Pontifícia Universidade Católica do Rio de Janeiro

Pós-graduação Lato Sensu em Ciência de Dados e Analytics

Sprint: Engenharia de Dados (40530010057_20250_01)

Aluna: Nahanni Taynah Jácome Rodrigues

Matrícula: 4052025000359

Documentação MVP Engenharia de Dados

Objetivo

O objetivo do MVP é analisar um conjunto de dados sobre informações de candidatos que buscam emprego na área de ciência de dados, função que vem sendo bastante explorada atualmente, tendo em vista as mudanças que estão ocorrendo nas empresas ao redor do mundo, que é a busca por agregar valor aos dados existentes nas mesmas. O profissional de ciência de dados entra, dessa forma, sendo umas das peças chaves para a implementação do uso inteligente desses dados para os gestores, de modo a buscar melhores resultados.

Algumas informações pertinentes podem ser levadas em conta, a partir de um conjunto de dados disponível no kaggle, por meio do endereço: https://www.kaggle.com/datasets/sachinkumar62/datascience-job-data, e seu uso é permitido sob licença MIT conforme os termos do link: https://www.mit.edu/~amini/LICENSE.md.

Tem-se alguns dados dos candidatos, como: dados demográficos, histórico educacional, experiência de trabalho e horas de treinamento. A variável de destino (*target*) indica se um candidato está procurando um novo emprego (1) ou não (0). A tabela 1 possui a descrição de todos os atributos presentes no *dataset*.

Tabela 1. Atributos e descrições do dataset

Atributo	Descrição
enrolle_id	Identificador exclusivo para cada candidato
city	Código da cidade onde o candidato está localizado.

city_development_index	Pontuação do índice de desenvolvimento da cidade (escala de 0 a 1).
gender	Gênero do candidato (masculino, feminino, outro).
relevent_experience	Se o candidato tem experiência relevante na área.
enrolled_university	Status de matrícula do candidato (período integral, meio período ou não matriculado).
education_level	Nível de educação do candidato.
major_discipline	Área de estudo (STEM, negócios, etc.)
experience	Anos de experiência profissional
company_size	Tamanho da última empresa em que o candidato trabalhou.
company_type	Tipo de empresa (por exemplo, Pvt Ltd, Startup, etc.)
training_hours	Horas gastas em treinamento.
target	Indicador binário (1 = procurando um novo emprego, 0 = não procurando).

A partir dessas informações, pode-se estabelecer alguns questionamentos que faz com que nós, estudantes da pós graduação da PUC-Rio em Ciência de dados e Analitycs, possamos nos orientar em relação a esse cenário. Quatro avaliações diferentes podem ser levadas em consideração e as perguntas para cada cenário estão descritas a seguir:

1. Perfil dos profissionais:

- Qual a distribuição de gênero dos candidatos?
- Qual o nível de formação mais comum entre os profissionais de ciência de dados?
- Qual a experiência média dos candidatos?
- Quantas pessoas já têm experiência relevante na área?

2. Fatores que Impactam a Contratação:

- O nível de educação influencia na intenção de trocar de emprego?
- Candidatos que fizeram mais horas de treinamento têm mais chances de mudar de emprego?
- Qual o impacto do tamanho da empresa na retenção de talentos?
- Empresas privadas (Pvt Ltd) perdem mais funcionários do que startups?

3. Localização e Desenvolvimento do Mercado:

- Em quais cidades há maior concentração de profissionais de ciência de dados?
- O índice de desenvolvimento da cidade (city_development_index) tem impacto na retenção de talentos?

4. Análises sobre Empresas

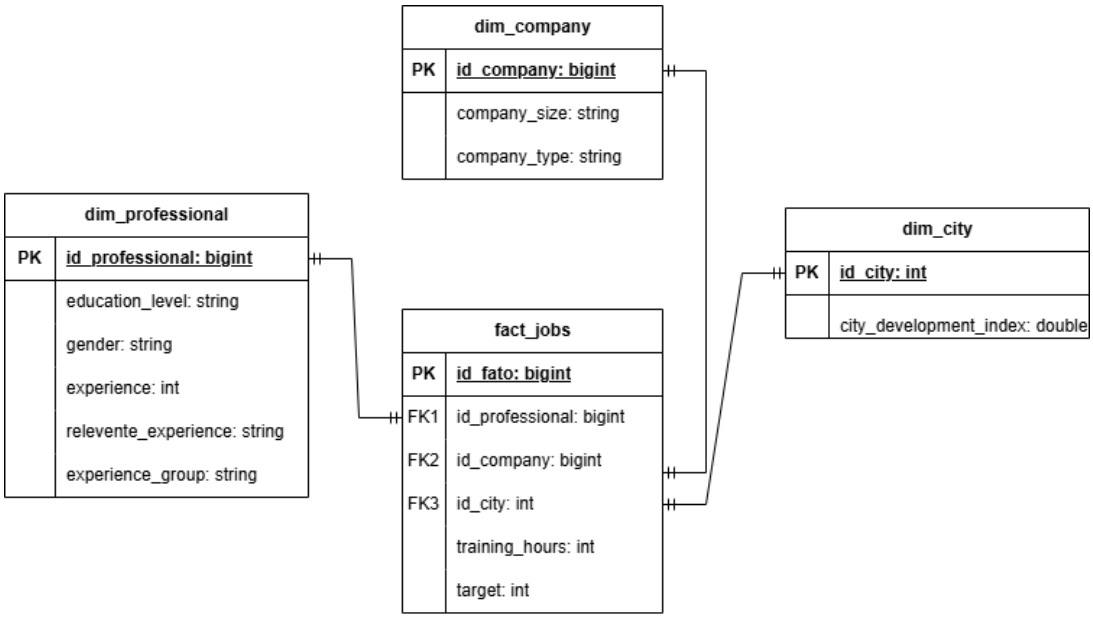
- Qual o porte das empresas que mais contratam profissionais de ciência de dados?

- Startups contratam mais profissionais juniores ou seniores?
- Empresas grandes (5000+ funcionários) investem mais em treinamento do que empresas pequenas?

Essas perguntas ajudam a entender o cenário do qual se trata esse banco de dados, onde os candidatos e as cidades onde os mesmos se encontram são dados anônimos, porém pode-se compreender quais fatores influenciam na retenção (*target* = 0) ou na busca por emprego (*target* = 1) dos candidatos. Com a escolha desse banco de dados disponível no Kaggle, os dados foram baixados localmente no dia 12/02/2025, sendo essa a etapa de coleta dos dados, para posteriormente serem inseridos em uma plataforma de armazenamento e analise de dados, a *Databricks Community Edition*. Apesar de ser uma versão de uso gratuito e possuir limitações, é possível construir uma pipeline de dados nessa plataforma e obter *insights* a partir dos dados estudados.

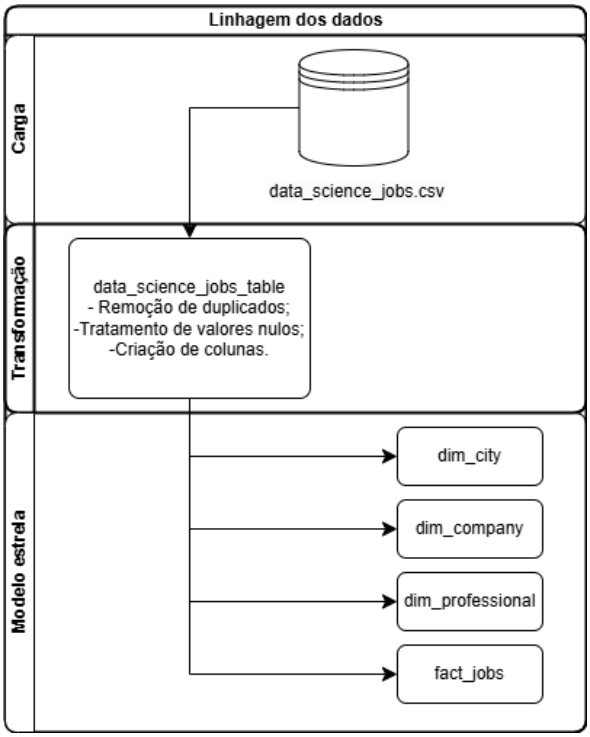
O *dataset* utilizado nesse projeto é em formato CSV, e a modelagem escolhida foi o modelo estrela, como apresentado na Figura 1, onde a tabela original foi dividida da seguinte forma: três tabelas dimensão (*dim_city, dim_company e dim_professional*) e uma tabela fato (*fact_jobs*) que apresenta as chaves estrangeiras das tabelas dimensão e as variáveis numéricas '*training_hours*' e '*target*'.

Figura 1. Modelagem de dados



A linhagem dos dados está apresentada na Figura 2 abaixo.

