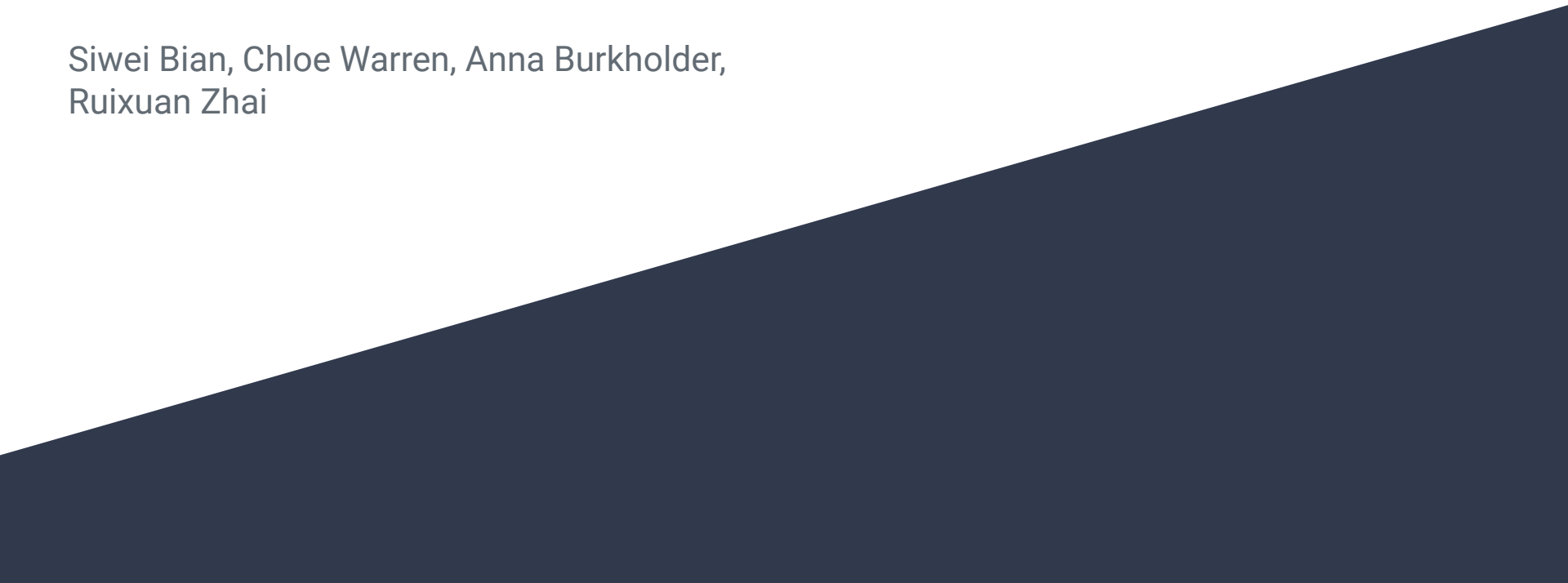


Housing Loan Prediction

Siwei Bian, Chloe Warren, Anna Burkholder,
Ruixuan Zhai

A dark blue diagonal gradient bar that starts from the bottom left and extends towards the top right, covering the lower half of the slide.

Self introduction

Group 14

Chloe Warren, Anna Burkholder, Siwei Bian, Emily Zhai

Problem that we work on

1. Loan prediction
2. Why is this issue relevant/useful to examine?
 - a. Eliminating bias
 - b. Improving efficiency

Deliverables breakdown

1. Dataset downloaded from <https://www.kaggle.com/burak3ergun/loan-data-set/version/1>
2. Accuracy goals
 - a. 65%,
 - b. 70%,
 - c. 90%
3. Skill application goals
 - a. functionality
 - b. Mixture of categorical and quantitative features
 - c. Feature engineering
4. Variation
 - a. Random forest with 1 other method
 - b. Two measures of accuracy
 - c. Deal with missing data differently

What we did

About our data:

1. Contains 614 unique loan candidates.
2. Has missing data.
3. Each sample has 12 features.
4. Very unbalanced result (y value).

