Predicting Credit Card Defaults

W207: Applied Machine Learning

Ryan Chen, Mehmet Inonu, Daniel Myers

Berkeley
UNIVERSITY OF CALIFORNIA

Dataset and EDA

Berkeley
UNIVERSITY OF CALIFORNIA

Default of Credit Card Clients

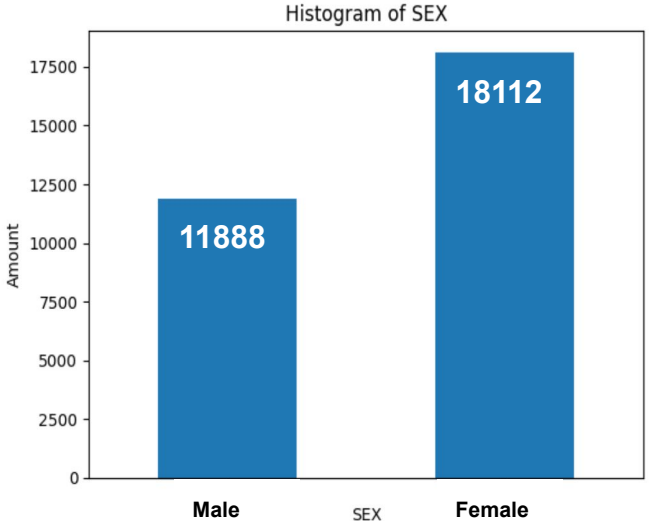
Contains demographic info, credit information, and history of payments and defaults of 30k credit card customers in Taiwan from April to September 2005.

Data source: UCI Machine Learning Repository

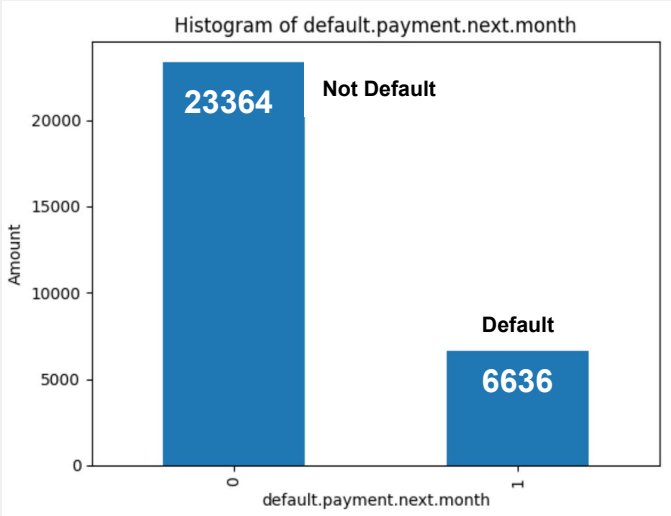
Variables (24 Total) and Description:

- **Client ID**
- **Gender**
- **Education**
- **Marital Status**
- **Age**
- **Credit Limit**
- **Repayment status**
  - -1=paid on time, 1=payment delay for one month, 2=two month delay, ... 8=eight month delay, 9=payment delay for nine months and above)
- **Amount of bill and payment amount made by month (12 total variables)**
- **Default status**
  - 1=default, 0=no default

Visualizing the data - Imbalance of categorical features



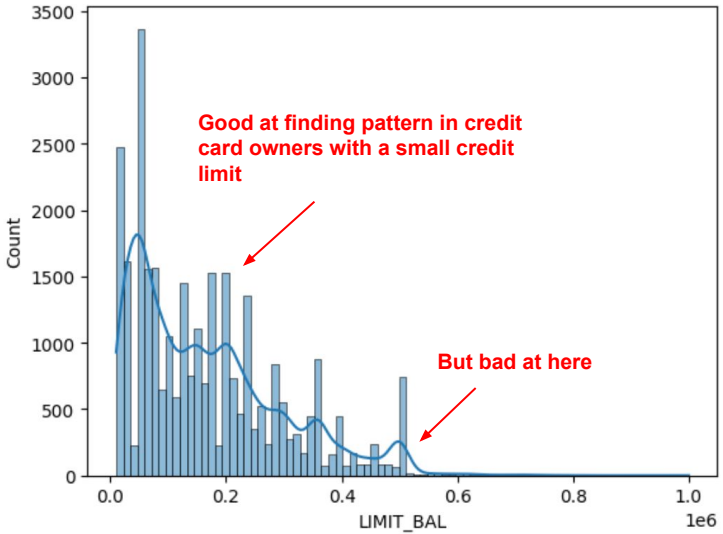
This may induce potential unfairness that the model performs better on Female than Male or that gender is a critical factor of prediction.



