Data Science Analyst

(Software Product Ops Analyst) - Assessment

Horacio Morales
August; 2023

Horacio Morales
August; 2023

Data Science Analyst

(Software Product Ops Analyst) - Assessment

Summary

Objective 2
Problem 1 2
Problem 2 6
Problem 3 7

Objective

The objective of this document is to present the lead insights from Bright sales activities.

The data presented here was obtain from the database provided from https://ops.thinkbright.mx/api/files/eaf67029-57f1-4f5a-83b6-d97e1484682d?download=true&name=bright.sqlite

Problem 1

The dashboard was generated with datapane python library and saved as a HTML file: bright.html , in order to get easier reading is presented in this document.

The HTML presents 3 sections:

1. The main KPI's from sales activities: conversion rates from created leads to visited leads and approved leads. And the top 10 locations with approved leads.

2. How much time takes a lead to convert into a visited lead.

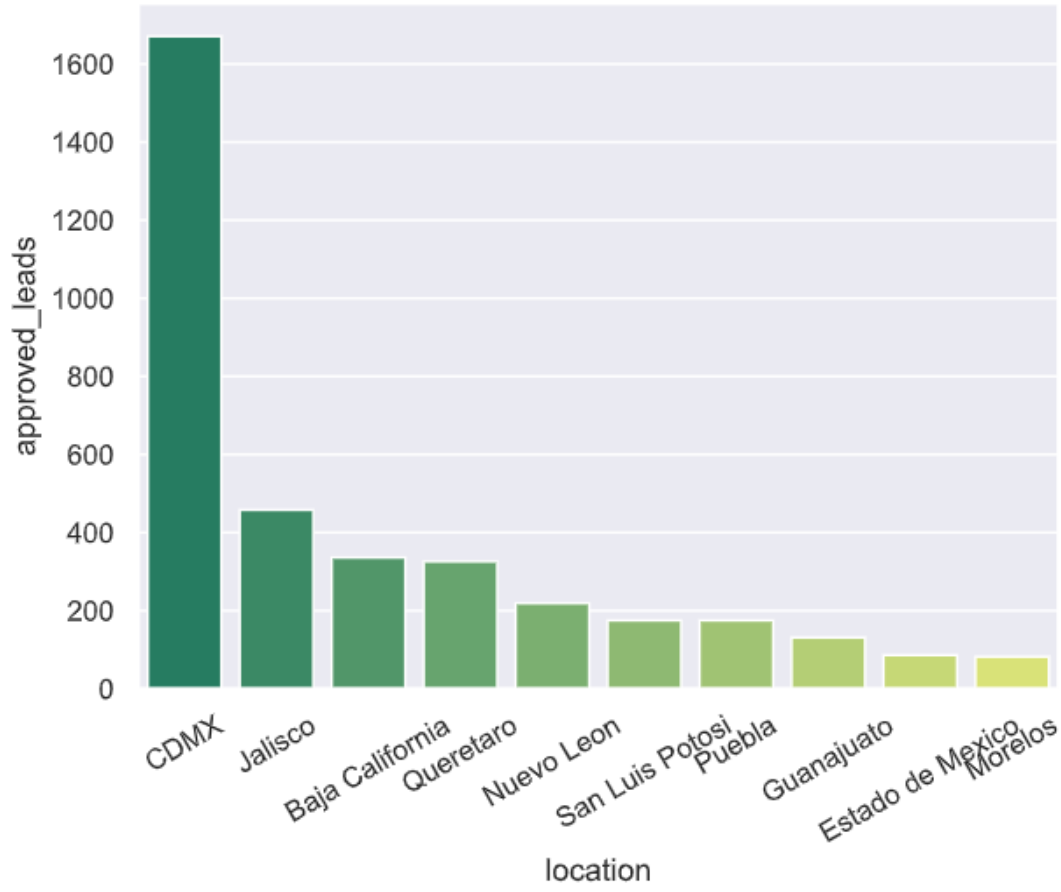
3. How much time takes an approved lead convert into a connected

BRIGHT. Leads Dashboard

Leads approved vs created	Leads Visited vs created	Leads visited vs approved
2.36 -2.64	14.78 +4.78	15.99 -34.01

Top 10 of locations with approved leads

Chart Data



What time takes a lead to convert a visited lead?