What we did

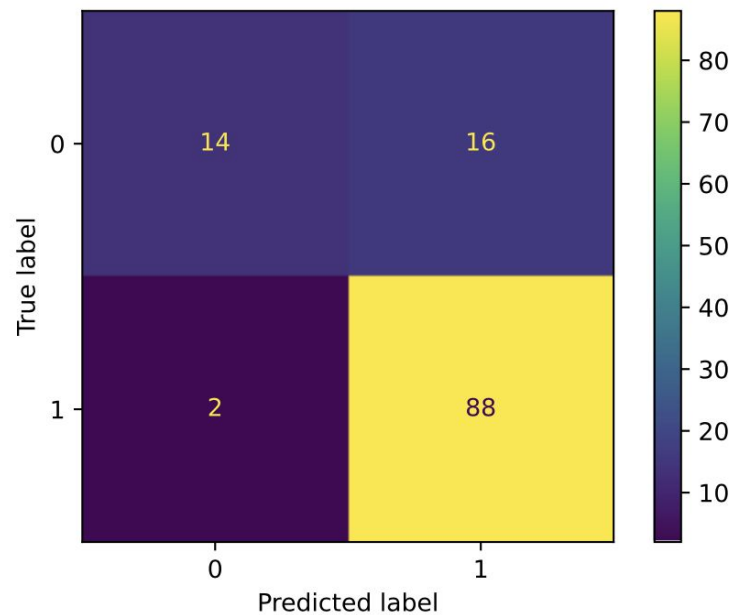
Preprocessing data:

1. Get rid of rows that have missing data. (down to 480 samples)
2. Using onehotencoder to encode non-numerical features (include 'Gender', 'Married', 'Dependents', 'Education', 'Self_Employed', 'Property_Area')
3. Get rid of “trivial” features (include 'Loan_ID', 'Loan_Amount_Term' and 'Credit_history')

What we did

Training using random forest:

Here is a confusion matrix of our initial run.



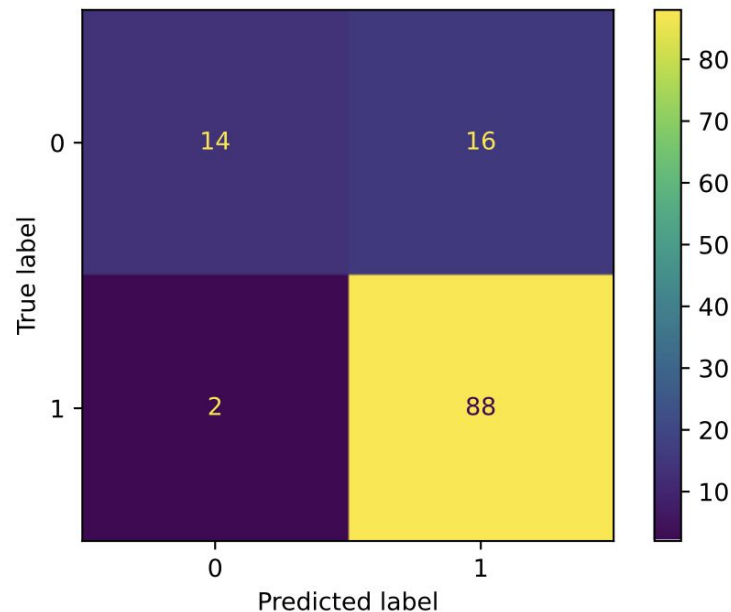
What we did

Training using random forest:

We see that there are lots of false positive cases.

Our data is really unbalanced with many cases that has result 1.

We used default loss function which doesn't apply different weights for classes.



What we did

Training using random forest:

Optimize hyperparameter choices using GridSearchCV.

We picked 'n_estimators', 'criterion', 'max_depth', 'class_weight', 'min_samples_split', 'max_features'.

N_estimators: Number of trees that are build before taking the max voting. (Higher number of trees will give better performance but slows down the code)

Criterion: Different optimization approach (either gini or entropy).

Max_depth: Max depth of the tree.

Class_weight: Assigning weights for each class (help solve the data unbalance problem).

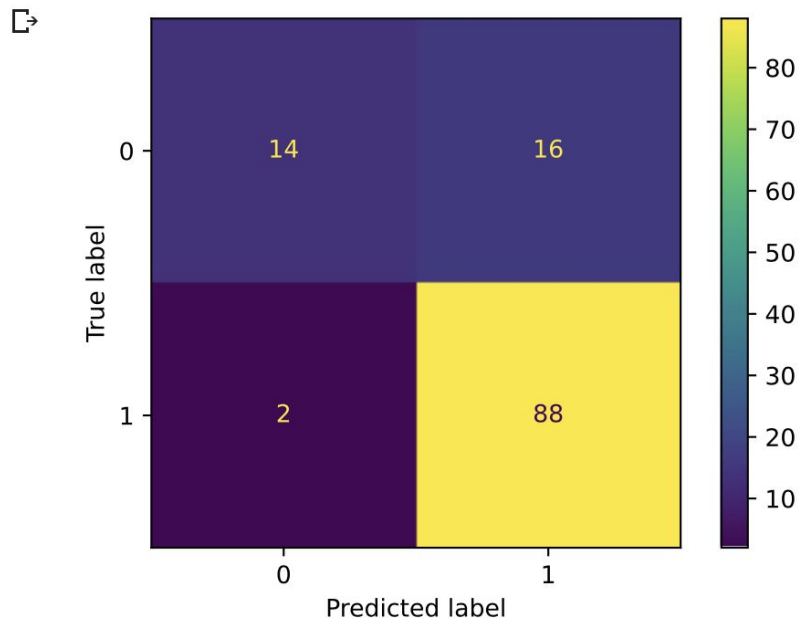
Min_samples_split: The minimum number of samples required to split an internal node. (Don't want our model to be overfitting).