Figura 2. Linhagem dos dados



Catálogo de dados

Abaixo apresenta-se os catálogos de dados para cada tabela criada no modelo estrela, como citado anteriormente:

- **dim_city:**

Nome da coluna	Tipo de Dado	Descrição	Unidade	Valor mínimo	Valor máximo	Categorias
Id_city	INT	Código da cidade onde o candidato está localizado.	N/A	N/A	N/A	N/A
city_development_index	DOUBLE	Pontuação do índice de desenvolvimento da cidade (escala de 0 a 1).	N/A	0	1	N/A

- **dim_company:**

Nome da coluna	Tipo de Dado	Descrição	Unidade	Valor mínimo	Valor máximo	Categorias
id_company	BIGINT	Identificador único criado para cada empresa	N/A	N/A	N/A	N/A
company_size	STRING	Tamanho da última empresa em que o candidato trabalhou.	N/A	N/A	N/A	'<10', '10-49', '50-99', '100-500', '500-999', ‘1000-4999', '5000+'.

company_type	STRING	Tipo de empresa (por exemplo, Pvt Ltd, Startup, etc.).	N/A	N/A	N/A	'Pvt Ltd', 'Funded Startup', 'Early Stage Startup', 'NGO', 'Public Sector', 'Other'

- **dim_professional:**

Nome da coluna	Tipo de Dado	Descrição	Unidade	Valor mínimo	Valor máximo	Categorias
id_professional	INT	Identificador exclusivo para cada candidato	N/A	N/A	N/A	N/A
education_level	STRING	Nível de educação do candidato.	N/A	N/A	N/A	'Primary School', 'High School', 'Graduate', 'Masters', 'Phd'
gender	STRING	Gênero do candidato	N/A	N/A	N/A	'Male', 'Female', 'Other'
experience	INT	Anos de experiência profissional	Years	N/A	N/A	N/A
relevent_experience	STRING	Se o candidato tem experiência relevante.	N/A	N/A	N/A	'Has relevant experience', 'No relevant experience'
experience_group	STRING	Classificação do candidato de acordo com os anos de experiencia profissional	N/A	N/A	N/A	'Junior', 'Pleno', 'Senior'

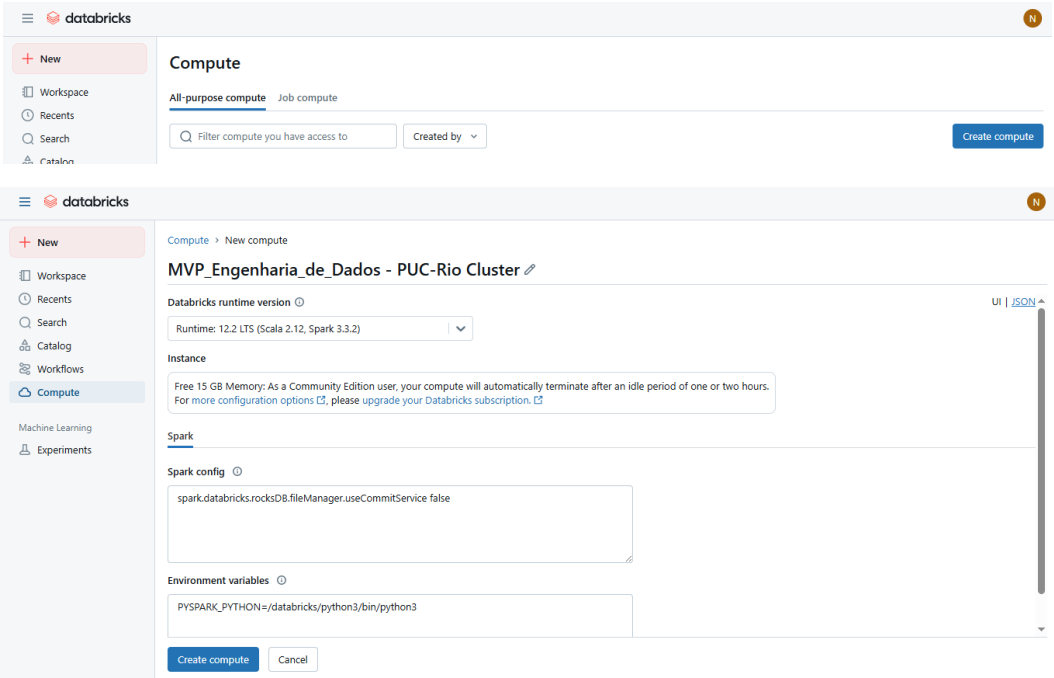
- **fact_jobs:**

Nome da coluna	Tipo de Dado	Descrição	Unidade	Valor mínimo	Valor máximo	Categorias
id_fato	BIGINT	Identificador único criado para tabela fato	N/A	N/A	N/A	N/A
id_professional	BIGINT	Chave estrangeira da tabela dim_professional	N/A	N/A	N/A	N/A
id_company	BIGINT	Chave estrangeira da tabela dim_company	N/A	N/A	N/A	N/A
id_city	STRING	Chave estrangeira da tabela dim_city	N/A	N/A	N/A	N/A
training_hours	INT	Horas gastas em treinamento.	Horas	N/A	N/A	N/A
target	INT	Indicador binário (1 = procurando um novo emprego, 0 = não procurando).	N/A	N/A	N/A	0 (Não), 1 (Sim)

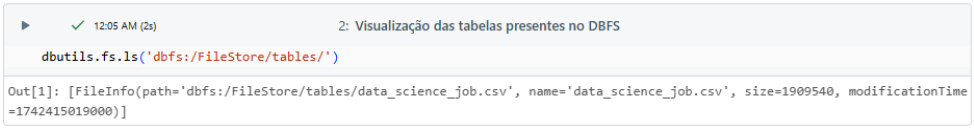
Carga e transformação dos dados

Os dados coletados foram carregados no Databricks Community Edition, seguindo os passos abaixo:

1. Acesso ao Databricks Community Edition;
2. Criação de uma conta de acesso a plataforma;
3. Na aba ‘Compute’, clicar em ‘criar um novo cluster’;

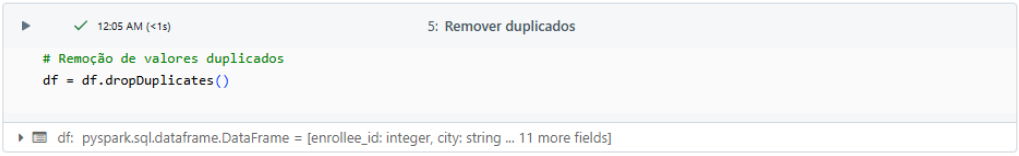


4. Após a inicialização do cluster, clicar na opção ‘New’, em seguida ‘Add or upload data’;
5. Adicionar o arquivo csv ‘data_science_jobs.csv’ baixado localmente no Databricks File System (DBFS);



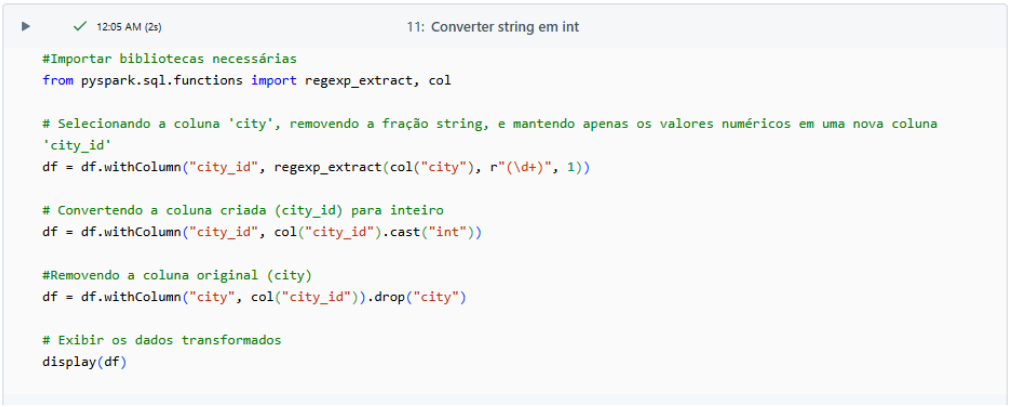
6. Clicar em ‘Create Table With UI’;
7. Selecionar o Cluster Criado na 3° etapa;
8. Escolha da linguagem Python para construção do Notebook;
9. Leitura do arquivo CSV na linguagem Python com as configurações necessárias para esse tipo de arquivo;
10. Nomear o Notebook para MVP Engenharia de Dados – PUC_Rio.

O conjunto de dados tem ao todo 19158, nomeado por ‘df’, dos quais o atribulo *enrrole_id* é a chave identificadora de cada candidato, a qual não se repete. As demais colunas foram avaliadas também. Inicialmente foram eliminados possível valores duplicados com o comando:

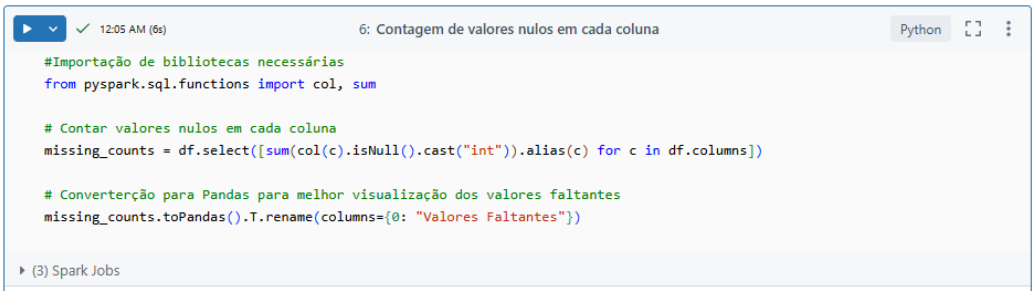


Após essa etapa, foi criado um novo atributo derivado do atributo ‘*experience*’, que foi o ‘*experience_group*’, onde os candidatos foram classificados como Junior, Pleno e Senior de acordo com os anos que possuíam de experiencia, de acordo com os comandos:

Outra transformação que precisou ser feita foi no atributo ‘*city*’, onde o mesmo representa um identificador para as cidades de onde são os candidatos, porém o atributo é do tipo *string*. Desse modo, foi feito uma extração apenas da parte numérica dessa coluna, para posteriormente a mesma ser usada como id da tabela ‘*dim_city*’. Os comandos usados foram:



Foi feito posteriormente uma contagem dos valores nulos presentes em cada atributo, a partir dos comandos:



Com essa consulta, pode-se obter a seguinte saída:

	Valores Faltantes
enrollee_id	0
city	0
city_development_index	479
gender	4508
relevent_experience	0
enrolled_university	386
education_level	460
major_discipline	2813
experience	65
company_size	5938
company_type	6140
training_hours	766
target	0

Abaixo serão descritas as formas de tratamento dos valores nulos presente em cada coluna:

Coluna	Tratamento realizado
enrollee_id	Nenhum tratamento necessário, pois não há valores nulos.
city	Nenhum tratamento necessário, pois não há valores nulos.
city_development_index	Substituir valores nulos pela mediana, pois é um índice numérico
gender	Preencher valores nulos com " Not informed".
relevent_experience	Nenhum tratamento necessário, pois não há valores nulos.
enrolled_university	Preencher valores nulos com " Not informed".
education_level	Preencher valores nulos com a moda.
major_discipline	Preencher valores nulos com " Not informed".
experience	Substituir valores nulos pela mediana, pois é um dado numérico relacionado à experiência.
company_size	Preencher valores nulos com " Not informed".
company_type	Preencher valores nulos com " Not informed".
training_hours	Substituir valores nulos pela mediana, pois é um dado numérico contínuo.
target	Nenhum tratamento necessário, pois não há valores nulos.