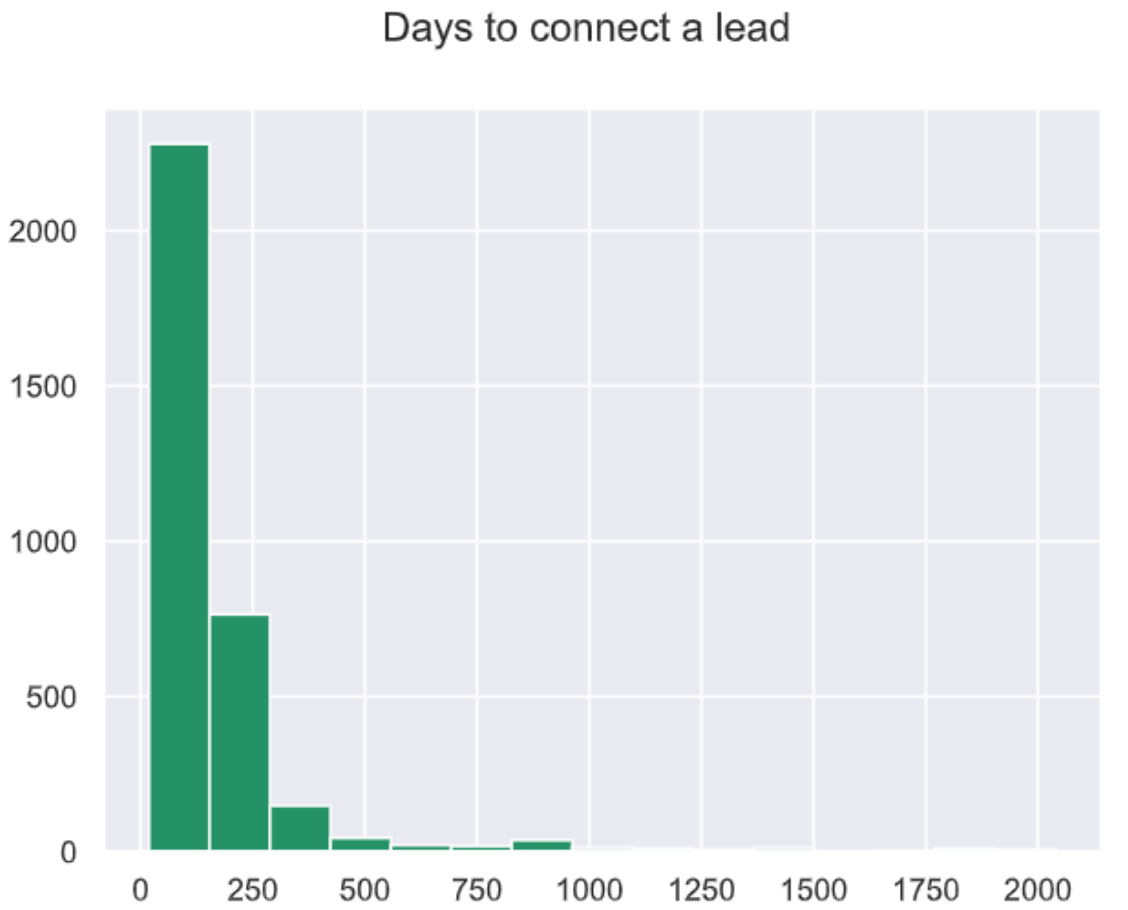
Days to visit - Max	Days to visit - Min	Days to visit - Avg
1769.0	0.0	41

Chart Data

What time takes an approved lead to be connected to CFE?

Days to connect - Max	Days to connect - Min	Days to connect - Avg
2040.0	19.0	175

Chart Data



Assumptions

There are many repeated lead_id values per event_type, except in lead.created type.

event_type	cuantos	diferentes	% repetidos
lead.created	176229	176229	0
doc.salesVisitReport.uploaded	26041	23494	9.78
doc.signerCredit.approved	4629	4321	6.65
doc.subscriptionContract.uploaded	7449	4976	33.2
doc.subscriptionContract.approved	4165	4121	1.06
doc.installationComplete.uploaded	4588	3563	22.34
doc.interconnection.approved	3743	3379	9.72

Is clear there are rules to manage the repeated records in the database, however in order to know the time that a lead takes to move from one state to another I just considered the created_at's field maximum value of the repeated leads set.

It is the same case for outliers values for created_at field. The maximum value in days for a lead convert from created lead to a visited lead is 1814 days and to convert from an approved lead to a connected lead is 2404. In both cases I don't have the rule to consider those values as incorrect ones.

