Data Science Quiz – Ramy Ghorayeb

1. How would you explain the bias-variance tradeoff in simple terms?

If a model is too simple, it will not fit the data accurately (we say it is underfitting, meaning it has a high bias).
If a model is too complex, it may fit the training data accurately but will not generalize well to new data (we say it is overfitting, meaning it has a high variance).

The Bias-Variance tradeoff means finding the right balance between fitting data correctly (no underfitting) and sticking too much on the training set. (overfitting)

2. How would you predict revenue tomorrow given daily revenue from the past

Do I only have the daily revenues as input data?

Case 1: Only the daily revenues

What I would do is to consider this as a Time Series problem. I will try to model the time series by a SARIMA model, with its trends and seasonalities and look at whether the residuals can be considered as white noise, and eventually come up with a more sophisticated model.

Case 2: I can look for more data

First, I would try to see the relevant teams to ask questions and gather as much data as I can about these revenues to understand better what composed the revenues:

Games:

- What are the revenues by games category?
- What are the revenues by game franchise?
- What are the revenues by platform? If there is a new gaming console launch, what is the average revenue per game per launch?

Given Ubisoft's game calendar and gaming console calendar, I can then have a better idea of how the revenues will grow.

Customers:

- What are the revenues per country?
- What are the revenues per age?

I can also look for less directly related data like how a country's economy is doing, or how the consumers buying power is evolving over time.

Then, I would try to build a multivariate model that takes into account all these different data points (after removing the too correlated features of course)

3. What is R^2 ? What are some other metrics that could be better than R^2 and why?

R^2 it the coefficient of determination. It is a metric evaluating regression models. The formula is:

$$R^2 = SSres / SStot$$

SSres (residual sum of square): how much of the variation the model did not explain
SStot (total sum of square): how much the dataset varies around a central number

Measure the residual does not mean anything in absolute. We need to take into relative to how much variation the dataset originally has

Other metrics are:

Root Mean Squared Error (RMSE) measuring the squared error of each prediction. It is a suited metric when we want to emphasize the large errors (thanks to the square)
Mean Absolute Error (MAE): measuring the absolute error of each prediction. It is a suited metric when we want robustness against outliers

5. How would you build a model to predict when a player will churn (stop playing a given game)? How would you define this flag? Which features do you expect to have and would you build? Which models would you try training? What are the expected business use cases of such a model? You can make extensive use of the results of previous questions.

I can build a model relying on when players like this one churn in average.

- Cluster the player by their profile (how much in average they play video games per day per month) and look at the specific group of a player
- Filter the group to look at the people that played this specific game
- Segment the players based on when they started playing the game. If it is not available or inferable, I can simply use the date the player installed it.
- I would filter the players that didn't really played the game (aka the players that churned after the average playtime to complete the tutorial or first mission)
- Look at what is the average churn per playtime or number of session if it is available and more suited.
- If a player reach a critical time (for example when 75% of people have churned) and his last connection date is unusual (for example when it reach the maximum time between two connection dates of all his games), I can send him a notification about the game to remind him to play (for example, an achievement he still hasn't unlocked, or some news about the franchise).

This model enables Ubisoft to increase the stickiness of the customer with the brand, but can be dangerous as players could be spammed to be reminded of playing a game they stopped liking..

I can also build a model based on what makes people churn in a game:

- Cluster the players by their profile (taking into account how much they play and if they game to explore, or PVP for example)
- Filter the people that didn't really played the game like previously
- Look at what are the steps in the game that cause spikes in the churn (for example, a specific mission, or rank in online multiplayer)
- When the players reach the specific mission, he receives an in-game notification helping him go past this mission more easily in order for him to not get bored.

This model enables Ubisoft to have first a better understanding of what are the pain points of the game for the player and correct them with an update, or simply design differently the next games, leading to an increased customer satisfaction.
It would also enable Ubisoft to have a better understanding of what this specific player likes to see and do in a game, and what makes him uncomfortable. Ubisoft will then be able to push to him better targeted new games, leading to a revenue increase..

By combining both models, an interesting application would be to know when the right time for Ubisoft is to push a DLC offer to the player, because it would know that the player stops playing the game because he is bored of the content, while not being bored of the game itself and not having completed the game. Ubisoft would then be able to increase its revenues.