Análise: Qualidade dos dados

Os dados da tabela 'data_science_jobs.csv' representam candidatos da área de Ciência de dados de diferentes cidades, com algumas informações a respeito de nível de educação, anos de experiencia na área, se possui experiencia relevante, horas de treinamento, se o candidato está em busca de emprego ou não (*target*). Trata-se de um panorama geral dos candidatos, sem especificar quais seriam as cidades usadas na coleta dos dados, também não possui possibilidade de uma análise temporal de quando os candidatos forneceram essas informações.

Avaliando cada atributo de forma individual, a respeito da qualidade, tem-se:

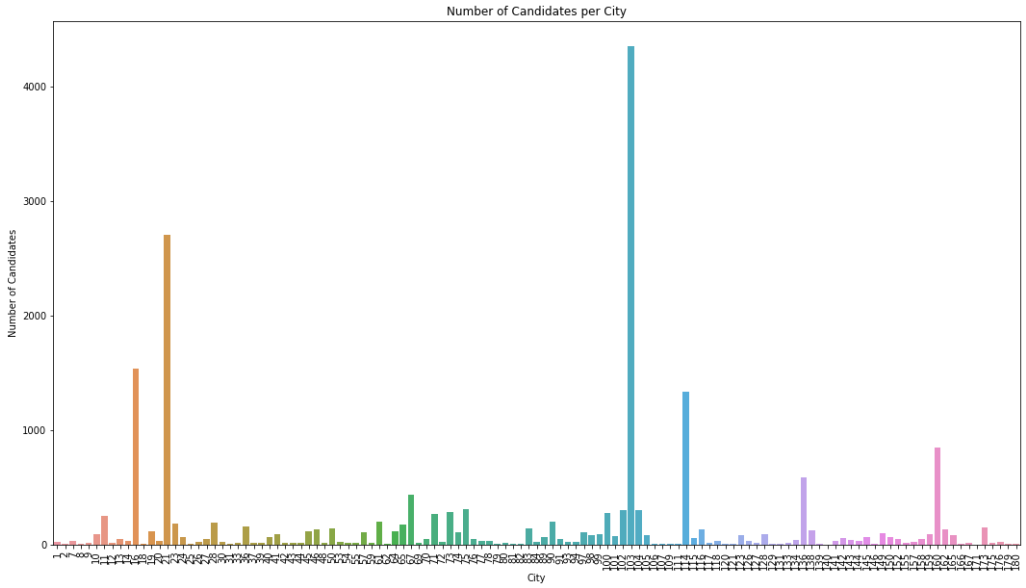
Tabela 2. Análise da qualidade dos dados do *dataset*

Coluna	Análise da qualidade
enrollee_id	Possui 19158 valores distintos para cada candidato, não possuindo inconsistências no mesmo
city	Possui 123 valores distintos para a cidade onde os candidatos se encontram, possuindo apenas a correção necessária para eliminar a parte 'strig' e deixar em formato 'int', que foi descrito anteriormente.
city_development_index	São os índices de desenvolvimento de cada cidade, podendo variar entre 0 e 1. A inconsistência encontrada foi valores diferentes para a mesma cidade, nesse caso foi feito a média entre os valores diferentes e esse valor foi inserido para cada cidade apresentar apenas um índice de desenvolvimento e não apresentar informações duplicadas.
gender	Valores categóricos que representam o gênero dos candidatos, onde os da tabela original eram 'Male', 'Female', 'Other', e adicionado 'Not informed' para os valores nulos. Não apresenta inconsistências.
relevent_experience	São valores categóricos que informam se o candidato possui experiencia relevante na área ou não. Não apresenta inconsistências.
enrolled_university	Informa qual o status de matricula do candidato na universidade, onde possuía alguns valores nulos que foram tratados como descrito anteriormente. Porém essa informação não será utilizada nas análises feitas posteriormente.
education_level	Indica o nível de educação dos candidatos. Não apresenta inconsistências.
major_discipline	Indica a área de estudos do candidato. Não apresenta inconsistências.
experience	Representa os anos de experiencia de cada candidato na área de Ciência de dados. Não apresenta inconsistências.
company_size	Indica o tamanho da companhia onde o candidato trabalha. Não apresenta inconsistências.
company_type	Indica o tipo da companhia onde o candidato trabalha. Não apresenta inconsistências.
training_hours	Indica as horas de treino de cada candidato. Não apresenta inconsistências.
target	Representa, em números binários (0 ou 1), se o candidato está à procura de emprego. Não apresenta inconsistências.

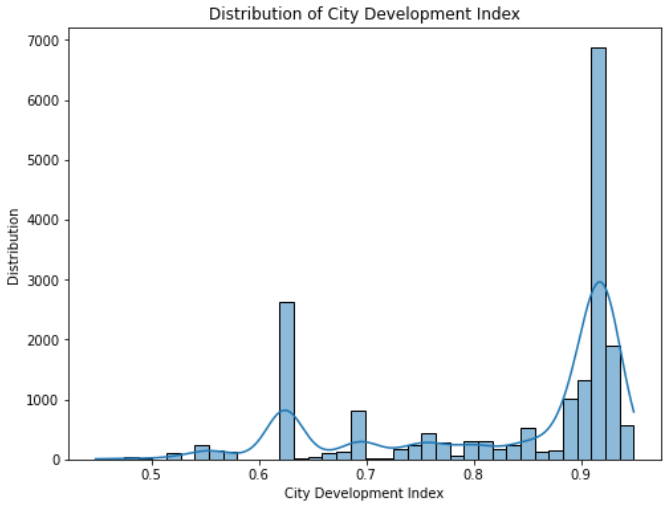
Análise: Solução do problema

Algumas análises gráficas foram feitas inicialmente para avaliar a distribuição de cada variável do conjunto de dados estudado. A biblioteca Matplotlib da linguagem Python possui ferramentas disponíveis para esse estudo. Os gráficos tornam os dados mais acessíveis e compreensíveis para pessoas que não são especialistas em análise de dados, como executivos ou gestores, ajudando a comparar diferentes grupos de dados de forma visual, facilitando a identificação das diferenças e apoiando a tomada de decisão.

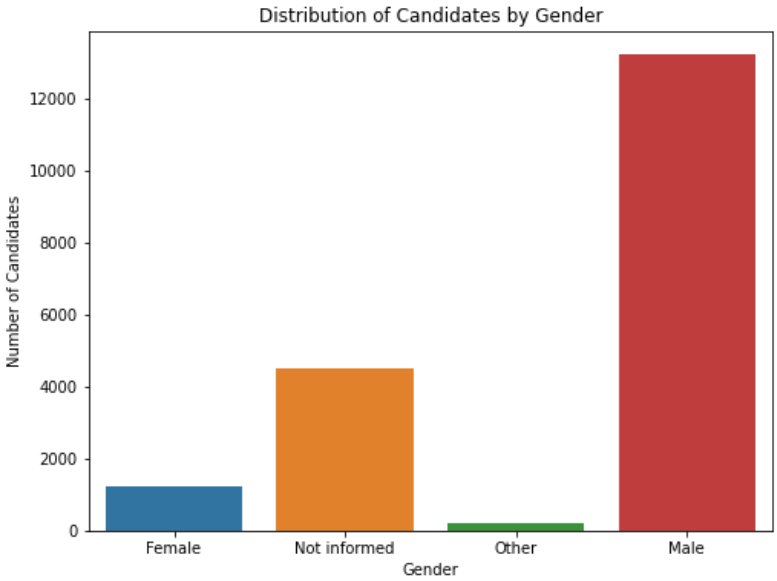
1. Distribuição dos candidatos por cidade



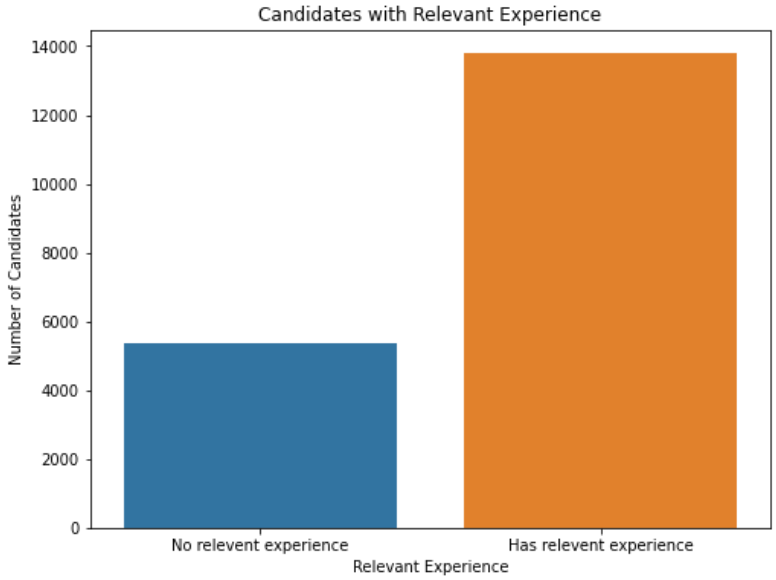
2. Distribuição de frequência (histograma) do índice de desenvolvimento das cidades



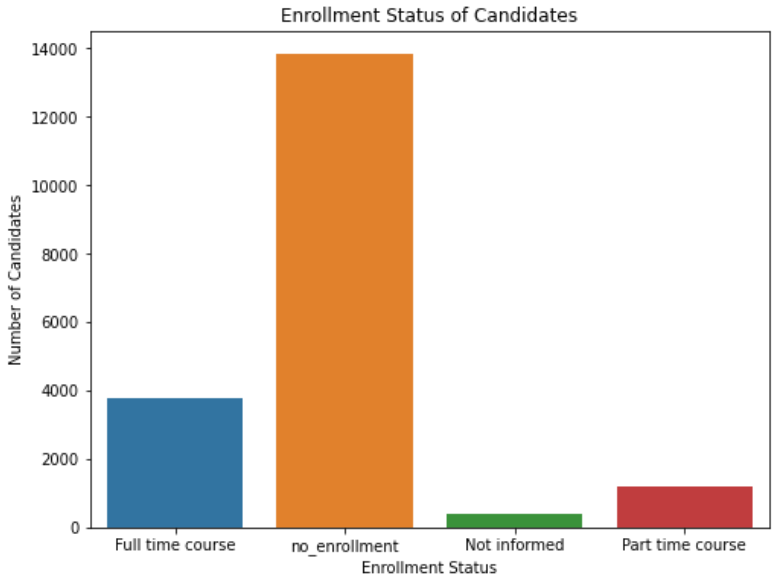
3. Distribuição dos candidatos por gênero



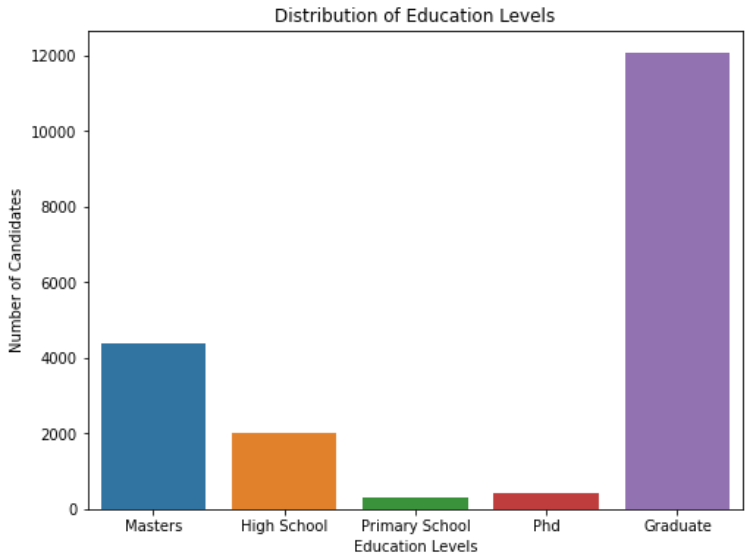
4. Distribuição dos candidatos por experiencia relativa na área de Ciência de dados



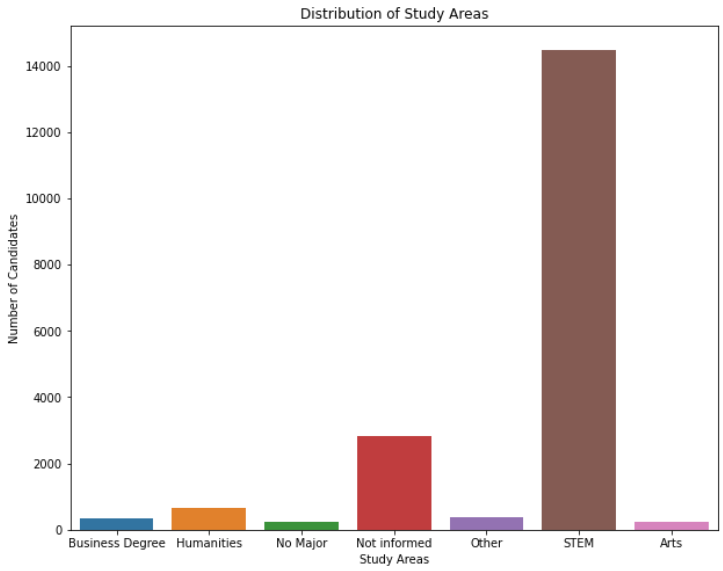
5. Distribuição dos candidatos por Status de matrícula na universidade



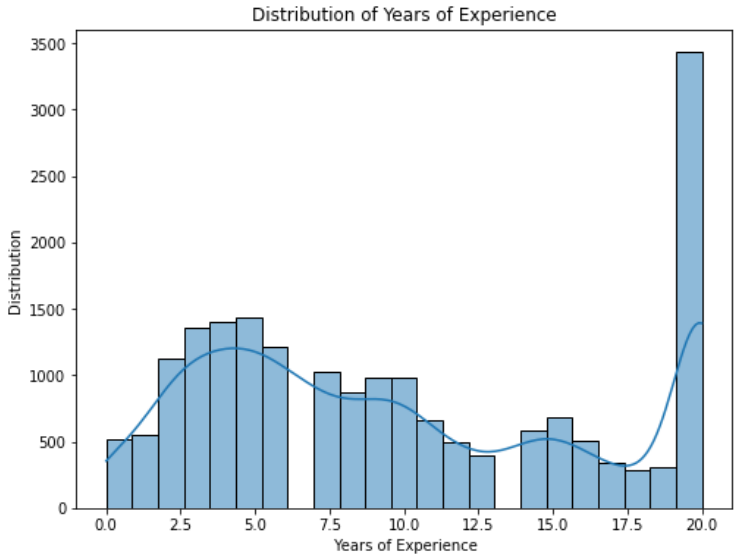
6. Distribuição dos candidatos pelo nível de educação



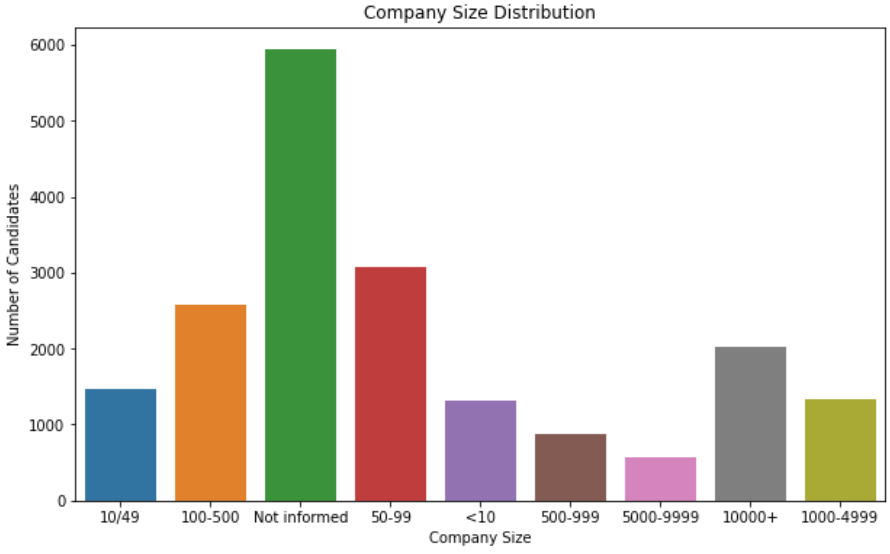
7. Distribuição dos candidatos pela área de estudos



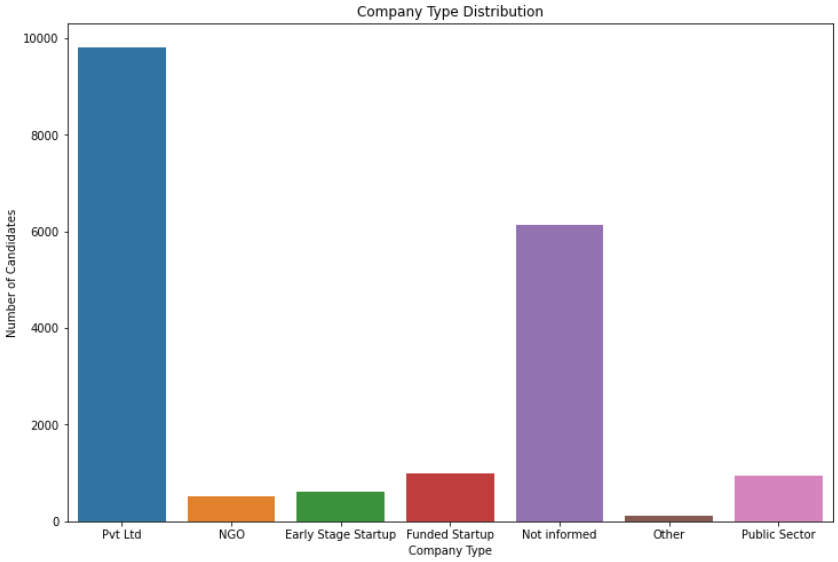
8. Distribuição dos anos de experiencia em Ciência de dados