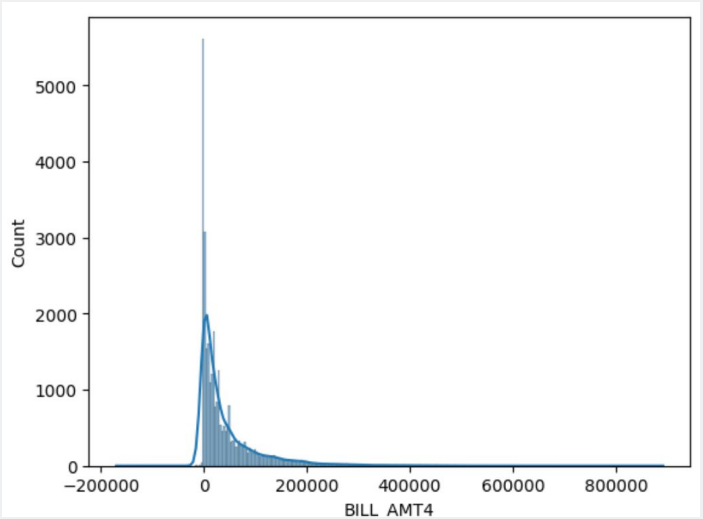
The imbalanced target data set might make our model only learn how to identify “non-default” (bad recall)

Goal: the data set according to the recall and precision of the model

Visualizing the data - Skewness of numerical features



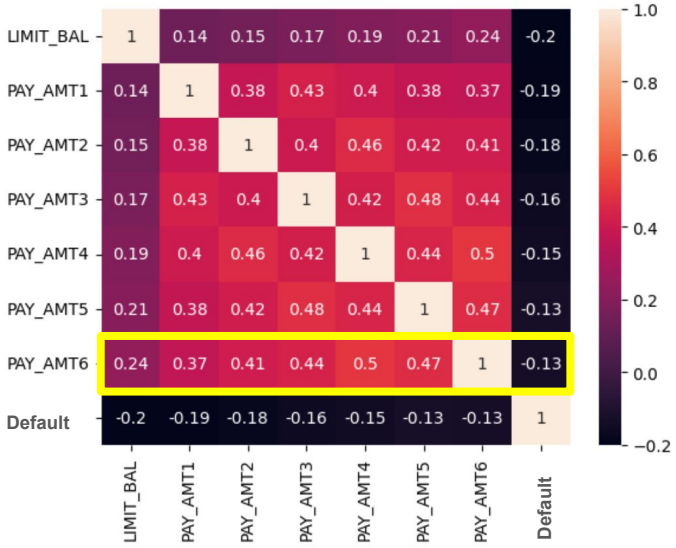
Our model may only learn the left part of the data set with severe right skewness

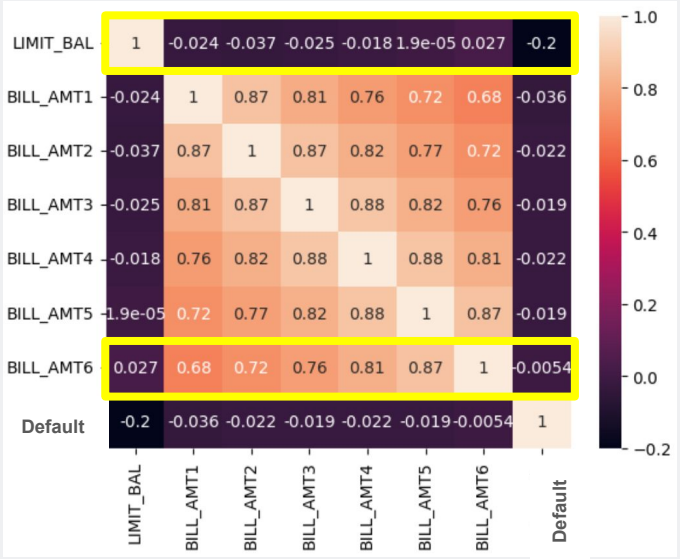


The amount of the bill also has a right skewed distribution.

