Analytics engineer take-home

Notes on submission:

The exercise has three parts: an analytics platform exercise, an analytics exercise, and a series of SQL questions. You have a week to finish the three exercises and send them back to us. Once we review it, we will set up time with a group of Justers for you to walk us through your solution. Some notes on your submission:

- **Show us your work -** Please provide us with enough information (code, writeup, notebooks, SQL fiddles…) so that we can understand your thought process and work.

- **Bring your preferred tools -** You can use whichever tools and format you feel more comfortable with, as long as we have access to the code to reproduce your results, and your findings and conclusions are well documented.

- **Quality over quantity -** We want to be respectful of your time, and know from experience that it is very easy for a data scientist to get carried away in an exciting new data set or problem. Please don't spend more than 5 hours on the assignment. We will grade knowing of the time limits, and incomplete assignments can still get a very good grade if the quality of the work merits it.

1. Analytics Exercise

Introduction

One of the most important steps in setting up a new insurance company are:
1) building its first pricing model (predicting a potential customer's future claims), and
2) deciding what its target market will be.

In this assignment, we ask you to leverage publicly available data to generate some recommendations about the second one, focusing on the existing premiums to spot arbitrage opportunities.

Dataset

You will be using an extract of publicly available data from the AUTOSEG ("Automobile Statistics System") database, provided by Brazil's Superintendence of Private Insurance (*Superintendência de Seguros Privados,* SUSEP). AUTOSEG compiles and maintains policy-characteristics-level data for personal auto from 2007 through the present for all insured vehicles in Brazil.

For your convenience, we have prepared an extract of the data in gzipped parquet format that you can access here, corresponding to policies that were in force for the year 2018 covering collision damages.

The dataset contains a variety of variables, from policyholder characteristics to losses by claim type. Each record contains the following variables:

1. *policy_id* - Unique ID of the policy.
2. *policy_start_date* - Start date of the policy's coverage.
3. *policy_exposure_days* - Days during the observation period that the policy was in force.
4. *policy_premium_received_brl* - Total premium collected by the insurance company during the period.
5. *policy_claims_num_reported* - Number of claims reported during the period.
6. *policy_claims_num_paid* - Number of claims paid during the period.
7. *policy_claims_total_amount_paid_brl* - Total amount paid by the insurer on all the paid claims.
8. *policy_holder_birth_date* - Birth date of the policy holder
9. *policy_holder_gender* - Gender of the policy holder (M for male, F for female)
10. *policy_holder_residence_city* - Policyholder residence's city.
11. *policy_holder_residence_region* - Policyholder residence's region.
12. *policy_holder_zipcode* - Zip code of the policy holder
13. *policy_holder_residence_latitude* - Latitude of the policyholder's zip code
14. *policy_holder_residence_longitude* - Longitude of the policyholder's zip code
15. *policy_holder_bonus_clas* - No-claim bonus of the policy holder (0 for none, 1 for 1 year without claims, and so on, until it reaches 9 years)
16. *vehicle_brand* - Car make of the insured vehicle.
17. *vehicle_model* - Model of the insured vehicle.
18. *vehicle_make_year* - Model year of the insured vehicle.
19. *vehicle_tarif_class* - Type of insured vehicle.
20. *vehicle_value_brl* - Estimated value of the insured vehicle at the policy start date.

Exercise

Now that you have downloaded the data, this is what we would want you to do with it:

1. **Competitive analysis:** One interesting feature we can compute from AUTOSEG's data is the *loss ratio*, that is, the amount of the received premium repaid to users to cover claims. A *loss ratio* close to 100% may indicate that the competition is underpricing risk in that segment, and it's likely losing money; on the other hand, a low *loss ratio* may indicate that the competition is overpricing risk in that segment, and this could signal a good opportunity for Justo to enter that market at a lower price. Bear in mind that because the collected premium is the commercial one (including taxes, claim handling expenses, profit for the insurer, …), and because some claims take time to be reported and/or closed, the average loss ratio in the dataset will be considerably lower than 100%. Your task is to use exploratory data analysis techniques to describe the loss ratio of different demographics, geographies, and other policy features.

2. Analytics platform exercise

You are Justos second analytics engineer. When you arrive, Justos has a Looker instance with a LookML model in development, a product transactional database in AWS Aurora, Segment as the CDP (with a Redshift backend). The dimensional model is being built using DBT/Redshift. Your job would be focusing on Looker/Amplitude and Segment:

1) Describe best practices when modeling in Looker.
2) What's the difference between Amplitude and Looker?
3) How would you engage business users to use Looker?
4) How would you handle governance in Looker / Segment / Amplitude?

3. SQL questions

1. Given table *T_A* with a varchar field "*Customer_ID*", write a script that gives you the values of "*Customer_ID*" that are non-unique.
2. Given tables *T_A* and *T_B*, both with a varchar field "*Customer_ID*", write a script that finds the elements in *T_A* that are not in *T_B*.
3. Given table *T_A* that has fields "*Customer_ID*", "*Claim_type*", "*Claim_value*", "*Claim_timestamp*" write a script that creates a new table, *T_A_last10*, where you store the sum-total of the last 10 paid claims ("*Claim_value*">0) for each customer and claim type.
4. Given table *T_A* that has fields "*Customer_ID*", "*Claim_type*", "*Claim_value*", "*Claim_timestamp*" write a script that creates a new table, *T_A_Claim_Value_With_Trend*, that adds a new field representing the percentage increase/decrease in claim value compared to the preceding one for each customer and claim type.
5. Given table *T_A* that has fields "*Customer_ID*", "*Trip_Score*", "*Trip_km*", "*Trip_timestamp*" write a script that creates a new table, "*T_A_Trip_Score_Rolling_Average_Last80km*", that adds a new field named "*Score_RAvg_Last80km*" representing the distance-weighted average score for the last 80 km the user has driven up to (and including) the last trip, rounded to second decimal point.
    a. Worked example:

Customer_ID	*Trip_Score*	*Trip_Km*	*Trip_Timestamp*	*Score_RAvg_Last80km*
1	80	45	2300000	**84,37**
1	90	35	1500000	**74,00**
1	60	40	1000000	**60,00**
2	83	80	1700000	**83,00**
2	53	10	1000000	**45,12**
2	43	60	900000	**44,40**
2	50	15	800000	**50,00**

3	90	60	3550000	**88,75**
3	85	60	1800000	**81,25**
3	70	50	1500000	**70,00**