

```
In [4]: import sqlite3
import pandas as pd
conn = sqlite3.connect("database.db")
```

```
In [5]: # Consulta dos dados no banco de dados

consulta_atividade = """

    SELECT
        fa.*
    FROM flight_activity fa
    WHERE
        fa.flights_booked > 11

"""

df_atividade = pd.read_sql_query(consulta_atividade, conn)
```

```
In [6]: df_atividade.head()
```

```
Out[6]:
```

	loyalty_number	year	month	flights_booked	flights_with_companions	total_flights	distance
0	471706	2018	7	12	10	22	58
1	105932	2017	1	13	6	19	16
2	107212	2017	1	12	3	15	24
3	100504	2017	10	14	0	14	38
4	572345	2018	5	12	7	19	47

```
In [7]: # Selecione as colunas: loyalty_number, year, month, flights_booked, total_flights
```

```
In [8]: consulta_atividade = """

    SELECT
        fa.loyalty_number,
        fa.year,
        fa.month,
        fa.flights_booked,
        fa.total_flights,
        fa.distance,
        fa.points_accumulated
    FROM
        flight_activity fa
    WHERE
        fa.flights_booked > 11

"""

df_atividade = pd.read_sql_query(consulta_atividade, conn)
```

```
In [9]: df_atividade.head()
```

Out[9]:

	loyalty_number	year	month	flights_booked	total_flights	distance	points_accumulated
0	471706	2018	7	12	22	5896	589.0
1	105932	2017	1	13	19	1653	165.0
2	107212	2017	1	12	15	2490	249.0
3	100504	2017	10	14	14	3570	357.0
4	572345	2018	5	12	19	4769	476.0

```
In [10]: # Selezione as

consulta_atividade = """

    SELECT
        fa.loyalty_number,
        fa.year,
        fa.month,
        fa.flights_booked,
        fa.total_flights,
        fa.distance,
        fa.points_accumulated
    FROM
        flight_activity fa
    WHERE
        fa.distance > 2000
    """

df_atividade = pd.read_sql_query(consulta_atividade, conn)
```

```
In [11]: df_atividade.head()
```

Out[11]:

	loyalty_number	year	month	flights_booked	total_flights	distance	points_accumulated
0	100102	2017	1	10	14	2030	203.0
1	100550	2017	1	3	3	2037	203.0
2	863070	2017	9	8	15	4245	424.0
3	100753	2017	1	8	12	3264	326.0
4	100816	2017	1	9	10	2340	234.0

```
In [12]: consulta_atividade = """

    SELECT
        fa.loyalty_number,
        fa.year,
        fa.month,
        fa.flights_booked,
        fa.total_flights,
        fa.distance,
        fa.points_accumulated
    FROM
        flight_activity fa
    WHERE
        fa.distance > 2000 and fa.month = 9
    """

df_atividade = pd.read_sql_query(consulta_atividade, conn)
```

```
In [13]: df_atividade.head()
```

Out[13]:

	loyalty_number	year	month	flights_booked	total_flights	distance	points_accumulated
0	863070	2017	9	8	15	4245	424.0
1	691626	2018	9	8	15	4245	424.0
2	444931	2017	9	11	18	4428	442.0
3	409051	2018	9	11	18	4428	442.0
4	975387	2018	9	13	18	4428	442.0

In [14]:

```
consulta_atividade = """

    SELECT
        fa.loyalty_number,
        fa.year,
        fa.month,
        fa.flights_booked,
        fa.total_flights,
        fa.distance,
        fa.points_accumulated
    FROM
        flight_activity fa
    WHERE
        fa.distance > 2000 or fa.points_accumulated < 100
    """
df_atividade = pd.read_sql_query(consulta_atividade, conn)
```

