Modeling Loan Quality & Approval Status

Vishnu Rao-Sharma - 001690174
Zehao Song - 001397052

Project Overview
What is our project?

- Machine-learning models
- PLAY Web App
- Data Visualization

We developed a web application that allows users to input loan applications and receive a grade and approval status based on our models' evaluations.

The app features informative graphs that explain the factors influencing the loan grade and approval decision, as well as a UI for grantors to efficiently manage multiple loan applications simultaneously

Goals of the Project

What we did accomplish

- Effectively predict the health and grades of a loan
  - Based off of this grade the system can further analyze it and provide end users with constructive feedbacks
- Produce informative visuals (in the form of graphs and charts) to better visualize the weight of each elements in the prediction phase, as well as summary on all the submitted loan applications on the Grantor Side.

Use Cases
Who would use these models/this app and why

- We have two user groups in mind
  - **Loan Grantors**
    - Inputs a candidate's profile, receives an estimated loan grade and approval status
  - **Loan Requesters**
    - Inputs their personal profile, receives an estimated loan grade and probability of an approval
    - Using SHAP scores or model coefficients, requesters can see which attributes of their profile matter the most and conceptualize how they might improve their likelihood of getting a loan approved

Use Cases (cont.)
Who would use these models/this app and why

Loan Grantors

- Input loan candidate details
  - Loan Status & Loan Grade predictions (& probability)
  - Retrieve 15 most similar candidates and display approval/grade statistics → Loan Dashboard with stats about graded applications

Loan Requesters

- Input personal details
  - Loan Status & Loan Grade predictions (& probability)
  - Display model information to provide requesters a glimpse into model weights

Methodology
How did it work out

- We successfully constructed two data pipelines and two distinct models that each predicts an aspect to the loan approval model.
- Multiple models were tested, and the one with the highest accuracy and correctness was selected.

Methodology (cont.)
How did it work out

- **Loan Status Model - Notebook**
  - Logistic Regression (❌)
    - 69% precision (scored using Kaggle competition test.csv)
  - Random Forest Classifier (✔)
    - 80% precision (scored using Kaggle competition test.csv)
    - Moved forward for 5-fold Cross-Validation
      - 93% Precision; 63% recall - a very selective model that has a bias for rejecting (which is preferred for our use case)
- **Loan Grade Model**
  - Gradient Boosting (42% precision on testing set, after cross validation went up to 44%)
  - MultiLayerPerceptron(46 - 47% precision on initial setup, after increasing the layer count and node count, was able to bump up the accuracy to a 50% with sacrifice to training time)

Data Source - Kaggle Link

A	B	C	D	E	F	G	H	I	J	K	L	M
id	person_age	person_income	person_home_o	person_emp_le	loan_intent	loan_grade	loan_amnt	loan_int_rate	loan_percent_in	cb_person_defa	cb_person_cred	loan_status
0	37	35000	RENT	0	EDUCATION	B	6000	11.49	0.17	N	14	0
1	22	56000	OWN	6	MEDICAL	C	4000	13.35	0.07	N	2	0
2	29	28800	OWN	8	PERSONAL	A	6000	8.9	0.21	N	10	0
3	30	70000	RENT	14	VENTURE	B	12000	11.11	0.17	N	5	0
4	22	60000	RENT	2	MEDICAL	A	6000	6.92	0.1	N	3	0
5	27	45000	RENT	2	VENTURE	A	9000	8.94	0.2	N	5	0
6	25	45000	MORTGAGE	9	EDUCATION	A	12000	6.54	0.27	N	3	0
7	21	20000	RENT	0	PERSONAL	C	2500	13.49	0.13	Y	3	0
8	37	69600	RENT	11	EDUCATION	D	5000	14.84	0.07	Y	11	0
9	35	110000	MORTGAGE	0	DEBTCONSOLI	C	15000	12.98	0.14	Y	6	0
10	30	78000	MORTGAGE	5	VENTURE	B	12800	10.59	0.17	N	5	0
11	22	33000	RENT	6	PERSONAL	B	10000	11.12	0.3	N	2	1
12	25	33000	MORTGAGE	1	EDUCATION	B	4000	10.75	0.12	N	3	0
13	31	70000	MORTGAGE	2	DEBTCONSOLI	B	16000	11.14	0.23	N	9	0

training.csv - 58.6K rows

test.csv -39.1K rows

Milestones

When we’re going to complete things

- 11/21, repository Setup, Development phase, nailing down the exact functionality of our applications, laying down some groundworks
- 11/28, Start feature engineering and model training, begin development on the side-functionality
- 12/5, Model fine tuning, integration with the web app, create and display visualizations
- 12/12, additional features, tying up loose ends

High-level Implementation Overview

What we will program in Scala vs. not

- Our code will be uploaded to a project Github Repo (https://github.com/vishnuraosharma/loan-quality)
  - Scala: Data processing and feature extraction, model training/testing and evaluation.
  - Play Framework: Used for creating web application, the design and UI will likely be done here.
  - ~~Python: Used for generating SHAP values and visualizations~~

Acceptance Criteria

How we will keep ourselves in check

- The model should process the data given and yield results in no more than 5 seconds. (✔️, 2-4 seconds taken to generate result on average)
- Retrieve and send data with 0 interruptions. (✔️)
- 100% errors to be handled, so that the program should be well-tested and crash-safe. (✔️)
- From the time a certain query is sent to retrieving a specific information should be less than 4 seconds. (✖️)
- Each model should achieve a precision of 65 percent. (Partially achieved. 80% Loan Status, 49% Loan Grade)

Questions?