Max_features: The maximum number of features Random Forest is allowed to try in individual tree.

What we did

Training using random forest:

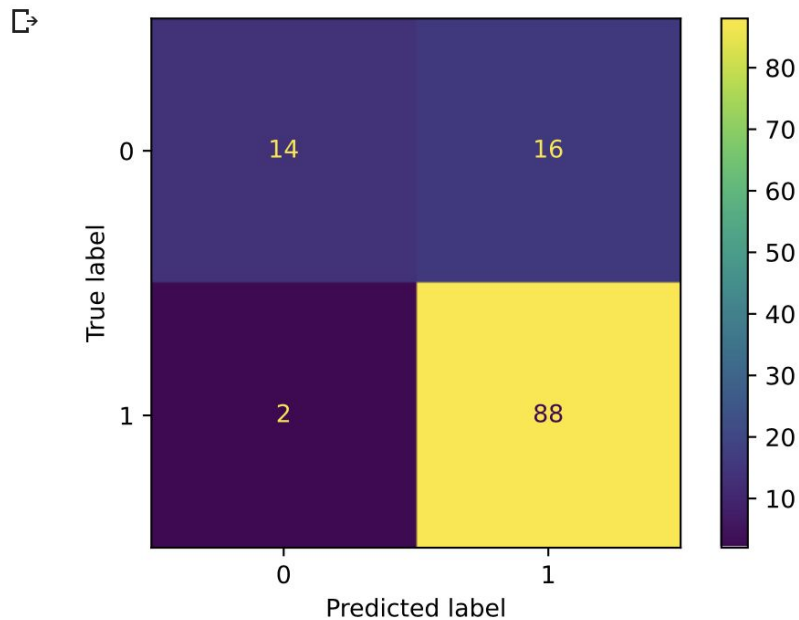
With the new hyperparameters we got, we run the model again and got the following result:



What we did

Training using random forest:

We do see some increment on training and validation set, however, the result is the same for testing set.



What we did

Training using random forest:

Our data is really limited (data set size is small) and random forest generally performs well.

Our first try might have already reached the “best” result we can get out of Random forest model.

Further changes doesn't make significant changes.

What we did

Training using random forest:

To further verify our thoughts, we tried other classifiers: gradientboosting, adaboost, svm, however, none of the models perform better.

What we did

Training using logistic regression:

Preliminary run with all default settings - test accuracy 85%

Tried different combinations of parameters:

- Solver setting: 'newton-cg', 'sublinear', 'saga' etc
- Max iterations (100 - 10000)
- Penalty norms: 'l1', 'l2', 'elasticnet'

Highest test accuracy achieved was still 85%

What we did

Dimensionality reduction:

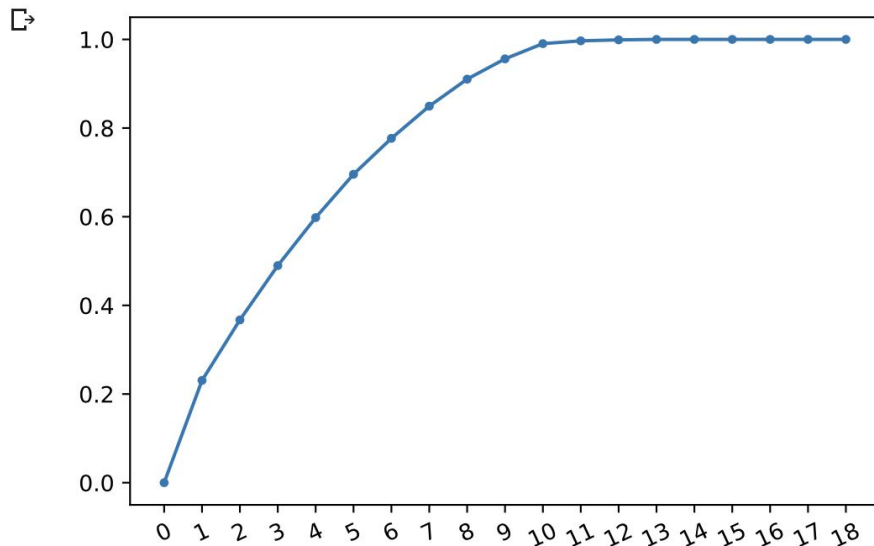
We then try to reduce the dimensionality (the number of features).

Use PCA (principle component analysis) to examine features and reduce number of features.

What we did

Dimensionality reduction:

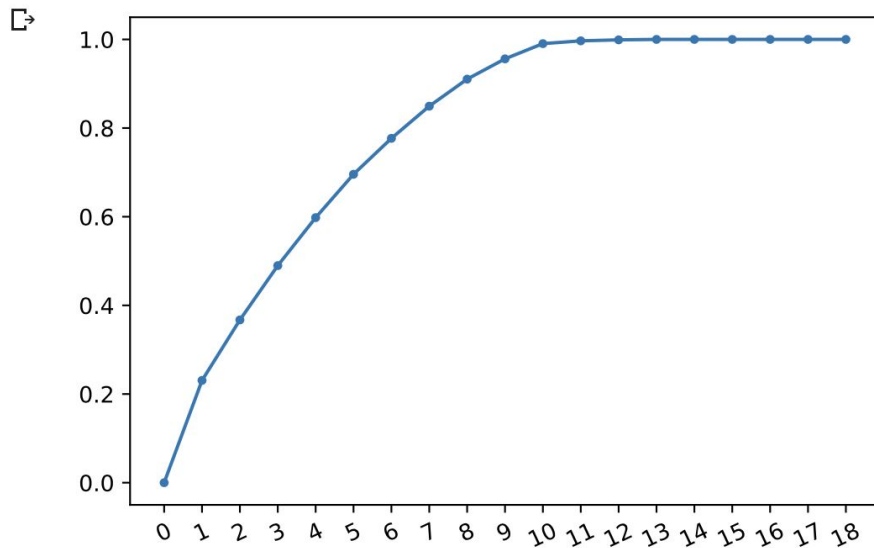
Out of 18 features, we might need only 11 of them.



What we did

Dimensionality reduction:

We then transform the training and validation sets (change them to 11 features, these 11 features come out of the previous features by doing some combination/calculation on them.)



What we did

Dimensionality reduction:

The results for both Logistic regression model and random forest model are not optimistic. (they even get lower)

What we accomplished

For both random forest and logistic regression, we get highest prediction accuracy to be 0.85.

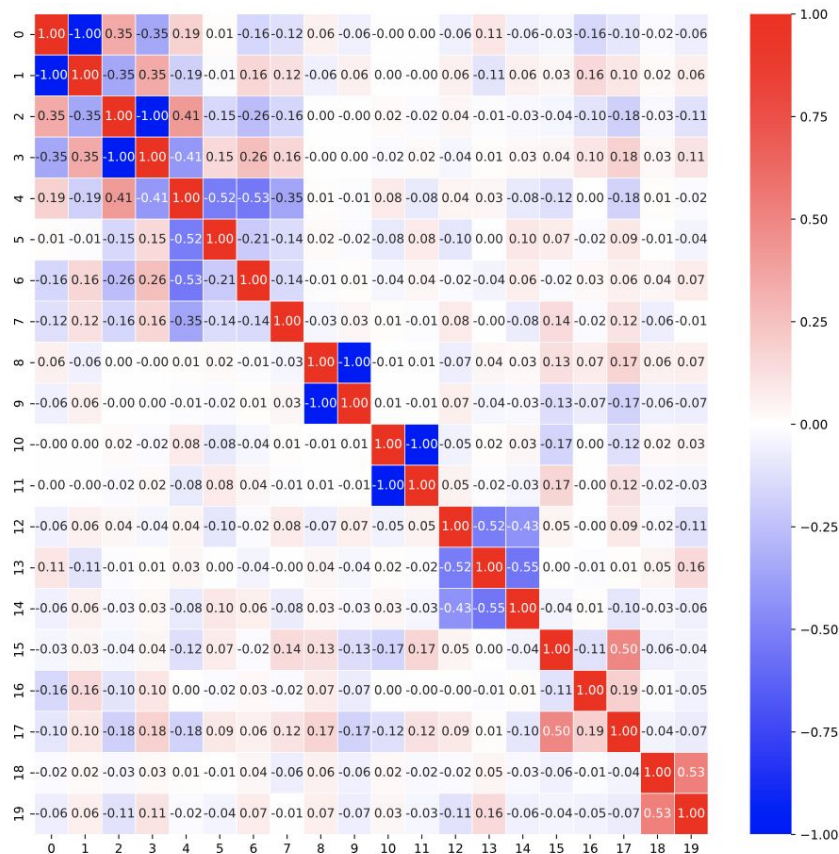
The obstacles

Data:

1. Our dataset is so limited that there isn't enough information for the models to learn from.
2. Our data is really unbalanced.
3. There are some weird behaviors of data.

The obstacles

Heat map



What to do in the future

1. Find larger and more balanced dataset.
2. Further shrink the number of features.
3. Dig into the relationship between features.