In [15]:

```
df_atividade.head()
```

Out[15]:

	loyalty_number	year	month	flights_booked	total_flights	distance	points_accumulated
0	100102	2017	1	10	14	2030	203.0
1	100214	2017	1	0	0	0	0.0
2	100272	2017	1	0	0	0	0.0
3	100301	2017	1	0	0	0	0.0
4	100364	2017	1	0	0	0	0.0

In [16]:

```
consulta_atividade = """

    SELECT
        *
    FROM
        flight_loyalty_history flh
    WHERE
        flh.loyalty_card = "Star"
    """
df_atividade = pd.read_sql_query(consulta_atividade, conn)
```

In [17]:

```
df_atividade.head()
```

Out[17]:

	loyalty_number	country	province	city	postal_code	gender	education	salary
0	480934	Canada	Ontario	Toronto	M2Z 4K1	Female	Bachelor	83236.0
1	549612	Canada	Alberta	Edmonton	T3G 6Y6	Male	College	NaN
2	429460	Canada	British Columbia	Vancouver	V6E 3D9	Male	College	NaN
3	608370	Canada	Ontario	Toronto	P1W 1K4	Male	College	NaN
4	530508	Canada	Quebec	Hull	J8Y 3Z5	Male	Bachelor	103495.0

2.0. Exercicios de SQL

In [18]:

```
consulta_atividade = """

    SELECT
        *
    FROM flight_activity fa LEFT JOIN flight_loyalty_history flh ON (fa.loya

    """

df_atividade = pd.read_sql_query(consulta_atividade, conn)

df_atividade.head()
```

Out[18]:

	loyalty_number	year	month	flights_booked	flights_with_companions	total_flights	distar
0	100018	2017	1	3	0	3	11
1	100102	2017	1	10	4	14	20
2	100140	2017	1	6	0	6	12
3	100214	2017	1	0	0	0	
4	100272	2017	1	0	0	0	

5 rows × 26 columns

3.0. Inspeccionando os dados

In [19]:

```
df_atividade.head()
```

Out[19]:

	loyalty_number	year	month	flights_booked	flights_with_companions	total_flights	distar
0	100018	2017	1	3	0	3	11
1	100102	2017	1	10	4	14	20
2	100140	2017	1	6	0	6	12
3	100214	2017	1	0	0	0	
4	100272	2017	1	0	0	0	

5 rows × 26 columns

In [20]:

```
# Curiosidade o que é DF_atividade (Data Frame)

type(df_atividade)
```

Out[20]: pandas.core.frame.DataFrame

```
In [21]: # verificar o numero de linhas de uma planilha de dados (dataframe)

df_atividade.shape[0]
```

Out[21]: 405624

```
In [22]: # Verificar o numero de colunas

df_atividade.shape[1]
```

Out[22]: 26

```
In [23]: # Verificar o panorama geral da tabela
# Insights iniciais da planilha de dados.

df_atividade.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 405624 entries, 0 to 405623
Data columns (total 26 columns):
 #   Column                                Non-Null Count  Dtype
---  -
 0   loyalty_number                       405624 non-null int64
 1   year                                 405624 non-null int64
 2   month                               405624 non-null int64
 3   flights_booked                      405624 non-null int64
 4   flights_with_companions             405624 non-null int64
 5   total_flights                       405624 non-null int64
 6   distance                             405624 non-null int64
 7   points_accumulated                  405624 non-null float64
 8   points_redeemed                     405624 non-null int64
 9   dollar_cost_points_redeemed         405624 non-null int64
10   loyalty_number                       405624 non-null int64
11   country                             405624 non-null object
12   province                             405624 non-null object
13   city                                405624 non-null object
14   postal_code                          405624 non-null object
15   gender                              405624 non-null object
16   education                           405624 non-null object
17   salary                              302952 non-null float64
18   marital_status                      405624 non-null object
19   loyalty_card                        405624 non-null object
20   clv                                 405624 non-null float64
21   enrollment_type                     405624 non-null object
22   enrollment_year                     405624 non-null int64
23   enrollment_month                    405624 non-null int64
24   cancellation_year                   50064 non-null float64
25   cancellation_month                  50064 non-null float64
dtypes: float64(5), int64(12), object(9)
memory usage: 80.5+ MB
```

```
In [24]: df_atividade.loc[:, "distance"]
```

```
Out[24]: 0          1521
         1          2030
         2          1200
         3           0
         4           0
         ...
        405619         0
        405620         0
        405621        1233
        405622         0
        405623         0
        Name: distance, Length: 405624, dtype: int64
```

```
In [25]: df_atividade.loc[:, "distance"].mean()
```

```
Out[25]: 1208.880058872256
```

```
In [26]: df_atividade.loc[:, "distance"].max()
```

```
Out[26]: 6293
```

```
In [27]: df_atividade.loc[:, "distance"].min()
```

```
Out[27]: 0
```

```
In [28]: soma_distancia = df_atividade.loc[:, "distance"].sum()
        menor_distancia = df_atividade.loc[:, "distance"].min()
        maior_distancia = df_atividade.loc[:, "distance"].max()
        media_distancia = df_atividade.loc[:, "distance"].mean()
```

```
In [29]: df_atividade.head()
```

```
Out[29]:
```