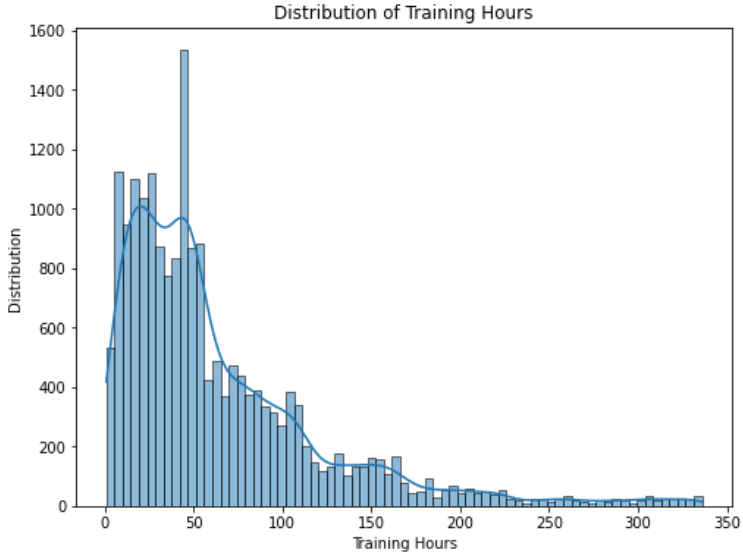
9. Distribuição dos candidatos em relação do tamanho da empresa



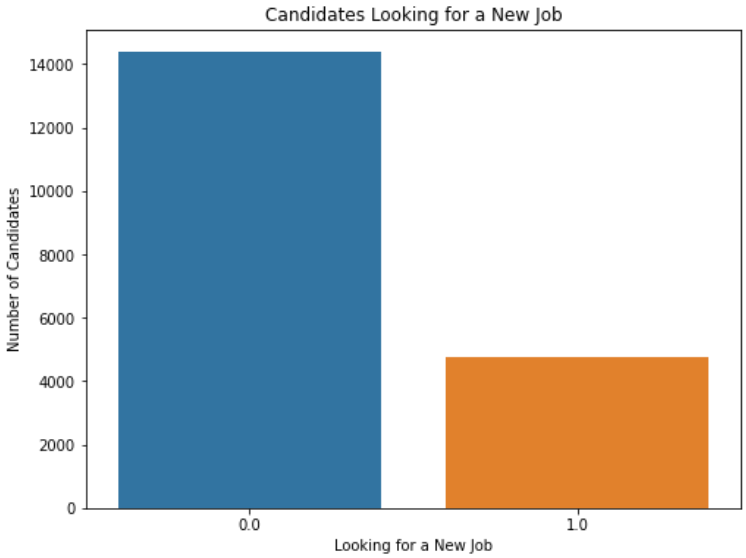
10. Distribuição dos candidatos em relação ao tipo da empresa



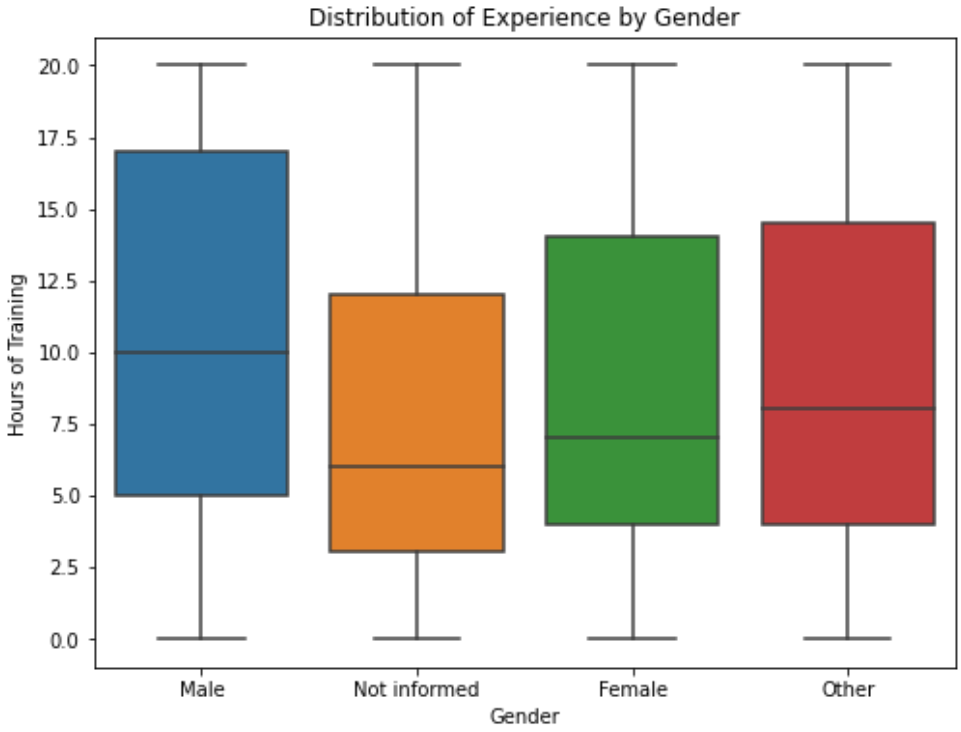
11. Distribuição de frequência (histograma) das horas de treinamento dos candidatos



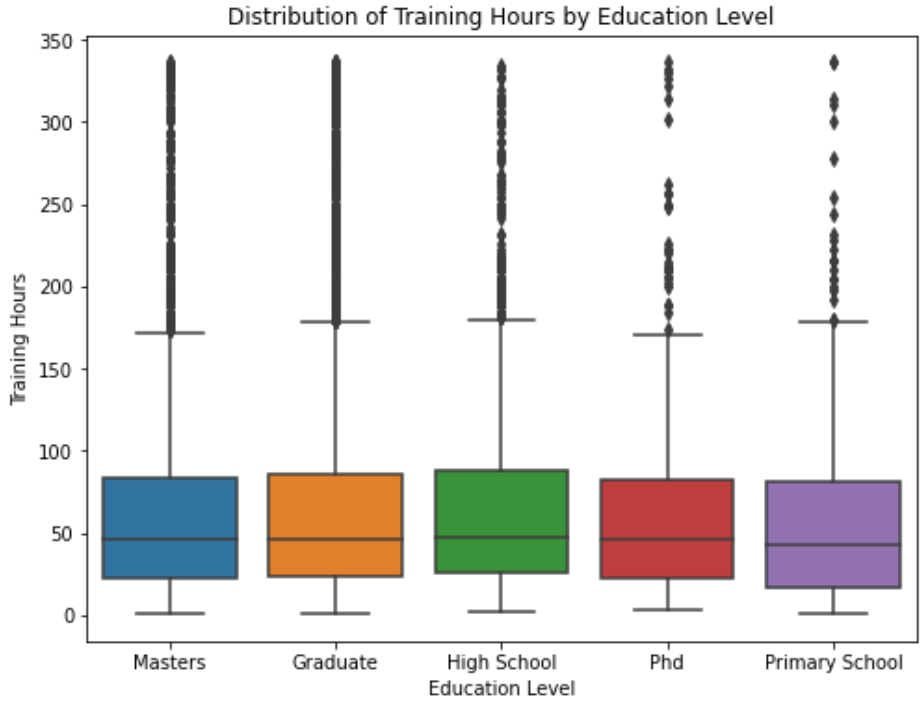
12. Distribuição da variável target



13. Boxplot para 'experiência' por 'gênero' para identificar outliers nas variáveis analisadas



14. Boxplot para 'training_hours' por 'education_level' para identificar outliers nas variáveis analisadas



Os gráficos de boxplot é uma representação gráfica que resume a distribuição de uma variável, destacando sua dispersão, simetria e possíveis **outliers**.

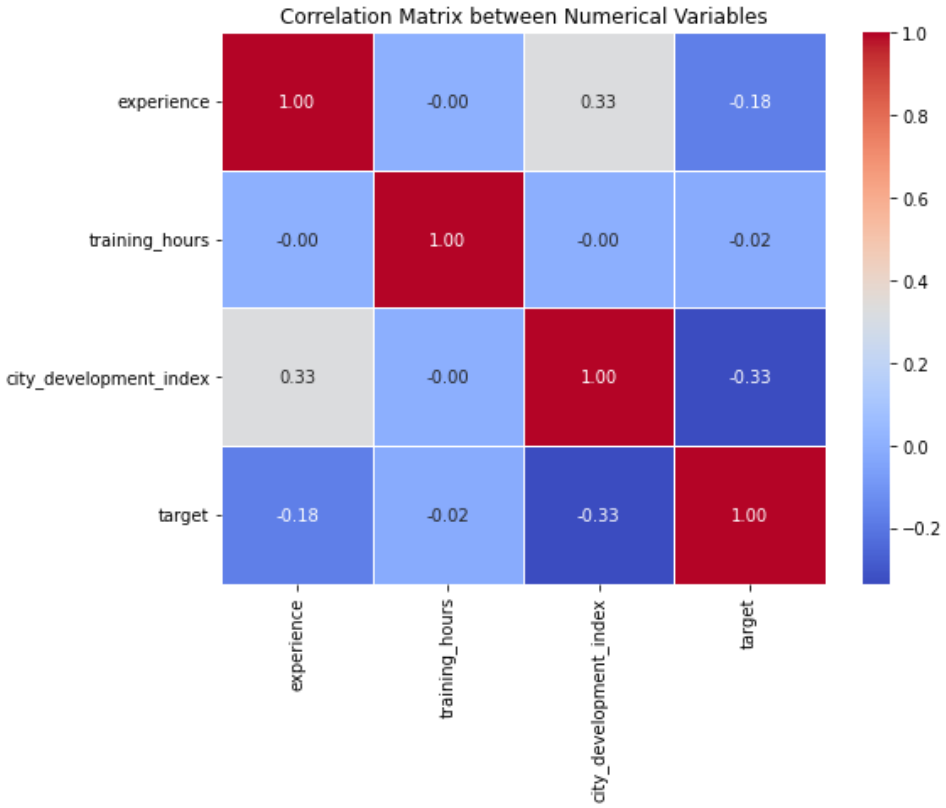
Explicando de modo geral, o que se observa nos gráficos acima é:

1. **Mediana (linha no meio da caixa)** → Indica o valor central da distribuição.

2. **Quartis →**
    - **Q1 (1º quartil, 25%)** → 25% dos dados estão abaixo desse valor.
    - **Q3 (3º quartil, 75%)** → 75% dos dados estão abaixo desse valor.
3. **Intervalo Interquartil (IQR - Interquartile Range)** → Diferença entre **Q3 e Q1**, mostrando a dispersão dos valores centrais.
4. **Extremos** → Representam a faixa onde os dados normalmente variam sem considerar outliers.
5. **Outliers** → Valores muito altos ou muito baixos que fogem do padrão observado nos dados.

O boxplot pode nos falar se a mediana está deslocada dentro da caixa, a distribuição pode ser inclinada (skewed), qual a variação da variável e se há pontos isolados fora dos limites, indicando valores atípicos.

15. Matriz de correlação entre as variáveis 'experience', 'training_hours', 'city_development_index' e 'target'



A matriz de correlação apresentada acima mostra a relação entre várias variáveis numéricas do conjunto de dados estudado. Cada célula na matriz representa o coeficiente de correlação entre duas variáveis, indicando o grau e a direção do relacionamento entre elas. Pela análise da matriz de correlação não existe uma correlação considerável entre as variáveis '*experience*', '*training_hours*', '*city_development_index*' e '*target*'.

Algumas das perguntas descritas nos objetivos podem ser respondidas a partir da análise dos gráficos anteriores, mas focando o MVP para consultas SQL, temos que na sequência podemos obter essas respostas.

1. Perfil dos profissionais:

A primeira análise foi feita em cima do perfil dos profissionais. Após a separação da tabela principal '*data_science_jobs_table*' nas tabelas: *dim_city, dim_company, dim_professional, fact_jobs*.

- Qual a distribuição de gênero dos candidatos?

Foi usada a query abaixo calcular a distribuição de gênero dos candidatos na tabela '*fact_jobs*', mostrando tanto o número total de candidatos por gênero quanto o percentual em relação ao total geral.

```
%sql
SELECT p.gender, COUNT(*) AS total_candidatos,
(COUNT(*) * 100.0 / SUM(COUNT(*)) OVER ()) AS percentual
FROM fact_jobs f
JOIN dim_professional p ON f.id_professional = p.id_professional
GROUP BY p.gender
ORDER BY total_candidatos DESC;
```

	gender	total_candidatos	percentual
1	Male	13221	69.01033510804886
2	Not informed	4508	23.53063994153878
3	Female	1238	6.46205240630546
4	Other	191	0.99697254410690

O resultado mostra que a maioria dos candidatos são do gênero masculino, uma boa parcela também não havia informado qual o gênero mas prevalece ainda assim o gênero masculino. Algo que realmente se identifica nos perfis dos candidatos da área de tecnologia, sendo uma predominância masculina.

- Qual o nível de formação mais comum entre os profissionais de ciência de dados?

A query abaixo tem o objetivo de analisar o nível de formação acadêmica dos candidatos, contando quantos profissionais pertencem a cada categoria de nível educacional. Dessa forma, seleciona a coluna '*education_level*' da tabela '*dim_professional*,' que representa o nível de formação do candidato. Conta-se o número total de candidatos para cada nível de formação usando 'COUNT(*) AS total_candidatos'. A tabela '*fact_jobs*' é unida (JOIN) à '*dim_professional*' através da chave '*id_professional*', que identifica cada candidato unicamente. E por último os resultados são agrupados pelo nível de formação (*education_level*). E ordenada em forma decrescente do nível de formação que possui mais candidatos para o que possui menos.

```
%sql
SELECT p.education_level, COUNT(*) AS total_candidatos
FROM fact_jobs f
JOIN dim_professional p ON f.id_professional = p.id_professional
GROUP BY p.education_level
ORDER BY total_candidatos DESC;
```

	education_level	total_candidatos
1	Graduate	12058
2	Masters	4361
3	High School	2017
4	Phd	414
5	Primary School	308

O resultado mostra que a maioria dos candidatos da tabela possuem apenas graduação, na sequencia mestrado, ensino médio, Phd e escola primaria. A área de Ciência de dados permite que possuindo conhecimento sobre o tema, mesmo sem graduação, existem pessoas que trabalham na área, pois possuem o conhecimento técnico necessário. Além de no setor a busca por mestrado ou doutorado é mais voltada para quem busca a área acadêmica.

- Qual a experiência média dos candidatos?

A query usada tem o objetivo de calcular a experiência média (em anos) dos candidatos que fazem parte da tabela '*fact_jobs*'. Calculando primeiro a média da coluna '*experience*' da tabela '*dim_professional*', o resultado representa a quantidade média de anos de experiência dos candidatos. A tabela '*fact_jobs*' é unida à '*dim_professional*' através da chave *id_professional*, que identifica cada candidato.

```
%sql
SELECT AVG(p.experience) AS experiencia_media
FROM fact_jobs f
JOIN dim_professional p ON f.id_professional = p.id_professional;
```

	1.2 experiencia_media
1	9.921494936841006

O resultado apresentado relaciona todos os candidatos que existem nas tabelas, porém não representa fielmente a variável estudada, visto que tem candidatos que possuem poucos anos de experiencia, enquanto outros estão anos a frente.

- Quantas pessoas já têm experiência relevante na área?

A query tem o objetivo de contar quantos candidatos possuem ou não experiência relevante na área. Primeiro analisa a coluna '*relevent_experience*' da tabela '*dim_professional*', que indica se o candidato tem experiência relevante na área de ciência de dados. Depois conta quantos registros existem para cada categoria da coluna '*relevent_experience*' (COUNT(*) AS total_candidatos). A tabela '*fact_jobs*' é unida à '*dim_professional*' através da chave '*id_professional*'. A query agrupa os resultados de acordo com o valor da coluna '*relevent_experience*', gerando uma contagem para cada categoria.

```sql
%sql
SELECT p.relevent_experience, COUNT(*) AS total_candidatos
FROM fact_jobs f
JOIN dim_professional p ON f.id_professional = p.id_professional
GROUP BY p.relevent_experience;
```

	relevent_experience	total_candidatos
1	No relevent experience	5366
2	Has relevent experience	13792

Pode-se observar que a maioria dos candidatos possuem experiencia relativa na área de ciência de dados, o que é interessante para empresas que buscam candidatos nesse perfil e que possuam experiencia que possa agregar no time de Ciência de Dados.

2. Fatores que Impactam a Contratação

- O nível de educação influencia na intenção de trocar de emprego?

Os candidatos possuem a intenção de trocar de emprego quando a variável '*target*' possui o valor igual a 1. Desse modo, a query abaixo analisa a relação entre o nível de educação dos candidatos e a intenção de trocar de emprego. Ela calcula quantos candidatos pertencem a cada nível educacional e a porcentagem dos que desejam mudar de emprego. Assim, obtém-se a coluna '*education_level*' da tabela '*dim_professional*', que contém informações sobre o grau de formação do candidato.

Conta quantos candidatos existem para cada categoria de nível educacional (COUNT(*) AS total_candidatos). Soma o total de candidatos que querem trocar de emprego (SUM(f.target) AS total_querem_trocar). Calcula o percentual de troca de emprego ((SUM(f.target) * 100.0 / COUNT(*)) AS percentual_troca). Depois, a '*fact_jobs*' é unida com '*dim_professional*' através da chave *id_professional*. E os resultados são agrupados pelo nível educacional para calcular as métricas de cada grupo. Os resultados são classificados do maior para o menor percentual de intenção de troca de emprego.

```sql
%sql
SELECT p.education_level,
       COUNT(*) AS total_candidatos,
       SUM(f.target) AS total_querem_trocar,
       (SUM(f.target) * 100.0 / COUNT(*)) AS percentual_troca
FROM fact_jobs f
JOIN dim_professional p ON f.id_professional = p.id_professional
GROUP BY p.education_level
ORDER BY percentual_troca DESC;
```

	education_level	total_candidatos	total_querem_trocar	percentual_troca
1	Graduate	12058	3349	27.77409188920219
2	Masters	4361	935	21.44003668883284
3	High School	2017	394	19.53396132870600
4	Phd	414	58	14.00966183574879
5	Primary School	308	41	13.31168831168831

O percentual de candidatos que querem trocar de emprego, ou seja, que estão à procura de emprego é maior para candidatos apenas com graduação (27,77%) , seguido pelos candidatos de mestrado (21,44%), ensino médio (19,53%) , Phd (14,00%) e escola primária (13,31%).

- Candidatos que fizeram mais horas de treinamento têm mais chances de mudar de emprego?

A query categoriza os candidatos com base no número de horas de treinamento e analisa a relação entre essa categoria e a intenção de trocar de emprego. O objetivo é verificar se há um padrão entre o tempo de treinamento e a decisão de buscar uma nova oportunidade. Primeiro os candidatos são separados em categorias de treinamento (CASE WHEN ... END AS categoria_treinamento), a coluna '*training_hours*' da tabela '*fact_jobs*' representa a quantidade de horas de treinamento dos candidatos. Os candidatos são categorizados em três grupos:

- "Pouco Treinamento": Menos de 20 horas.
- "Treinamento Moderado": Entre 20 e 50 horas.
- "Muito Treinamento": Acima de 50 horas.

Após essa separação, conta-se o total de candidatos por categoria de treinamento (COUNT(*) AS total_candidatos), Soma os candidatos que querem trocar de emprego (SUM(f.target) AS total_querem_trocar) e calcula o percentual de troca de emprego ((SUM(f.target) * 100.0 / COUNT(*)) AS percentual_troca). Os cálculos são feitos para cada grupo de treinamento e a saída é classificada do maior para o menor percentual de troca de emprego, permitindo analisar quais grupos têm maior propensão a querer mudar de trabalho.

```
%sql
SELECT
    CASE
        WHEN f.training_hours < 20 THEN 'Pouco Treinamento'
        WHEN f.training_hours BETWEEN 20 AND 50 THEN 'Treinamento Moderado'
        ELSE 'Muito Treinamento'
    END AS categoria_treinamento,
    COUNT(*) AS total_candidatos,
    SUM(f.target) AS total_querem_trocar,
    (SUM(f.target) * 100.0 / COUNT(*)) AS percentual_troca
FROM fact_jobs f
GROUP BY categoria_treinamento
ORDER BY percentual_troca DESC;
```

	categoria_treinamento	total_candidatos	total_querem_trocar	percentual_troca
1	Treinamento Moderado	6868	1756	25.56785090273733
2	Pouco Treinamento	3704	935	25.24298056155508
3	Muito Treinamento	8586	2086	24.29536454693687

Dessa forma, analisando o resultado dessa query, verifica-se que apesar de diferentes horas de treinamento entre os candidatos, o percentual de troca entre os três tipos de categorias é bem próximo, não possuindo influencia na intenção de trocar de emprego.

- Qual o impacto do tamanho da empresa na retenção de talentos?

Até o momento as análises estavam sendo feitas em cimas dos candidatos que possuem a intenção de buscar emprego, ou seja, o *'target'* igual a 1. Agora a avaliação será sobre a retenção dos candidatos nas empresas, logo o 'target' é igual a 0. O tamanho da empresa representa a quantidade de funcionários existentes na mesma.

Inicialmente seleciona o tamanho da empresa (*e.company_size*) na tabela '*dim_company*', calcula o total de candidatos (COUNT(*) AS total_candidatos) em cada categoria de empresa. Soma o número de candidatos retidos (SUM(CASE WHEN f.target = 0 THEN 1 ELSE 0 END) AS candidatos_retidos). Calcula o percentual de retenção ((SUM(CASE WHEN f.target = 0 THEN 1 ELSE 0 END) * 100.0 / COUNT(*)) AS percentual_retencao) e agrupa os resultados pelo tamanho da empresa (GROUP BY e.company_size). A Ordenação dos resultados é feita do maior para o menor percentual de retenção (ORDER BY percentual_retencao DESC).

```
%sql
SELECT e.company_size,
       COUNT(*) AS total_candidatos,
       SUM(CASE WHEN f.target = 0 THEN 1 ELSE 0 END) AS candidatos_retidos,
       (SUM(CASE WHEN f.target = 0 THEN 1 ELSE 0 END) * 100.0 / COUNT(*)) AS percentual_retencao
FROM fact_jobs f
JOIN dim_company e ON f.id_company = e.id_company
GROUP BY e.company_size
ORDER BY percentual_retencao DESC;
```

	company_size	total_candidatos	candidatos_retidos	percentual_retencao
1	1000-4999	1328	1128	84.93975903614458
2	100-500	2571	2156	83.85842084791910
3	<10	1308	1084	82.87461773700306
4	500-999	877	725	82.66818700114025
5	50-99	3083	2538	82.32241323386312
6	5000-9999	563	461	81.88277087033748
7	10000+	2019	1634	80.93115403665181
8	10/49	1471	1127	76.61454792658056
9	Not informed	5938	3528	59.41394408891883

Pode-se observar nos resultados que as empregas com tamanho '1000-4999' possui o maior percentual de retenção. Não ficando distante do percentual de retenção de empresas com '<10' funcionários. A maior parte dos candidatos, independente do tamanho da empresa, não estão em busca de emprego. O menor percentual ficou para a

categoria '*Not informed*' que são os candidatos que não informaram o tamanho da companhia, ou por não estarem trabalhando ou por não ter a informação.

- Empresas privadas (Pvt Ltd) perdem mais funcionários do que startups?

A análise será feita em duas categorias de Startups presentes na tabela, sendo: *'Early Stage Startup', 'Funded Startup'*, comparando com as empresas privadas. A variável *'target'* igual a 1 é a interpretação para a perda ou saída do funcionário (pois está a procura de emprego).

Primeiro seleciona o tipo de empresa (*e.company_type*), conta o total de candidatos (COUNT(*) AS total_candidatos), assim como o número de candidatos que desejam trocar de emprego (SUM(f.target) AS total_querem_trocar). Calcula o percentual de troca ((SUM(f.target) * 100.0 / COUNT(*)) AS percentual_troca) em relação ao total de funcionários daquela empresa. Filtra os resultados para incluir apenas os tipos de empresa desejados (WHERE e.company_type IN ('Pvt Ltd', 'Early Stage Startup', 'Funded Startup')). Os resultados são agrupados por tipo de empresa (GROUP BY e.company_type) e ordenados do maior para o menor percentual de troca (ORDER BY percentual_troca DESC).

```
%sql
SELECT e.company_type,
       COUNT(*) AS total_candidatos,
       SUM(f.target) AS total_querem_trocar,
       (SUM(f.target) * 100.0 / COUNT(*)) AS percentual_troca
FROM fact_jobs f
JOIN dim_company e ON f.id_company = e.id_company
WHERE e.company_type IN ('Pvt Ltd', 'Early Stage Startup', 'Funded Startup')
GROUP BY e.company_type
ORDER BY percentual_troca DESC;
```

	company_type	total_candidatos	total_querem_trocar	percentual_troca
1	Early Stage Startup	603	142	23.54892205638474
2	Pvt Ltd	9817	1775	18.08088010593868
3	Funded Startup	1001	140	13.98601398601399

Estabelecendo essa relação entre os candidatos dos três tipos de empresa, tem-se que o percentual de troca entre os candidatos de Startups em estágio inicial possuem uma intenção de troca maior do que de empresas privadas. Alguns motivos, como salário competitivo, a busca por desafios maiores, ou por ascensão de carreira possam causar essa procura por emprego.

3. Localização e Desenvolvimento do Mercado

- Em quais cidades há maior concentração de profissionais de ciência de dados?

Na tabela '*dim_city*', a coluna '*id_city*' é um identificador único para cada cidade onde os candidatos estão localizados. As cidades estão representadas por números, não havendo a possibilidade de saber qual seria a localização exata, e possui 123 valores distintos. Desse modo, a query irá apresentar apenas as 10 primeiras cidades com a maior concentração de profissionais de Ciência de dados.

A query seleciona o identificador da cidade (*l.id_city*) e o índice de desenvolvimento da cidade (*l.city_development_index*), conta o número total de candidatos por cidade (COUNT(*) AS total_candidatos), faz a junção (*JOIN*) entre a tabela fato e a dimensão cidade (JOIN dim_city l ON f.id_city = l.id_city). Os dados são agrupados por cidade (GROUP BY l.id_city)e ordenados de forma decrescente (ORDER BY total_candidatos DESC).

```
%sql
SELECT l.id_city, l.city_development_index, COUNT(*) AS total_candidatos
FROM fact_jobs f
JOIN dim_city l ON f.id_city = l.id_city
GROUP BY l.id_city, l.city_development_index
ORDER BY total_candidatos DESC
LIMIT 10;
```

	id_city	city_development_index	total_candidatos
1	103	0.919	4355
2	21	0.632	2702
3	16	0.91	1533
4	114	0.925	1336
5	160	0.92	845
6	136	0.897	586
7	67	0.856	431
8	75	0.938	305
9	102	0.807	304
10	104	0.923	301

O resultado não torna possível estabelecer uma relação entre a cidade, o índice de desenvolvimento da mesma e a quantidade de candidatos na área de Ciência de dados, pois a cidade de id 103, com índice de desenvolvimento acima de 0.9 possui a maior quantidade de candidatos, enquanto em segundo lugar está a cidade de id 21, com índice de desenvolvimento 0.632.

- O índice de desenvolvimento da cidade (*city_development_index*) tem impacto na retenção de talentos?

A relação será feita entre os maiores índices de desenvolvimento da cidade e a retenção de talentos, ou seja, o *'target'* igual a 0, os candidatos não estão em busca de emprego. A query seleciona a cidade (*l.id_city*) na tabela '*dim_city'*, representando a identificação única da cidade, arredonda o índice de desenvolvimento da cidade (ROUND(l.city_development_index, 2) AS cdi_arredondado). Em seguida, conta o total de candidatos por cidade (COUNT(*) AS total_candidatos), calcula o total de candidatos retidos (SUM(CASE WHEN f.target = 0 THEN 1 ELSE 0 END) AS candidatos_retidos), calcula o percentual de retenção de candidatos ((SUM(CASE WHEN f.target = 0 THEN 1 ELSE 0 END) * 100.0 / COUNT(*)) AS percentual_retencao).

