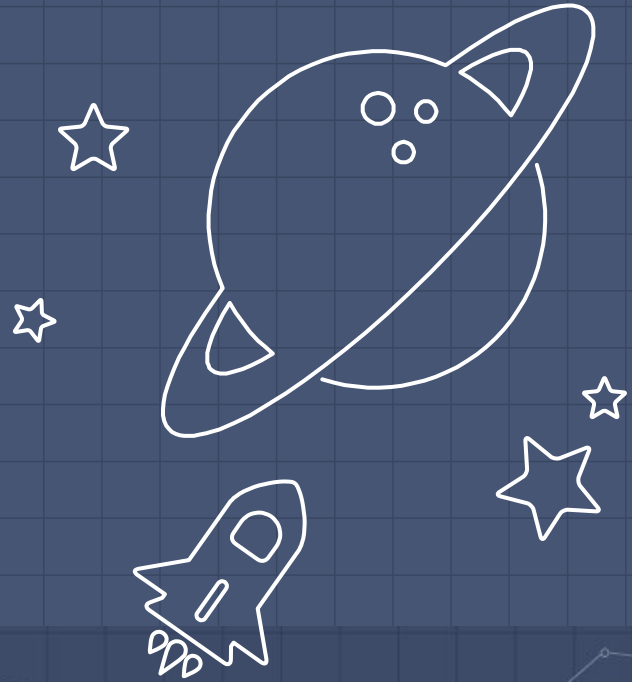
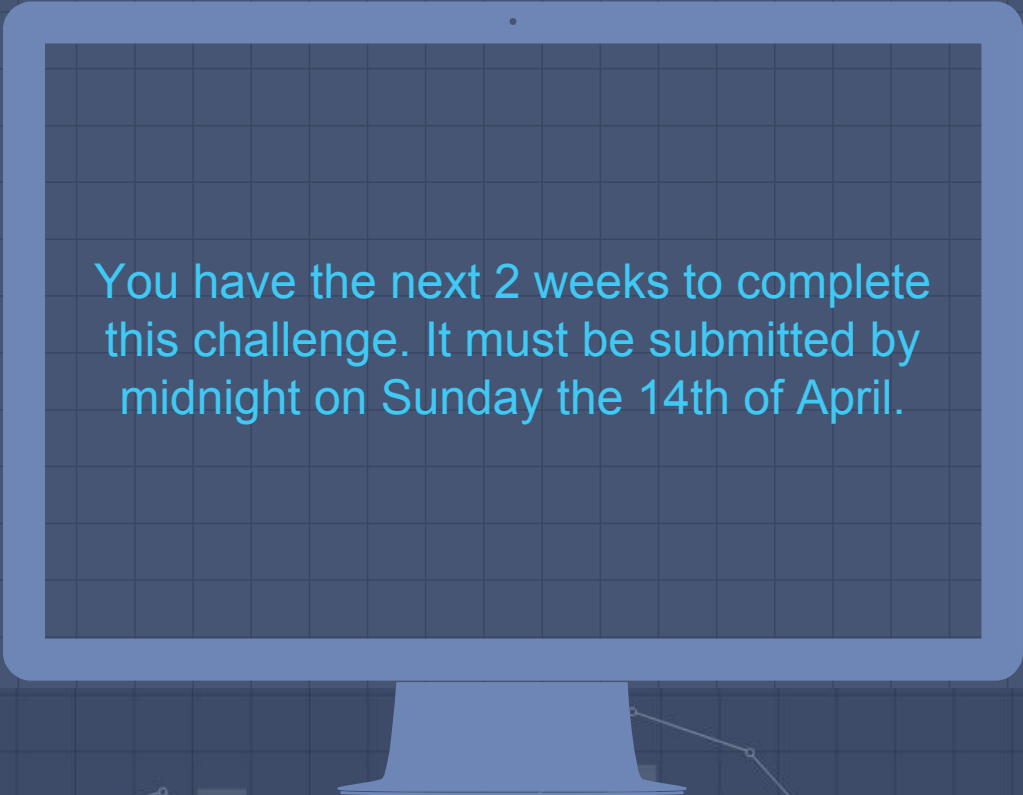


arise:

Final Challenge





You have the next 2 weeks to complete
this challenge. It must be submitted by
midnight on Sunday the 14th of April.

Introduction



This two week challenge is going to test your full capabilities as a data scientist. You are going to be required to wrangle a dataset using Bigquery, develop two loan pre-delinquency models, comprehensively compare the two models, and explain which one you would implement.

A pre-delinquency model is a model that is triggered after a loan is disbursed. The model uses features developed after loan disbursement to predict whether or not a loan will be defaulted on or paid back.

You are going to build both a random forest classifier and a neural network. Both models should be trained and tested on the same data. You will then have to properly compare the models.

The data that will be used for training and testing the models sits in Bigquery, but you will have to perform some transformations and manipulations to wrangle the data correctly. This is explained on the next slide.

SQL Section



In the ``propane-highway-202915`` dataset on Bigquery you will see two tables. The first table is titled ``ChallengeLoanInfo``. This table contains all loans that were disbursed in 2018 and have a due date within 2018.

There are some unwanted loans in this dataset. We only want to use loans in our model where:

- The loan was actually paid out (use `payout_status`)
- The loan is a Paylater loan (use `loanType`)
- The loan has a term of 60, 90 or 180 days (use `loanTerm`)

Using a where clause, keep only the loans that match the above cases. After you have removed the unwanted loans, you should be left with 159,596 loans.

We only want to use loans that were paid out and that were due within 2018, as we need to know if they were defaulted on or not in order to create a target variable for our models.

SQL Section



The second table in the `propane-highway-202915` is titled `ChallengeRepaymentsInfo`. This table contains the loan repayment information for all Paylater loans. This data is on an instalment level and as a result the loans will appear more than once in this dataset.

You need to isolate the **FIRST** payment of each of the loan. We are going to use this first payment in our later models. You can isolate the first payments using loanId, the `min` function and a group by. You should store this in a temporary table. The first payment is the payment with the earliest `dueDate` attached to the loan.

The next step is to join the temporary table you created above to the `ChallengeRepaymentsInfo` table. You can join the two tables on loanId and where `dueDate` matches the minimum `dueDate` you have isolated for that respective loanId.

You can now select all variables associated with the first payment of each loan. You now need to create a variable called `firstPaymentDefault` for each loan. If the `settleDays` variable associated with that first payment is greater than 7 or is null, that means the payment was made more than 7 days after the `dueDate` or the payment or not paid at all. When `settleDays` is greater than 7, or is null set `firstPaymentDefault` to 1, else set it to 0.

SQL Section



You now need to join the two datasets together. This will require you to ensure that the where clause written for `ChallengeLoanInfo` table holds and that the `firstPaymentDefault` variable is correctly joined for each loan. There should be no repeat loans in the final joined dataset. You will also be able to bring in the `paymentRatio` for first payment of each loan, but rename the variable `firstPaymentRatio`.


At this point you should still be left with 159,596 loans, each with two variables associated to their first repayments (`firstPaymentDefault`, `firstPaymentRatio`).

The final step in the wrangling process is to create the target variable, this variable should be called `loanDefault`. The variable needs to be created by using the `dueDate` and `paidAt` variables from the `ChallengeLoanInfo` table. For each loan, perform a date difference calculation between the `paidAt` and `dueDate` variables (in that order). If the difference is greater than 7 or if it is null (meaning no repayment was made) then set the default to 1, else set it to 0.

Exporting the Data



To export the data, you will need to log onto Superquery [here](#). Superquery is a querying tool that allows you to download a larger amount of data than Bigquery. Your query will produce too much data to be downloaded straight from Bigquery. You need to use the gmail address that you having been using for the program to get access into Superquery.

Once you have logged onto Superquery, you will see the same project folder that was shown in Bigquery. Copy and paste your query from Bigquery into Superquery. Then run the query by hitting the `superQuery` button. Once the query results are shown, simply click the  button and then select the CSV option. The file will then prepare to download. Once the preparation has completed, select `Download` and the csv file containing the data will download to your local downloads folder.

Models



We have our data and our target variable. The next step is to build the models. You are required to build a random forest and neural network, that both perform a binary classification on each loan. The classification is where or not a loan will be defaulted on (1) or not (0).

The models have to be built in [Google Colab](#) and must be written in Python. You are allowed to use any packages that you would like to in order to build these models (blow us away). Google Colab is a free tool that provides you with access to a free GPU. This will mean that computation will be easier. Please share both of your models with devon@paylater.ai. You are to share them on the platform and via email.

Please comment in your code as much as possible. Really explain what you are doing at each step. You will have to handle missing values, one-hot-encode categorical data, balance the classes and split your data into a training and testing set. These steps should be explained in detail in your code.

You are required to tune the hyper parameters of both models and explain your tuning methods.

Comparing Models



Once your models have been training and tested, you will have to comprehensively compare them and explain which is best. This explanation is up to you, but please include accuracy, precision, recall and F1-scores. You are to compare how both models performed at a testing and a training level.



Submission



You are to share your Google Colab projects for each model with devon@paylater.ai. Share your projects in Google Colab and platform and via a physical email. The subject of the email should be `final project`. You have to submit by midnight on Sunday the 14th of April.

Interviews



The next step in the process is to book an interview slot. Not everyone will reach the interview stage. You will be selected based on your performance in the weekly challenges and in the final project. You will be made aware if you have been selected for an interview within the next week.