	loyalty_number	year	month	flights_booked	flights_with_companions	total_flights	distar
0	100018	2017	1	3	0	3	15
1	100102	2017	1	10	4	14	20
2	100140	2017	1	6	0	6	12
3	100214	2017	1	0	0	0	0
4	100272	2017	1	0	0	0	0

5 rows × 26 columns

```
In [30]: # Identificar o numeros de dados faltantes
        df_atividade.isna()
        df_atividade.isna().sum

        # selecionar as colunas que contem numeros
        # remover as linhas que contem dados faltantes
        # Verificar se os dados existem fados faltantes
```

```

Out[30]: <bound method NDFrame._add_numeric_operations.<locals>.sum of
y_number    year    month    flights_booked    flights_with_companions    \
0                False    False    False                False                Fals
e
1                False    False    False                False                Fals
e
2                False    False    False                False                Fals
e
3                False    False    False                False                Fals
e
4                False    False    False                False                Fals
e
...                ...                ...                ...                ...
...
405619                False    False    False                False                Fals
e
405620                False    False    False                False                Fals
e
405621                False    False    False                False                Fals
e
405622                False    False    False                False                Fals
e
405623                False    False    False                False                Fals
e

                total_flights    distance    points_accumulated    points_redeemed    \
0                False                False                False                False
1                False                False                False                False
2                False                False                False                False
3                False                False                False                False
4                False                False                False                False
...                ...                ...                ...                ...
405619                False                False                False                False
405620                False                False                False                False
405621                False                False                False                False
405622                False                False                False                False
405623                False                False                False                False

                dollar_cost_points_redeemed    ...    education    salary    marital_status
\
0                False    ...                False    False                False
1                False    ...                False    True                False
2                False    ...                False    True                False
3                False    ...                False    False                False
4                False    ...                False    False                False
...                ...    ...                ...                ...
405619                False    ...                False    True                False
405620                False    ...                False    False                False
405621                False    ...                False    False                False
405622                False    ...                False    True                False
405623                False    ...                False    False                False

                loyalty_card    clv    enrollment_type    enrollment_year    \
0                False    False                False                False
1                False    False                False                False
2                False    False                False                False
3                False    False                False                False
4                False    False                False                False
...                ...                ...                ...
405619                False    False                False                False
405620                False    False                False                False
405621                False    False                False                False
405622                False    False                False                False
405623                False    False                False                False