Faz um *JOIN* entre a tabela fato e a dimensão cidade (JOIN dim_city l ON f.id_city = l.id_city), agrupa os dados por cidade e índice de desenvolvimento (GROUP

BY l.id_city, l.city_development_index), ordena as cidades pelo índice de desenvolvimento em ordem decrescente (ORDER BY l.city_development_index DESC) e apenas as 10 cidades mais desenvolvidas são exibidas.

```
%sql
SELECT
    l.id_city,
    ROUND(l.city_development_index, 2) AS cdi_arredondado,
    COUNT(*) AS total_candidatos,
    SUM(CASE WHEN f.target = 0 THEN 1 ELSE 0 END) AS candidatos_retidos,
    (SUM(CASE WHEN f.target = 0 THEN 1 ELSE 0 END) * 100.0 / COUNT(*)) AS percentual_retencao
FROM fact_jobs f
JOIN dim_city l ON f.id_city = l.id_city
GROUP BY l.id_city, l.city_development_index
ORDER BY l.city_development_index DESC
LIMIT 10;
```

	id_city	cdi_arredondado	total_candidatos	candidatos_retidos	percentual_retencao
1	98	0.95	79	71	89.87341772151899
2	75	0.94	305	274	89.83606557377049
3	28	0.94	192	177	92.18750000000000
4	114	0.93	1336	1203	90.04491017964072
5	97	0.93	104	96	92.30769230769231
6	89	0.92	67	51	76.11940298507463
7	104	0.92	301	273	90.69767441860465
8	83	0.92	143	120	83.91608391608392
9	167	0.92	10	7	70.00000000000000
10	160	0.92	845	646	76.44970414201183

O resultado da query mostra que apesar de possuir o índice de desenvolvimento da cidade de id 98, com valor aproximadamente 0.95, o percentual de retenção não é superior ao da cidade de id aproximadamente 0.93, que possui um percentual de 92,3% de retenção de candidatos. Essa analise provavelmente mudaria de resultado se os resultados fossem organizados de acordo com o percentual de retenção.

4. Análises sobre Empresas

- Qual o porte das empresas que mais contratam profissionais de ciência de dados?

A query analisa a distribuição de candidatos conforme o tamanho da empresa (*company_size*), identificando quantos profissionais trabalham em empresas de diferentes portes. Seleciona o tamanho da empresa (*e.company_size*) na tabela '*dim_company*', conta o total de candidatos por tamanho de empresa (COUNT(*) AS total_candidatos), faz um JOIN entre a tabela fato e a dimensão empresa (JOIN dim_company e ON f.id_company = e.id_company) e agrupa os dados pelo tamanho da empresa (GROUP BY e.company_size). Os resultados são ordenados pelo total de candidatos em ordem decrescente (ORDER BY total_candidatos DESC).

```
%sql
SELECT e.company_size, COUNT(*) AS total_candidatos
FROM fact_jobs f
JOIN dim_company e ON f.id_company = e.id_company
GROUP BY e.company_size
ORDER BY total_candidatos DESC;
```

	company_size	total_candidatos
1	Not informed	5938
2	50-99	3083
3	100-500	2571
4	10000+	2019
5	10/49	1471
6	1000-4999	1328
7	<10	1308
8	500-999	877
9	5000-9999	563

O resultado mostra que essa analise seria interessante para verificar se o tamanho das empresas influencia na quantidade de contratação de candidatos de Ciência de dados, porém uma grande quantidade de candidatos não informou o tamanho da empresa, fazendo com que essa análise não tenha uma conclusão tão clara. De todo modo, é interessante constatar que empresas com tamanho '50-99' possuem mais candidatos contratados do setor do que empresas de tamanho '5000-9999'.

- Startups contratam mais profissionais juniores ou seniores?

A query analisa a distribuição de candidatos por nível de experiência (*experience_group*), considerando apenas aqueles que trabalham em startups (*Early Stage Startup e Funded Startup*). Seleciona o grupo de experiência *(p.experience_group*) na tabela *'dim_professional'*. Conta o total de candidatos por grupo de experiência (COUNT(*) AS total_candidatos). Faz um *JOIN* entre a tabela fato e as dimensões empresa e profissional

- JOIN 'dim_company' e ON 'f.id_company' = 'e.id_company' → Relaciona a tabela fato com a tabela de empresas para obter informações sobre o tipo da empresa.
- JOIN 'dim_professional' p ON 'f.id_professional' = 'p.id_professional' → Relaciona a tabela fato com a tabela de profissionais para obter informações sobre sua experiência.

Filtra apenas candidatos que trabalham em startups (WHERE e.company_type IN ('Early Stage Startup', 'Funded Startup')), agrupa os dados pelo grupo de experiência (GROUP BY p.experience_group) e ordena os resultados pelo total de candidatos em ordem decrescente (ORDER BY total_candidatos DESC).

```
%sql
SELECT p.experience_group, COUNT(*) AS total_candidatos
FROM fact_jobs f
JOIN dim_company e ON f.id_company = e.id_company
JOIN dim_professional p ON f.id_professional = p.id_professional
WHERE e.company_type IN ('Early Stage Startup', 'Funded Startup')
GROUP BY p.experience_group
ORDER BY total_candidatos DESC;
```

	experience_group	total_candidatos
1	Sênior	1082
2	Pleno	363
3	Júnior	159

Pelo resultado pode-se constatar que as startups contratam mais profissionais Sênior do que Junior.

- Empresas grandes (5000+ funcionários) investem mais em treinamento do que empresas pequenas?

A query seleciona os dados da coluna '*company_size'*, da tabela '*dim_company'* nomeando de 'porte_empresa': Esta coluna é gerada por uma instrução CASE que classifica o tamanho da empresa com base no valor da coluna *company_size*. As classificações são:

- "Grande" para empresas com tamanho 5000-9999 ou 10000+.
- "Pequena" para empresas com tamanhos <10, 10-49, 50-99, ou 100-500.
- "Média" para qualquer outro valor que não se encaixe nas classificações anteriores.

É calculada como a média (AVG) das horas de treinamento registradas na tabela '*fact_jobs'*(coluna '*training_hours'*). A query realiza um *JOIN* entre as tabelas '*fact_jobs*' e '*dim_company*' com base na correspondência entre os campos '*id_company*' de ambas as tabelas. A consulta agrupa os resultados pela coluna calculada *'porte_empresa'*. Em seguida, a consulta ordena os resultados pela média das horas de treinamento (media_horas_treinamento) de forma decrescente (DESC).

```
%sql
SELECT
    CASE
        WHEN e.company_size IN ('5000-9999', '10000+') THEN 'Grande'
        WHEN e.company_size IN ('<10', '10-49', '50-99', '100-500') THEN 'Pequena'
        ELSE 'Média'
    END AS porte_empresa,
    AVG(f.training_hours) AS media_horas_treinamento
FROM fact_jobs f
JOIN dim_company e ON f.id_company = e.id_company
GROUP BY porte_empresa
ORDER BY media_horas_treinamento DESC;
```

	porte_empresa	1.2 media_horas_treinamento
1	Pequena	65.1581442114335
2	Média	64.1251300187227
3	Grande	63.51781564678544

Os resultados dessa query ficaram bem próximos, mostrando que não há uma relação entre o tamanho da empresa e as horas de treinamento dos candidatos.

Conclusão: Autoavaliação

A utilização de dados nos ajuda a obter *insights* que se tornam difíceis sem os mesmos. A intenção do MVP foi a análise de um *dataset* de candidatos de Ciência de dados em diferentes cidades. A profissão tem se tornado bastante procurada atualmente devido a busca pela utilização dos dados da empresa como fonte de respostas e soluções inteligentes para os gestores.

A partir do conjunto de dados estudado, assim como de uma plataforma que possa fazer o armazenamento e a análises desses dados utilizando a estrutura disponível (*Cluster*) dessa plataforma, torna-se acessível o uso de recursos poderosos, sendo uma excelente solução para empresas que buscam ter uma análise rápida e aprofundada de diferentes tipos de dados.

O uso das linguagens Python e SQL possibilitou a leitura, transformação, armazenamento das variáveis de interesse, a criação de gráficos que resumem os dados de forma visual e atrativa para os gestores, assim como a criação de diferentes cenários que agrupam e resumem os dados tornando possível uma compreensão rápida sem precisar ler registro por registro de uma tabela.

Os objetivos do MVP foram construídos de modo a estudar a distribuição dos dados dos candidatos e suas informações pertinentes a área de Ciência de dados. Informações que se tornam valiosas na mão de recrutadores de empresas, por exemplo, pois mostra as tendencias encontradas nos candidatos que estão em busca de emprego no setor, assim como para pessoas que estão em transição de carreira para a área, saber quais as áreas de estudo mais buscadas, como os anos de experiencia influenciam na contratação ou não dos candidatos, entre outras conclusões e observações que podem ser feitas em cima do *dataset*. Algumas inconsistências foram encontradas em alguns atributos da tabela, que influenciaram na análise, mas foram tratados ao longo da manipulação da mesma.

Apesar de terem sido respondidas as perguntas propostas inicialmente, algumas considerações podem ser feitas e que agregariam mais valor à análise, como por exemplo, os dados não possuem uma dimensão de tempo, não sendo possível obter uma visualização da tendencia dos candidatos buscarem emprego ou não ao longo do tempo. Outra possibilidade seria a junção de *dataset* com informações de salário, tipo de contratação (PJ, CLT,...), regime de trabalho (remoto, híbrido, presencial,...), entre outras informações, que permitiria uma análise ainda mais robusta e com *insights* importantes.

Todos os objetivos propostos na construção do MVP puderam ser respondidos ao longo do trabalho, mostrando que a linguagem SQL é aliada na análise de resultados para compreensão dos dados estudados.