The project is available in https://github.com/hmorales21/dashboard

Horacio Morales
August; 2023

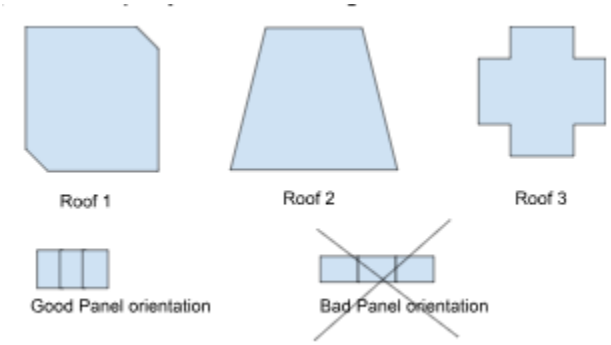
Problem 2

We want to add a “Panel Placing Tool” (PPT) to the existing site selection tool. This PPT’s function would be to maximize the number of vertically placed panels that you can install on any roof shape. The panels are always the same size, shape, and vertically oriented.

Instructions: Explain step by step how you would fit and calculate the maximum amount of panels that can be placed on the selected area. What assumptions you would make and why. What other information would you need from the final user.

NOTE: we’re looking for a path to solve the problem, not the actual solution. There’s not one solution but many possibilities, and we’d be looking more at your approach on solving the challenge than the actual solution, so please be descriptive on how you got there

Example: The input you have is a figure selected like the ones below



Response

1. I would reduce the roof surface in order to obtain a regular shape as a rectangle or a square.
2. then I would divide the length of the resultant surface between the size of a panel

3. The results should be the number of panels that you can install on the roof.

Another approach would be the next:

1. To Reduce the roof surface to obtain all available regular shapes.
2. Calculate the number of panels, according to the previous process, for each regular shape individually.
3. Sum the number of available panels for every regular shape obtained in the previous step.

Problem 3

SQL sentences:

```
-- Horacio Morales González

-- Problem 1

-- Get how many classes are per department


select distinct data->'departamento'
    ,count(distinct title) as hoy_many_classes
from classes
group by data->'departamento';


-- Problem 2

-- Get the number of months each student has been in college until today.

select name
    ,current_date
```

```
    ,college_enrollment_date
    ,extract(YEAR FROM age(current_date ,college_enrollment_date))*12 +
extract(MONTH FROM age (current_date, college_enrollment_date)) as
num_months
from students;



-- Problem 3
-- Get the list of students (id and name). That is enrolled on ?Matematicas II?

Select student_id
    ,name
from students
WHERE 102 = ANY(enrolled_classes)
    or 202 = ANY(enrolled_classes);

-- Problem 4
-- Get the list of all classes that have surpassed the maximum capacity of the class

with cupo_por_clase as (select A.class_id
    ,A.title
    ,cast(A.data->>'max_cupo' as integer) as cupo
    ,count(B.student_id) over (PARTITION BY A.class_id) as how_many
```

```
from classes A left join students B on A.class_id = any(B.enrolled_classes)
) select distinct title
    ,cupo
    ,how_many
from cupo_por_clase
where how_many > cupo;


-- Problem 5
-- Get the list of professors that haven't submitted one or more grades.
select professor
from classes where class_id not in (select distinct class_id
                                    from grades A);


-- Problem 6
-- Explain the order in which prostgresql will run each section of the query and show what will be the final result.
explain
select
    students.name
    ,classes.title
    ,pass.result
```

```
from grades
    left join students using (student_id)
    left join classes using (class_id)
cross join lateral(
            select
                    case
                            when grades.grade>= 7 then 'pass'
                            else 'fail'
                    end as result
            ) as pass;


-- EXECUTION ORDER
--  1) extract records from grades and the first join with students to get "name" field
--  2) extract records from classes in the second join to get the "title" field
--  3) create the lateral combination, every record in grades, with every record in subquery to get the action's name  pass or field according with the student's grade.
--result:
--Juan    Matematicas I       pass
--Juan    Matematicas II      fail
--Juan    Fisica  fail
--Pedro   Laboratorio quimica fail
--Pedro   Quimica Avanzada  pass
```

--Ana Literatura pass

--Ana Laboratorio quimica pass

--Ana Matematicas II fail

--Ana ?tica fail

--Silvia Fisica pass

--Silvia ?tica pass

--Silvia Laboratorio quimica pass

--Gerardo Laboratorio quimica pass