de 55,72% para homens e 44,28% para mulheres revela uma diferença significativa na proporção de depressão entre os gêneros na amostra estudada. Observa-se que o número de homens com depressão é maior que o número de mulheres, indicando que, neste conjunto de dados específico, os homens estão mais afetados por depressão do que as mulheres.

7.4. Qual a faixa etária dos homens e mulheres com depressão?

Abaixo, código com a contagem de idade com maior número de casos de depressão entre os estudantes, segmentado por gênero:

MVP_Student_Depression

Python

File Edit View Run Help

Last edit was 27 minutes ago

Run all

My Cluster

Share

Publish

2 hours ago (2s)

60

```

from pyspark.sql import functions as F
from pyspark.sql.window import Window

# Criar o DataFrame unificado com a contagem
df_joined = spark.sql("""
    SELECT d.Gender, d.Age, COUNT(*) as count
    FROM Student_Depression_Facts_Dataset f
    INNER JOIN Student_Dimensions_Dataset d
    ON f.ID = d.ID
    WHERE f.Depression_Status = 1
    GROUP BY d.Gender, d.Age
""")

# Definir janela particionada por gênero ordenando pela contagem decrescente
window_spec = Window.partitionBy("Gender").orderBy(F.desc("count"))

# Adicionar a coluna de rank e filtrar apenas a primeira posição de cada gênero
df_top_age_by_gender = df_joined.withColumn("rank", F.row_number().over(window_spec)) \
    .filter(F.col("rank") == 1) \
    .drop("rank")

# Exibir resultado
df_top_age_by_gender.show()

```

(4) Spark Jobs

df_joined: pyspark.sql.dataframe.DataFrame = [Gender: string, Age: integer ... 1 more field]

df_top_age_by_gender: pyspark.sql.dataframe.DataFrame = [Gender: string, Age: integer ... 1 more field]

df_joined: pyspark.sql.dataframe.DataFrame = [Gender: string, Age: integer ... 1 more field]

df_top_age_by_gender: pyspark.sql.dataframe.DataFrame = [Gender: string, Age: integer ... 1 more field]

```

+-----+-----+
|Gender|Age|count|
+-----+-----+
|Female| 20|   766|
|Male|  24|   826|
+-----+-----+

```

O gráfico a seguir apresenta a idade com maior número de casos de depressão entre os estudantes, segmentado por gênero:

MVP_Student_Depression

Python

File Edit View Run Help

Last edit was 28 minutes ago

Run all

My Cluster

Share

Publish

2 hours ago (2s)

62

```

# Converter para Pandas
pdf_top_age = df_top_age_by_gender.toPandas()

# Criar gráfico de barras
plt.figure(figsize=(8,5))
bars = plt.bar(pdf_top_age['Gender'], pdf_top_age['count'], color=['#66b3ff', '#ff9999'])

# Adicionar os rótulos de idade acima das barras
for bar, age in zip(bars, pdf_top_age['Age']):
    height = bar.get_height()
    plt.text(bar.get_x() + bar.get_width()/2, height + 100, f'Idade: {age}', ha='center', fontsize=10, color='black')

# Ajustes de visualização
plt.title('Idade com Maior Número de Casos de Depressão por Gênero')
plt.ylabel('Quantidade de Pessoas com Depressão')
plt.xlabel('Gênero')
plt.ylim(0, max(pdf_top_age['count']) * 1.2)
plt.grid(axis='y', linestyle='--', alpha=0.7)
plt.tight_layout()

# Exibir gráfico
plt.show()

```

(4) Spark Jobs

Idade com Maior Número de Casos de Depressão por Gênero

Idade: 24



A análise dos dados apresentados mostra a distribuição de estudantes com depressão por gênero e faixa etária, com base no seguinte:

Gênero:

- **Mulheres:** 766 estudantes com 20 anos de idade.
- **Homens:** 826 estudantes com 24 anos de idade.

Idade:

A maior volumetria de estudantes com depressão **entre as mulheres ocorre com 20 anos**.

A maior volumetria de estudantes com depressão **entre os homens ocorre com 24 anos**.

Gênero:

Embora o número de estudantes com depressão seja maior entre os homens (826 contra 766), é importante observar que a faixa etária é diferente, com homens sendo mais afetados aos 24 anos e mulheres aos 20 anos.

8. CONCLUSÃO

A análise dos dados revelou que um número considerável de estudantes com depressão está associado a alto desempenho acadêmico, indicando que a pressão para alcançar bons resultados pode ser um fator significativo de risco para problemas de saúde mental. A pressão acadêmica se destacou como um fator crucial, impactando 932 estudantes com depressão, o que reforça a necessidade de estratégias de apoio psicológico para lidar com as exigências do ambiente acadêmico.

Outro ponto importante foi a observação de hábitos alimentares predominantemente não saudáveis entre os estudantes, o que pode estar relacionado ao agravamento da saúde mental. Embora tanto homens quanto mulheres apresentem padrões semelhantes, as mulheres têm uma ligeira prevalência de hábitos alimentares mais saudáveis. Esses dados sugerem que

intervenções voltadas para melhorar a saúde mental, como programas de apoio psicológico, promoção de bons hábitos alimentares e qualidade do sono, são essenciais para ajudar os estudantes a lidarem com a depressão e outros desafios associados à vida acadêmica.