We first took the log transformation on heavily right skewed features and normalized by the z-score.

Visualizing the data - Collinearity





There is modest correlation between monthly payments. We may be able to make a prediction based on the repayment status of the first couple of months.

The amount of credit has little correlation with default and in general. Correlation between bill payments month-to-month is relatively strong.

Approach and Model Training

Berkeley
UNIVERSITY OF CALIFORNIA

Data Preprocessing and model building up

```
#Convert categorical columns into type categories and make them in the form of one hot coding.
categorical_cols = ['SEX', 'EDUCATION', 'MARRIAGE']
df[categorical_cols] = df[categorical_cols].astype('category')

df_EDA = pd.get_dummies(df).fillna(0)
```

```
# Normalize the input features using Z-score normalization
scaler = StandardScaler()
X_normalized = scaler.fit_transform(X)
```

```
# Create a TensorFlow Dataset from the normalized input features and target labels
dataset = tf.data.Dataset.from_tensor_slices((X_normalized, y))

# Shuffle the dataset
dataset = dataset.shuffle(buffer_size=len(X_normalized))

# Split the data into train, validation, and test sets
train_size = int(0.8 * len(X_normalized))
val_size = int(0.1 * len(X_normalized))
test_size = int(0.1 * len(X_normalized))

# Split the dataset into train, validation, and test sets
train_dataset = dataset.take(train_size)
val_dataset = dataset.skip(train_size).take(val_size)
test_dataset = dataset.skip(train_size + val_size).take(test_size)

# Batch the datasets
batch_size = 8
train_dataset = train_dataset.batch(batch_size)
val_dataset = val_dataset.batch(batch_size)
test_dataset = test_dataset.batch(batch_size)
```

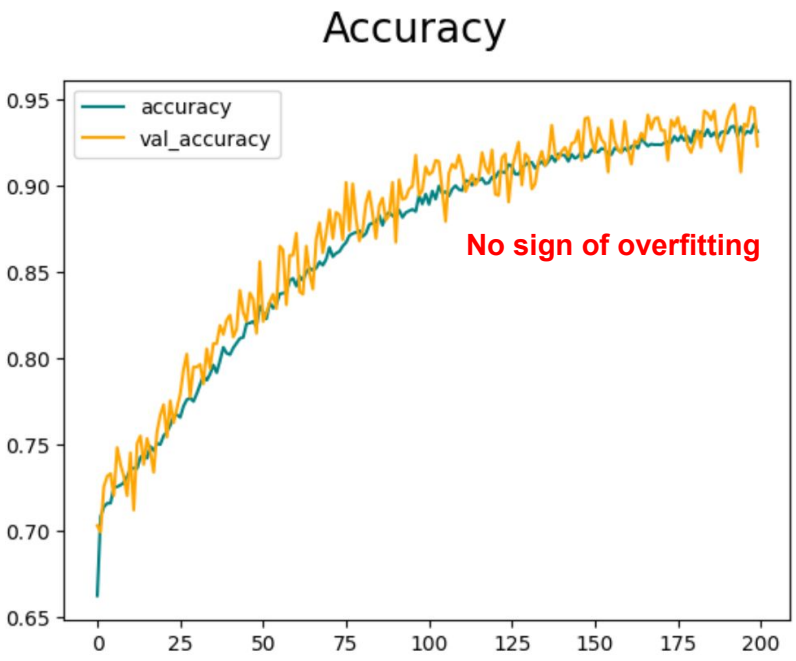
Input shape (8, 33)

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense (Dense)                (None, 256)               8704

dense_1 (Dense)              (None, 64)                16448

dense_2 (Dense)              (None, 16)                1040

dense_3 (Dense)              (None, 32)                544

dense_4 (Dense)              (None, 64)                2112

dense_5 (Dense)              (None, 2)                 130

=================================================================
Total params: 28,978
Trainable params: 28,978
Non-trainable params: 0
_________________________________________________________________
```

```
model.add(Dense(2, activation='softmax'))
```

Treated the output layer as multi class classification with two classes instead of binary classification (sigmoid) for clear evaluation.

Model training (All 33 features)

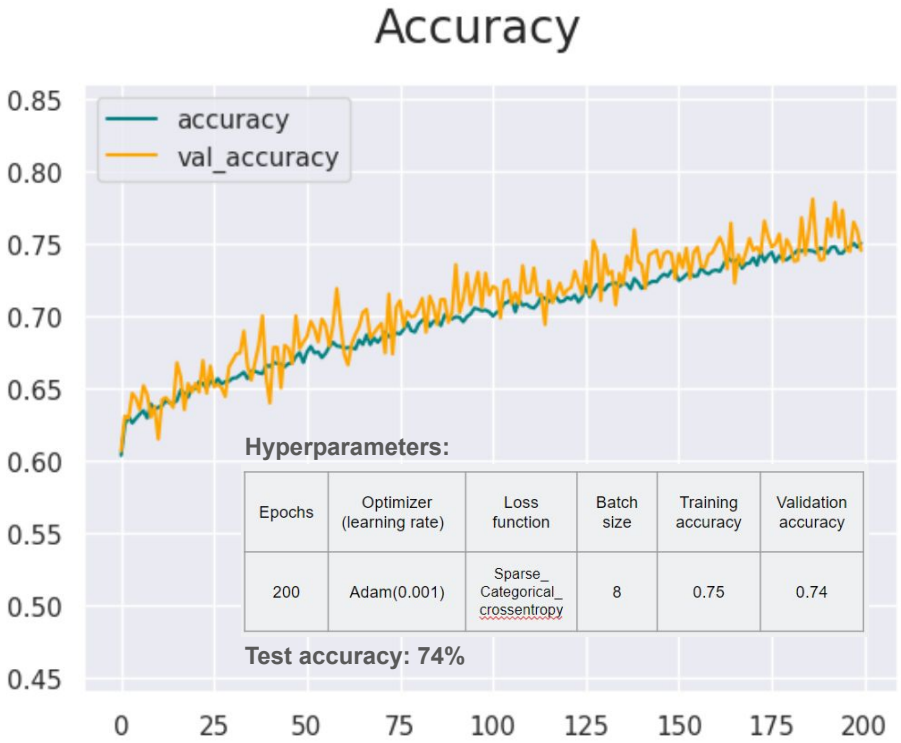


Hyperparameters:

Epochs	Optimizer (learning rate)	Loss function	Batch size	Training accuracy	Validation accuracy
100	Adam(0.01)	Sparse_ Categorical_ crossentropy	16	90%	90%
150	Adam(0.01)		16	92.5%	93%
190	Adam(0.01)		16	93%	93%
200	Adam(0.001)		8	93%	94%

Test accuracy: 93.6%

Model training (13 features)



Hyperparameters:

Epochs	Optimizer (learning rate)	Loss function	Batch size	Training accuracy	Validation accuracy
200	Adam(0.001)	Sparse_ Categorical_ crossentropy	8	0.75	0.74

Test accuracy: 74%

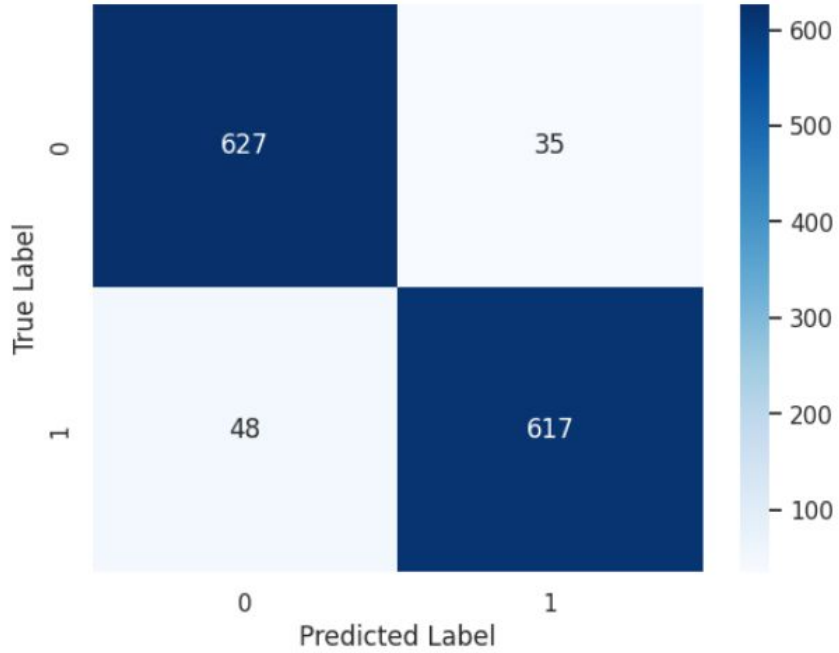
```
#Removing collinear features may make us lose some information under these columns.
#Know whether default or not 6 months ahead.
#Remove features that contain less data

X_reduced_2 = df[['LIMIT_BAL', 'AGE', 'PAY_6','BILL_AMT6','PAY_AMT6',
            'SEX_1', 'SEX_2', 'EDUCATION_0', 'EDUCATION_1', 'EDUCATION_2',
           'EDUCATION_3','MARRIAGE_1', 'MARRIAGE_2']].values
```

33 features may be too dense for the model to learn. Furthermore, if we only adapt the payment status of the first month and make a prediction only based on this, our model is predicting default five months ahead which is meaningful in a business context.

Evaluation

Berkeley
UNIVERSITY OF CALIFORNIA

Evaluation of the model (all 33 features) for an imbalanced data set (default)

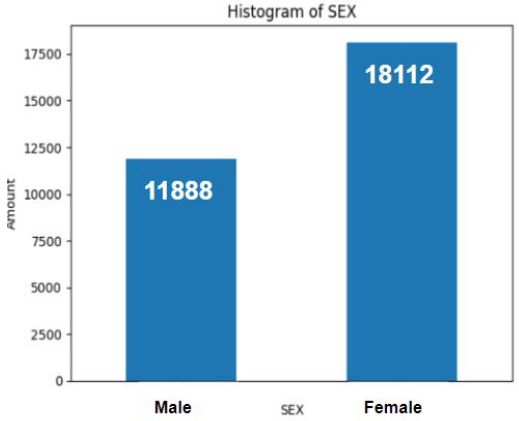


Data set and performance:

Total	Non-Default	Default	Epochs	Test accuracy	Recall	Precision
30000	23364	6636	100	90%	74%	88%
22636	16000	6636	100	90%	78%	89%
16636	10000	6636	100	89.7%	84%	89%
13272	6636	6636	100	89.5%	86%	94%
13272	6636	6636	200	93%	94%	93%

For a given default data example, we can be 94% sure to identify it as default.

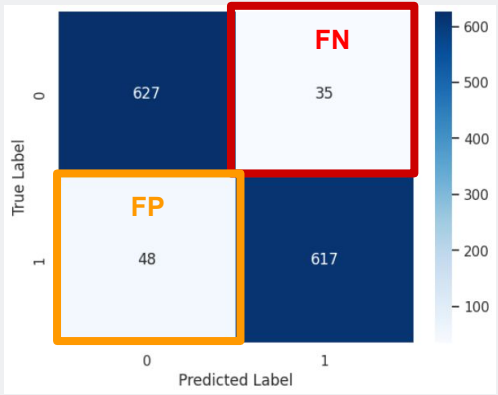
Evaluation of the model (all 33 features) for an imbalanced data set (sex)



We have imbalanced gender data set, around 40% of male and 60% of female so our model is likely to learn more from female over male and cause gender bias.

In this case, the false negative rate on both genders is similar to the gender distribution in the data. However, the FP rate of females is slightly higher, which implies for a given default example, females are more likely to be predicted as non-default.

False positive and negative of the gender subgroup

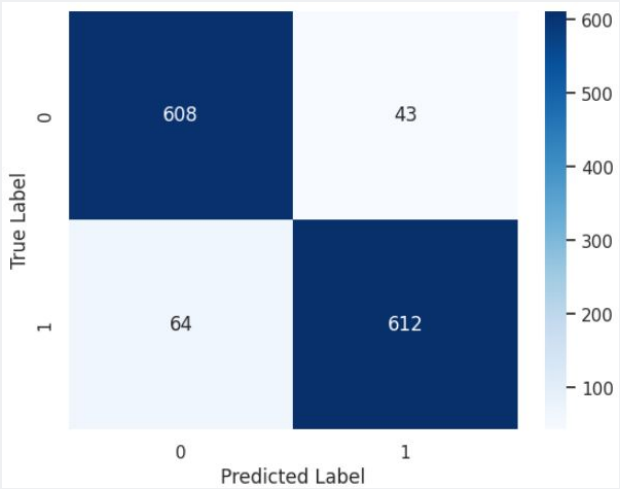


Gender rate	FN_Male	FN_Female	FP_Male	FP_Female
41% male 59% female	16	19	15	33
	45%	55%	31%	69%

Training and Evaluation of the model - No personal information

```
#Removing categorical features (Gender, Education, Marriage)

X_reduced_3 = df[['LIMIT_BAL', 'AGE', 'PAY_0', 'PAY_2', 'PAY_3', 'PAY_4', 'PAY_5',
                  'PAY_6', 'BILL_AMT1', 'BILL_AMT2', 'BILL_AMT3', 'BILL_AMT4',
                  'BILL_AMT5', 'BILL_AMT6', 'PAY_AMT1', 'PAY_AMT2', 'PAY_AMT3',
                  'PAY_AMT4', 'PAY_AMT5', 'PAY_AMT6']].values
```

Our model makes a prediction based on the users' payment status, and is not using any personal information such as gender, education level, or marital status to prevent bias. We only feed the users' age and their payment status as input features, reducing the feature set from 33 features to 20 features.



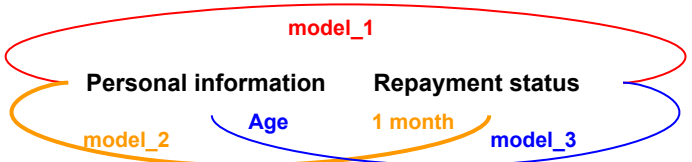
Features	Not Default	Default	Epochs	Test accuracy	Recall	Precision
33	6636	6636	200	93%	94%	93%
20	6636	6636	200	90%	90%	93%

Ensemble
(Bagging: Weighting Average)

So, we have

1. **model_1:** a comprehensive model with personal information and repayment status for the last 6 months as input(33 features)
2. **model_2:** a model based on collinearity of repayment status between months so it can make a prediction based on the payment status of the first month only. (13 features)
3. **model_3:** a model with only repayment status and age to remove potential bias. (20 features)

For a business scenario we would apply model_3 and boost the accuracy through other models using the ensemble method.



Sample size	Epochs	Weights of models	Test accuracy	Recall	Precision
13272	200	[1,0,0]	93%	94%	93%
13272	200	[0,0,1]	90%	90%	93%
13272	200	[0.2, 0.1, 0.7]	93%	92%	94.1%
13272	200	[0.4, 0.1, 0.5]	95%	95%	95.7%
13272	200	[0.7,0.5,0.6]	95.7%	95.4%	96%

Conclusions

Berkeley
UNIVERSITY OF CALIFORNIA

Conclusions

Societal impacts:

The model identifies certain demographics as risky, hence the model can be used to limit credit access to certain demographics, which is unfair and not ethical.

Limitations:

The data spans a short amount of time of 6 months and comes from a specific geography. The model therefore is not taking into consideration the economic environment and it might be inaccurate for other geographies.

A model trained on data from 2005 might not be accurate nearly two decades later.

Key Results and Future Work

- Removing features like gender, level of education, marital status results in a lower but sufficient test accuracy of 90%
- By using ensemble modeling - learning from all previous models - we achieved an accuracy of 95.7% .95.4% recall, and a precision of 96%

Future work:

- Tune hyperparameters, use additional inputs like interest rate, inflation rate in country, number of credit cards owned
- Research if models with local data work better, or is that overfitting?

Contribution

- **(20%) Code and presentation submission**: Daniel
- **(20%) Dataset and EDA:** Daniel/Ryan
- **(30%) Approach and Models:** Ryan
- **(20%) Tuning and Improvements:** Mehmet
- **(10%) Conclusions and Checklist:** Mehmet