

# Even Data Science Code Challenge

Founded in 2015, EVEN Financial, an NYC based FinTech company, is the leading search, comparison and recommendation engine for financial services. EVEN provides essential infrastructure for financial institutions and channel partners, connecting consumers with personalized products (loans, deposits, credit cards and more), utilizing its robust yet simple API. A fundamental part of what makes EVEN a powerful recommendation engine is data science and engineering. By efficiently collecting and analyzing data, we provide real time offer recommendations with statistical algorithms.

The goal of this challenge is to create a **model server**, a web app that serves ML model predictions via a REST API, which a user can request by providing feature data.

## ***The Data***

The data is a sample of leads that are applying for a **personal loan**. They receive offers from some **lenders**. A lead may **click** on an offer, and we're interested in predicting the probability of this event.

The data is split between three tables. We have the following identifiers:

1. **lead\_uuid**: Unique person applying for loan offers
2. **offer\_id**: Unique offer. A lead may receive *multiple* offers
3. **lender\_id**: Unique lender. Every offer has *one* lender. A lender may give many offers

Besides these identifiers, we have the following *feature data*:

1. **requested**: The amount of money a lead is requesting
2. **loan\_purpose**: A lead's intended usage of funds
3. **credit**: A lead's self-reported credit bucket
4. **annual\_income**: lead's self-reported annual income
5. **apr**: A loan offer's APR (annual percentage rate)

Finally, we record an offer's **clicked\_at** timestamp, if it exists.

## ***The Challenge***

Please write the code to perform the following tasks (we ask that you use **Python**, unless otherwise specified):

1. Define a schema for each table, create these tables in a relational SQL database.  
Please use an open source database, such as SQLite, Postgres, or MySQL.
2. Load the data into the database tables you've designed, using Python.

3. Write code that will read the data from the database, do any necessary processing (joining tables, feature pre-processing, etc) and train a **logistic regression** on the features described above. Your target variable is whether or not an offer is **clicked**.
4. Create a simple web application that has **two** endpoints that receive the following:
  - a. A POST request that takes a JSON body of feature data and produces a **probability of click** for a single offer and lead.
  - b. Another POST request that takes a JSON body of feature data for **multiple offers** for a lead and produces a list of **probabilities, one for each offer**.

For the project deliverable, please ensure to include the following:

1. Detailed instructions for setting up the environment (pip, sbt, maven, etc), including libraries/dependencies, and any setup scripts needed.
2. Please ensure any necessary installation steps are documented.
3. If desired, a Github link is also acceptable.
4. Provide SQL, Python, and model files as needed to reproduce the app locally.

An important requirement is that this project ***should be completely reproducible*** in a Linux/Mac environment.

---

**Download data from these links:**

[ds\\_leads.parquet.gzip](#)

[ds\\_clicks.parquet.gzip](#)

[ds\\_offers.parquet.gzip](#)