```

```

      enrollment_month  cancellation_year  cancellation_month
0                False                True                True
1                False                True                True
2                False                True                True
3                False                True                True
4                False                True                True
...                ...                ...                ...
405619            False                True                True
405620            False                True                True
405621            False                True                True
405622            False                True                True
405623            False                True                True

[405624 rows x 26 columns]>
```

```
In [31]: # Identificar o numeros de dados faltantes
df_atividade.isna().sum()
```

```
Out[31]: loyalty_number          0
year                          0
month                        0
flights_booked               0
flights_with_companions      0
total_flights                0
distance                     0
points_accumulated           0
points_redeemed              0
dollar_cost_points_redeemed  0
loyalty_number               0
country                      0
province                     0
city                         0
postal_code                  0
gender                       0
education                    0
salary                       102672
marital_status               0
loyalty_card                 0
clv                          0
enrollment_type              0
enrollment_year              0
enrollment_month             0
cancellation_year            355560
cancellation_month            355560
dtype: int64
```

```
In [32]: df_atividade.info()
```



```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 405624 entries, 0 to 405623
Data columns (total 26 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   loyalty_number                        405624 non-null  int64
1   year                                  405624 non-null  int64
2   month                                405624 non-null  int64
3   flights_booked                       405624 non-null  int64
4   flights_with_companions              405624 non-null  int64
5   total_flights                        405624 non-null  int64
6   distance                             405624 non-null  int64
7   points_accumulated                  405624 non-null  float64
8   points_redeemed                     405624 non-null  int64
9   dollar_cost_points_redeemed         405624 non-null  int64
10  loyalty_number                       405624 non-null  int64
11  country                              405624 non-null  object
12  province                             405624 non-null  object
13  city                                 405624 non-null  object
14  postal_code                          405624 non-null  object
15  gender                               405624 non-null  object
16  education                            405624 non-null  object
17  salary                              302952 non-null  float64
18  marital_status                       405624 non-null  object
19  loyalty_card                         405624 non-null  object
20  clv                                  405624 non-null  float64
21  enrollment_type                      405624 non-null  object
22  enrollment_year                      405624 non-null  int64
23  enrollment_month                     405624 non-null  int64
24  cancellation_year                    50064 non-null   float64
25  cancellation_month                   50064 non-null   float64
dtypes: float64(5), int64(12), object(9)
memory usage: 80.5+ MB
```

```
In [33]: colunas = ["year", "month", "flights_booked", "flights_with_companions", "to
```

```
In [34]: df_atividade.loc[:, colunas]
```

Out[34]:

	year	month	flights_booked	flights_with_companions	total_flights	distance	point
0	2017	1	3	0	3	1521	
1	2017	1	10	4	14	2030	
2	2017	1	6	0	6	1200	
3	2017	1	0	0	0	0	
4	2017	1	0	0	0	0	
...
405619	2018	12	0	0	0	0	
405620	2018	12	0	0	0	0	
405621	2018	12	3	0	3	1233	
405622	2018	12	0	0	0	0	
405623	2018	12	0	0	0	0	

405624 rows × 10 columns

```
In [35]: print(colunas)
```

```
['year', 'month', 'flights_booked', 'flights_with_companions', 'total_flights', 'distance', 'points_accumulated', 'salary', 'clv', 'loyalty_card']
```

```
In [36]: df_colunas_numericas = df_atividade.loc[:, colunas]
```

```
In [37]: # remover as linhas que contem dados faltantes
```

```
df_dados_completos = df_colunas_numericas.dropna()
```

```
In [38]: # verificar se existe dados faltantes
```

```
df_dados_completos.isna().sum()
```

```
Out[38]: year                0
month                0
flights_booked       0
flights_with_companions  0
total_flights        0
distance             0
points_accumulated   0
salary              0
clv                 0
loyalty_card         0
dtype: int64
```

```
In [39]: df_dados_completos.shape[0]
```

```
Out[39]: 302952
```

