Scope of Work for Data Science Project

Version 1.1, 2017-02-01

Prepared by	Alice Smith, alice.smith@fas.harvard.edu Bob Doe, robert.doe@fas.harvard.edu Charlie Brown, charlie.brown@fas.harvard.edu
Prepared for	Dana Client, dana@example.com
Summary of changes	• Version 1.1, 2017-02-01, Add web service to deliverables per phone conversation • Version 1.0, 2017-01-13, Initial draft

Background

Example Loan Ltd ("the Client") is a microfinancial institution (MFI) that makes thousands of microloans each month to people in the developing world. They have been making these loans for four years, and have been using roughly the same go/no go decision flowchart for the past two years.

These loans, typically ranging from €100 to €2,000 in size and 2 months to 6 months in duration are used by the borrower to make small capital investments, for example to buy livestock or crop seeds which will then generate revenue, to receive professional training for licensure, or as capital investment for materials necessary to start a small business such as a mobile phone airtime stand.

Like many microloans, the loans that the Client makes are unsecured, which means that the borrower provides no collateral and therefore all of the counterparty risk is borne by the Client. Loan defaults, while rare, are the single most serious threat to the long term sustainability of the Client's business model. The Client believes based on industry research that their current borrower default rate of 3.2%, while not unsustainable, is high for the MFI industry.

The Client has requested that the students listed above ("the Team") provide a way to incorporate predictive modeling into their approval process in order to make more informed lending decisions.

Problem statement

Goal

Using past data on loan applications and outcomes, optimize the Client's lending decision model for whether to **lend** or **not lend** to improve the current default rate by at least 70 basis points (≤ 2.5%). They also want a way to system which automatically runs new applications through the model and displays this recommendation to loan officers along with justification for decision in a user friendly way.

If time permits, an even more in depth solution might assess applications which are classified be rejected to see if a lower loan amount should be offered. Client is also interested in exploring leading indicators that active loans may start becoming "at risk" in order to intervene.

Note: Client needs any software or libraries used must have licenses that permit client to use software without limitation for any purpose, per e-mail exchange of January 30, 2017.

Resources available

Data available includes:

- Over 12,000 records of loan applications, will be a structured JSON dump containing:
    - Which loan officer was assigned
    - Basic personal and financial information (e.g. age, gender, geographic location, number of children, history of previous loans from Client if any, self reported cash on hand, self reported loan history, forms of government identification confirmed by the lender, etc)
    - Requested loan amount (in EUR)
    - For a sample of loan applications (~4,000), a third party credit score service was used to get an additional indicator. This is a numeric value scaled between 200 and 900.
- Payment records for each approved loan, connected by primary key to the application, containing:
    - Timestamp payment received
    - Amount of payment
- Spreadsheet of all loan IDs and status of the loan according to Client's business rules (active, completed, delinquent, defaulted).
- Free entry text from the last 1 year of loan applications which are responses to a SMS questionnaire about what the loan will be for and why it is being requested.

Deliverables

The deliverables will be all necessary code, assets, and documentation necessary for the Client to run on their system fulfilling the following requirements:

Deliverable 1	Predictive **model** trained on past data which: ● Predicts probability of loan default ($p \in [0, 1]$) more accurately than the current non-probabilistic business rule flowchart ● Also select optimal decision threshold π such that $p \geq \pi$ indicates that a loan should be made. This will be built with scikit-learn.
Deliverable 2	Python **module** which: ● Can be run as a standalone script or imported programmatically into other parts of the Client's codebase ● Import the frozen model from part one and: ○ Given a new loan application and amount requested, predict probability of default ○ Based on π, indicate loan/no-loan decision ○ Provide some sort of interpretable feedback (e.g. a list of strings) indicating which inputs were largest factors in rejecting loan, and if possible a plain language note
Deliverable 3	Internal **web service** proof of concept which provides: ● API endpoint (JSON response) given an application primary key, gets the output from the model and serializes all of the decision information into a JSON payload ● Application list page (HTML response) show list of all pending applications with Loan Officer assigned ● Application detail page (HTML response): given an application primary key, can pull up the application details, the client's past record, and display output from the API endpoint in a clean and useful fashion This will be built on Flask with Bootstrap styling for HTML pages.
Deliverable 4	Brief **recommendations** on any data quality issues and what data might be useful to have going forward

[Note: It is okay if this changes along the way, but it should be completely filled out. Once the scope is approved, the TF should be consulted before changes are made.The most important part at any given time is the next milestones; the TF will be using these to help set expectations for what the team is working on and hoping to achieve.]

Project timeline

Sprint ending	**Tentative milestone or goal**
2017-02-07	● Receive data from client ● Preliminary data exploration, compile list of technical/data and business questions for Client ● Complete first draft of scope document and send to Client for review and approval ● Project set up ○ Private git repository created, TF and professor shared ○ Team communication channel (e.g. Slack) selected, TF added ○ Project management tool selected (e.g. Github projects, Trello, waffle.io, ScrumDo, etc.), TF added
2017-02-14	● Create an automated data cleaning pipeline ● Confirms team is convinced data on hand is adequate for the rest of the project ● Brief literature review ● Choose 3 supervised learning models to explore ● Jupyter notebook which clearly compares status quo to preliminary best model
2017-02-21	
2017-02-28	
2017-03-07	
2017-03-14	
2017-03-21	
2017-03-28	
2017-04-03	
2017-04-10	