5.0. Machine Learning

```
In [40]: from sklearn import tree as tr
```

```
In [41]: X_atributos = df_dados_completos.drop( columns="loyalty_card" )
y_rotulos = df_dados_completos.loc[:, "loyalty_card"]
```

```
# Definição do algoritmo
```

```
modelo = tr.DecisionTreeClassifier(max_depth=5)
```

```
# Treinamento do algoritmo
```

```
modelo_treinado = modelo.fit(X_atributos, y_rotulos)
```

```
In [ ]: tr.plot_tree(modelo_treinado, filled = True);
```



```
In [47]: X_novo = X_atributos.sample()
previsao = modelo_treinado.predict_proba(X_novo)

print( "Prob - Aurora: {:.1f}% - Nova: {:.1f}% = Star: {:.1f}%" .format(100*previsao[0][0], 100*previsao[0][1], 100*previsao[0][2]) )

Prob - Aurora: 33.9% - Nova: 0.3% = Star: 32.7%
```

7.0. Painel de Visualização

```
In [48]: import gradio as gr
import numpy as np
```

```
In [49]: X_atributos.loc[:, 'year'].min()
```

```
Out[49]: 2017
```

```
In [50]: X_atributos.loc[:, 'year'].max()
```

```
Out[50]: 2018
```

```
In [ ]: def predict(*args):
    X_novo = np.array( [args] ).reshape(1, -1)
    previsao = modelo_treinado.predict_proba( X_novo )

    return {"Aurora": previsao[0][0], "Nova":previsao[0][1], "Star": previsao[0][2]}

with gr.Blocks() as demo:
    # Titulo do Painel
    gr.Markdown( """ # Propensão de Compra """ )

    with gr.Row():
        with gr.Column():
            gr.Markdown( """ # Atributos do Cliente """ )
            year = gr.Slider( label= "year", minimum=2017, maximum=2018, value=2017 )
            month = gr.Slider( label= "month", minimum=1, maximum=12, value=1 )
            fligths_booked = gr.Slider( label= "fligths_booked", minimum=0, maximum=10, value=5 )
            fligths_with_companions = gr.Slider( label= "fligths_with_companions", minimum=0, maximum=10, value=5 )
            total_flights = gr.Slider( label= "total_flights", minimum=0, maximum=10, value=5 )
            distance = gr.Slider( label= "distance", minimum=0, maximum=1000, value=500 )
            points_accumulated = gr.Slider( label= "points_accumulated", minimum=0, maximum=1000, value=500 )
            salary = gr.Slider( label= "salary", minimum=58, maximum=1000, value=58 )
            clv = gr.Slider( label= "clv", minimum=2119, maximum=10000, value=2119 )

            with gr.Row():
                gr.Markdown( """# Botão de Previsão """ )
                predict_btn = gr.Button( value = "Previsao" )

        with gr.Column():
            gr.Markdown( """# Coluna 2 """ )
            label = gr.Label()

    # Botão de predict
    predict_btn.click(
        fn=predict,
        inputs=[
            year,
            month,
            fligths_booked,
            fligths_with_companions,
            total_flights,
            distance,
            points_accumulated,
            salary,
            clv
        ],
        outputs=[label]
    )
```

```
        points_accumulated,\n        salary,\n        clv\n    ],\n    outputs=[label])\n\ndemo.launch(debug=True, share=False)
```

Running on local URL: http://127.0.0.1:7860

To create a public link, set `share=True` in `launch()`.

Propensão de Compra

Atributos do Cliente

year

2018

month

12

fligths_booked

0

fligths_with_companions

4

total_flights

30

distance

1871

```
/Users/wallacefirmo/opt/anaconda3/lib/python3.9/site-packages/sklearn/base.py:450: UserWarning: X does not have valid feature names, but DecisionTreeClassifier was fitted with feature names\n  warnings.warn(\n/Users/wallacefirmo/opt/anaconda3/lib/python3.9/site-packages/sklearn/base.py:450: UserWarning: X does not have valid feature names, but DecisionTreeClassifier was fitted with feature names\n  warnings.warn(
```

COMUNIDADE DS

Wallace da Silva Firmo

Data Science

In []: