Dean&Francis
ISSN 2959-6130

House loans decisive factors (or features) comparison: helping customers get loans

Lingxiao He[1],

Zhekai Yu[2*],

Chenshirui Wang[3],

Yue'er Yang[4]

[1]*International Department, Qingdao No.58 Middle School, Qingdao, 266000, China, Douglass.hlx@outlook.com*
[2]*Edmonds college, Lynnwood, 98026, the United States, 15337243239y@gmail.com*
[3]*Merchiston International School, Shenzhen, 518000, China, rruimw704@outlook.com*
[4]*Zhejiang Province Hangzhou No.2 Middle School, Hangzhou, 310053, China, 18868871096@163.com*

**Corresponding author email: 15337243239y@gmail.com*

Abstract:
This article provides a method, system, and related equipment for comparing decisive factors of house loans to assist consumers in securing loans. We chose the data that includes many factors about personal such as marriage/single, income. The study was conducted applying decision tree, neural network, and logic regression. The method includes obtaining consumer financial data; segmenting this data into multiple overlapping financial facets; generating a preliminary financial profile from these facets; performing noise reduction on the preliminary profile to create a detailed financial model; slicing the detailed model into various financial segments; and analyzing these segments to derive a multidimensional understanding of key house loan factors. This approach minimizes both explicit and implicit risks, ensuring a precise alignment with consumer financial profiles. Additionally, converting three-dimensional financial data into two-dimensional segments reduces computational complexity and errors, ensuring overall accuracy in loan evaluation and decision-making. In the last, we use these models to compare which factor influence the risk of loan mostly, so we advise if customers do not have house or car, keep working in one field and carefully view different places' loan policies still can be helpful for their loaning.

Keywords: decision tree, neural network, logic regression, prediction, house loans, mortgage loans, consumers.

1. Introduction

Nowadays unstable economic environment requires banks to be more rigorous in their approach to lending, and the factors affecting lending have become more varied. But we discover that there is too much research talk about which model can best predict the loan default or availability. So we want to try the different way that focuses on which factors will significantly affect the loaning availability. The main features we cover include credit score, income stability, down payment, debt-to-income ratio, profession, professional experience, city, married or single, house ownership and car ownership. And the models we use in this paper are Decision Tree, Neural Network, Logistics Regression.

A high credit score often results in lower interest rates and better loan terms. Consistent income and employment history reassure lenders of repayment capability. A substantial down payment can reduce loan amount and monthly payments, while a favorable debt-to-income ratio indicates manageable debt levels relative to income. Professional experience, on the other hand, shows how stable a person's life is in general. A stable marriage refers to greater ability to pay, in addition, with kids and parents to support, applicant will have to spend more money on basic life. Ownership of cars and houses make sure the banks can get the money back because they can serve as a pledge if applicant fails to pay the debt. Different cities represent different rates of consumption.

2. Literature Review

The prediction of house loan approval is a critical topic in finance, aiming to improve loan granting efficiency and minimize defaults. This literature review explores key studies on predicting loan defaults using decision trees, neural networks, and logistic regression, providing insights into their methodologies, findings, and contributions.

Li et al. (2022) proposes a multi-model fusion approach for loan default prediction, highlighting the effectiveness of integrating various models to enhance accuracy. Their study underscores the importance of combining different techniques for robust predictions.[9]

Wang et al. (2023) apply a heterogeneous ensemble learning approach and SHAP method to predict national student loan defaults. This research emphasizes the significance of interpretability in machine learning models, demonstrating how SHAP values can provide insights into feature importance.

Li and Wu (2024) focus on explainable machine learning for loan default predictability. Their findings stress the necessity of transparency in predictive models, which can build trust and provide clear justifications for loan approval decisions.[10]

Ayberk and Önder (2022) explore the relationship between house prices and bank loan portfolios, particularly the role of bank ownership. Their study reveals how external economic factors, and institutional characteristics influence loan decisions.[11]

The reviewed literature highlights the evolving landscape of loan default prediction, with multi-model approaches and interpretability being central themes. While these studies contribute significantly to the field, further research is needed to refine predictive models and explore additional factors influencing loan approval.

3. Models and Methods

Logistic Regression:

Logistic regression is a statistical method for binary classification tasks. It models the probability that a given input belongs to a particular class. As for binary variables, they can be generalized to categorical variables when there are more than two possible values. [1]

Formula:

m is the midpoint of the logistic curve (the location parameter), and s is scale parameter which makes the distribution more spread out as it gets larger.

In this variable formula, (we've known the $y = b_0+b_1x$) [2]

$b_0 = -m/s$, is the y-intercept.

$b_1 = 1/s$, is the rate parameter.

A common way to measure the goodness of fit of logistic regression is to use logical loss (also known as log loss), or negative log-likelihood.

For given *xk* and *yk* in the regression:

We use *pk* to represent the probabilities of *yk* equals to 1 (which means the house loans could be permitted in this research);

On the contrary, the 1−*pk* are the probabilities that they will be zero (which means the house loans are not permitted in this research)

Usually, we use ℓk to present the log loss of the data and define it: [1]

$$yk = 1, lk = -lnpk$$

$$yk = 0, lk = -(1 - lnpk)$$

Since Log loss is always greater than or equal to 0, equals 0 only in case of a perfect prediction, and approaching infinity as the prediction gets worse. And in a logistic regression it is not possible to have zero loss at any points, since *yk* is either 0 or 1, but $0<pk<1$.

Combining with log loss formulas above because they are measuring a same event's different results, we can get new formula:

In professional words, this expression is more formally known as the cross-entropy of the predicted distribution (*pk*, (1−*pk*)) from the actual distribution (*yk*, (1−*yk*)),

On the contrary, we could use the Log Loss formula to maximize the Log-likelihood: [1]

$$\iota = \sum_{k:yk=1}^{\mp} \sum_{k:yk=0}^{-} ln(1-pk) = \sum_{k=1}^{K} \left(ykln(pk) + (1-yk)ln(1-pk) \right)$$

Decision Tree Learning

Decision tree learning is a supervised learning approach. In this learning, we use classification and regression Decision Tree as a model to predict our dataset.

A tree model in which the target variable can take a set of discrete values is called a classification tree. In these tree-like structures, leaves represent class labels, and branch-

ISSN 2959-6130

ing tables represent the conjunctions of features that lead to these class labels.
There are many different kinds of Decision Tree: [4]
ID3 (Iterative Dichotomiser 3)
C4.5 (successor of ID3)

CART (Classification and Regression Tree)

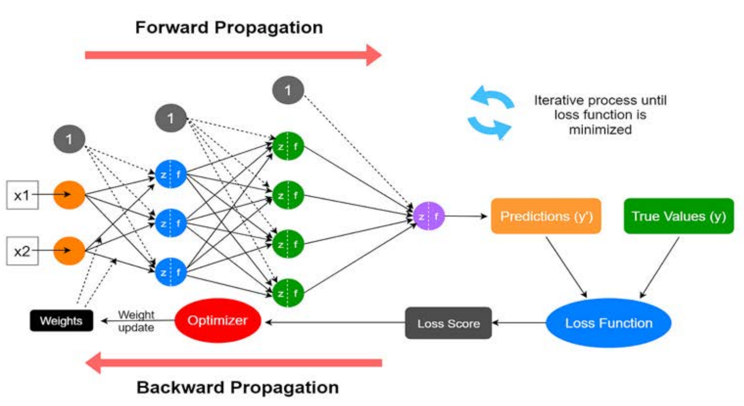
OC1 (Oblique classifier 1). First method that created multivariate splits at each node.
Chi-square automatic interaction detection (CHAID). Performs multi-level splits when computing classification trees.
In our research, we mainly use CART to analyze the house loans because it is the most efficient and general one.
Decision Tree method is used mainly in two main types:
1. Classification Tree: Predict the result category of the desired data, such as defining faces to human class and cat faces to animal class.
2. Regression Tree: Regression tree analysis is used when we want to get a real number when predicting, such as the price of house loan or the years people have spent on recent jobs.
Advantages: intelligibility and simplicity.
Dataset form: $(x,Y) = (x_1,x_2,x_3,\ldots,x_k,Y)$ [4]
Metrics
Since the decision tree is constructed from the top down, the best variable is selected (split) through each branch of the tree, and different Metrics are developed to measure whether this action is the "best".

Gini impurity

In our experiments, we mainly use Gini impurity to measure the "best" since Gini impurity was first developed by analyzing with CART algorithms for classification trees. With similar situations of Corrado Gini, thus we choose Gini impurity.
Gini Impurity value is between 0 and 0.5,
The closer to 0, the better splits done by CART is.
The closer to 0.5, the worse splits sone by CART is.
For a set of items with J classes and relative frequencies pi, $i \in \{1,2,3..., J\}$, the probability of choosing an item with label i is pi, and the probability of mis categorizing that item is $\sum_{(k \neq i)}^{-} pk = 1 - pi$.
Then, we can sum up all the pairwise products of these probabilities for each label. [5]

Neural Network

An artificial neural network consists of connecting units, or nodes, of artificial neurons that loosely mimic neurons in the brain.
They are connected by the edge, which mimics the synapses in the brain and the way nerves transmit electrical signals. Each artificial neuron receives signals from connected neurons, then processes those signals and sends the signals to other connected neurons.
In machine learning, the "signal" is a real number, and the output of each neuron is calculated as a nonlinear function of the sum of its inputs, called the activation function.
The signal strength at each connection is determined by a weight, and the optimal solution is found by repeating the balance weight several times.
Advantages:
1. Neutral networks that can work continuously and are more efficient than humans or simpler analytical models. [6]
2. Using algorithmic computing in the cloud can reduce risks such as data breaches.
3. This is the Most comprehensive and complicated model, which fits mostly to all situations when put into varies industries to analyze and predict.
4. It can be programmed to learn from prior outcomes to strive to make smarter future calculations. [6]



Convolutional Neural Networks and Deconvolutional Neural Networks

The data of a convolutional neural network is divided into multiple layers, including an input layer, an output layer, and a large number of hidden convolutional layers in the middle for "thinking". The whole process is like our linear thinking, the cause goes through the effect, and finally comes to the desired conclusion.

Formula: $Y = \beta 0 + \beta 1 \cdot x1 + \beta 2 \cdot x2 + \beta 3 \cdot x3 + \ldots + \beta k \cdot xk + \epsilon$

Deconvolution neural network is the category that we use most in the analysis of weights, which is the reverse of the convolutional neural network. After many more neural networks "thinks", we can reverse the search for the weights of each independent variable x. It's like reverse thinking in humans. In this research, it can help us to advise borrowers and make it easier for them to get loans.

4. Experiment and research

4.1 Data processing

In this paper, the database we use is Loan Prediction Based on Customer Behavior, which is open source on Kaggle website. And the whole original dataset has 252,000 loan data from all different individuals, with 13 columns.

Data Balance:

First, there are too many "0" in the Dependent variable, which means most of the applications for house loans were denied by the banks in the original chosen dataset. Thus, we need to balance the data probabilities in our chosen dataset by using the method of sampling balance. To be specific, in this research, we increase the weight of "1" (means access the house loans) in the dataset to make sense for the predictions.

Data cleaning:

The team has checked all the columns in the original dataset. Some columns are pretty homogeneous, such as the city and the state, which might influence the accuracy of the analyzing, thus we delete the column.

The whole training data 252,000 individuals which is too much to make analysis and prediction without any overfitting. Thus, we used python to randomly choose 20,000 unique (approximately 8.7%) values in the training data by programming.

4.2 Independent Variables

After the data cleaning, we deleted City, ID, Profession since they are not really related to the Risk_Flag many. This is caused by the uniqueness of these factors, causing the weight to become too low.

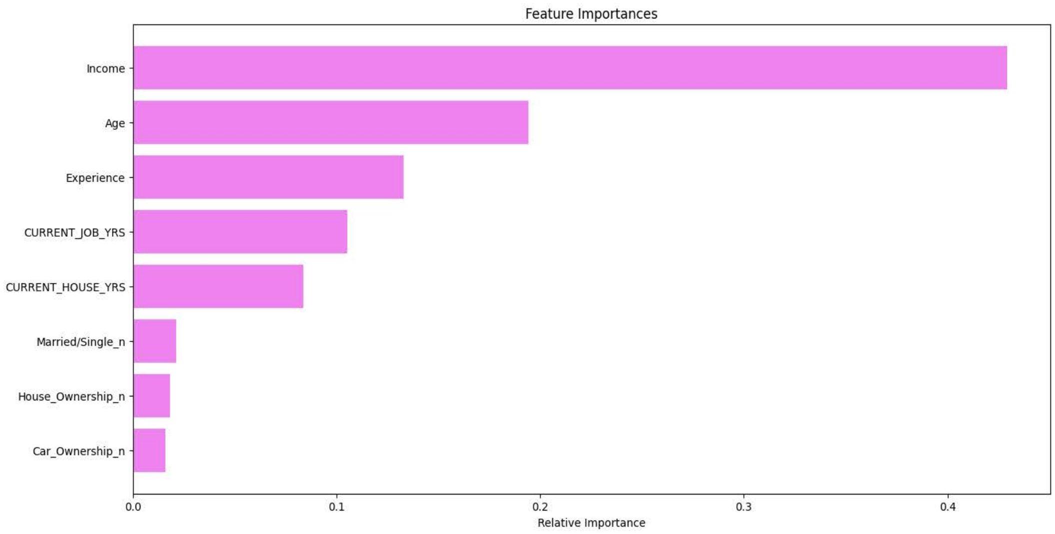
We Keeps income, age, Experience, house/car owning and some other high correlation factors.

Features (factors)	Description	Data Type
Income	Income of the user	Int
Age	Age of the user	Int
Experience	Professional experience of the user in years	Int
Married/ Single	Whether married or single	String
House_Ownship	Owned or rented or neither	String
Car_Ownship	Does the person own a car	String
States	The state people live in	String
Current_House_Yrs	Years of experience in the current residence	Int
Risk_Flag	Defaulted on a loan	String

Y: "0" = refuse/ deny the house loans; "1" = accept the applications of house loans.

After analyzing by the Neural Network, it found out the most credible ranking of the importance of the effective factors.

ISSN 2959-6130



As shown above, neural network has automatically learned the sample dataset.

4.3 EDA

```
"E:\Users\15337\PycharmProjects\Hello python\pythonProject1\.venv\Scripts\python.exe" "E:\Users\15337\PycharmProjects\Hello
 python\pythonProject1\orange3"
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 252000 entries, 0 to 251999
Data columns (total 9 columns):
 #   Column              Non-Null Count   Dtype
---  ------              --------------   -----
 0   Income              252000 non-null  int64
 1   Age                 252000 non-null  int64
 2   Experience          252000 non-null  int64
 3   CURRENT_HOUSE_YRS   252000 non-null  int64
 4   Married/Single_n    252000 non-null  int32
 5   House_Ownership_n   252000 non-null  int64
 6   Car_Ownership_n     252000 non-null  int32
 7   Profession_n        252000 non-null  int64
 8   State_n             252000 non-null  int64
dtypes: int32(2), int64(7)
memory usage: 15.4 MB
None

Process finished with exit code 0

```

```
"E:\Users\15337\PycharmProjects\Hello python\pythonProject1\.venv\Scripts\python.exe" "E:\Users\15337\PycharmProjects\Hello
 python\pythonProject1\orange3"
             Income            Age  ...   Profession_n       State_n
count  2.520000e+05  252000.000000  ...  252000.000000  252000.000000
mean   4.997117e+06      49.954071  ...       0.023639       0.202873
std    2.878311e+06      17.063855  ...       0.151922       0.603813
min    1.031000e+04      21.000000  ...       0.000000       0.000000
25%    2.503015e+06      35.000000  ...       0.000000       0.000000
50%    5.000694e+06      50.000000  ...       0.000000       0.000000
75%    7.477502e+06      65.000000  ...       0.000000       0.000000
max    9.999938e+06      79.000000  ...       1.000000       2.000000

[8 rows x 9 columns]

Process finished with exit code 0

```

Before we build the models, we need to preprocess the data. This part will skip introducing the common preprocessing data but put more concentration on our special works for data.

First, our data is not balanced data, for example, in our features “House_Ownership” that column, “rented” take 92% place, this will cause our models especially for classification models to ignore the effect of “owned” feature since “owned” just has 3%. This problem also occurred in dependent variable “Risk_Flag”. In”Risk_Flag”, 0 is the majority, so it will let the model believe that some features changing will not make sense. Thus, we balanced the training data after we splited the test and train data to build the models.

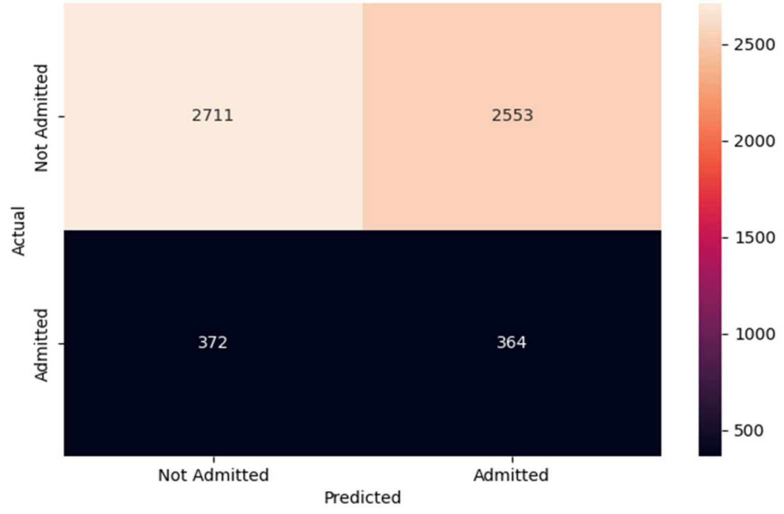
Second, we dropped two columns “CITY” and “CURRENT_JOB_YRS”. The reason we made this decision was because both columns overlapped. In the data frame, we have “STATE” and “CITY” two features, but they are too similar, they all represent the feature of area, if we want to see what will bring to dependent variable by the feature of area, we want to see only one column to represent the feature of area since it is clear and easier to build the models. And "CITY" is more complex than "STATE", so we chose "STATE" as our feature of area. Also, "Experience" and "CURRENT_JOB_YRS" are similar, too. "Experience" is the feature that measures the customer's professional experience in years like how many years you work in this field. "CURRENT_JOB_YRS" is how many years does customer works in this job. The short of "CURRENT_JOB_YRS" is when the customer changes the job, this feature will come to 0. Thus, if we try to know how many years a customer stays in one field, "Experience" is better than "CURRENT_JOB_YRS.

Third, we scaled the data. "Income" is the special one column since the number of this feature is too big, we want to decrease the influence of big numbers. Thus, we scaled the data.

4.4 Data Analysis and Prediction by models

Following the standard procedure of verifying the accuracy and conformity of the model, the three models were all programmed to produce the accuracy and other measuring standards after the analyzing and prediction.

4.4.1 Logistic Regression

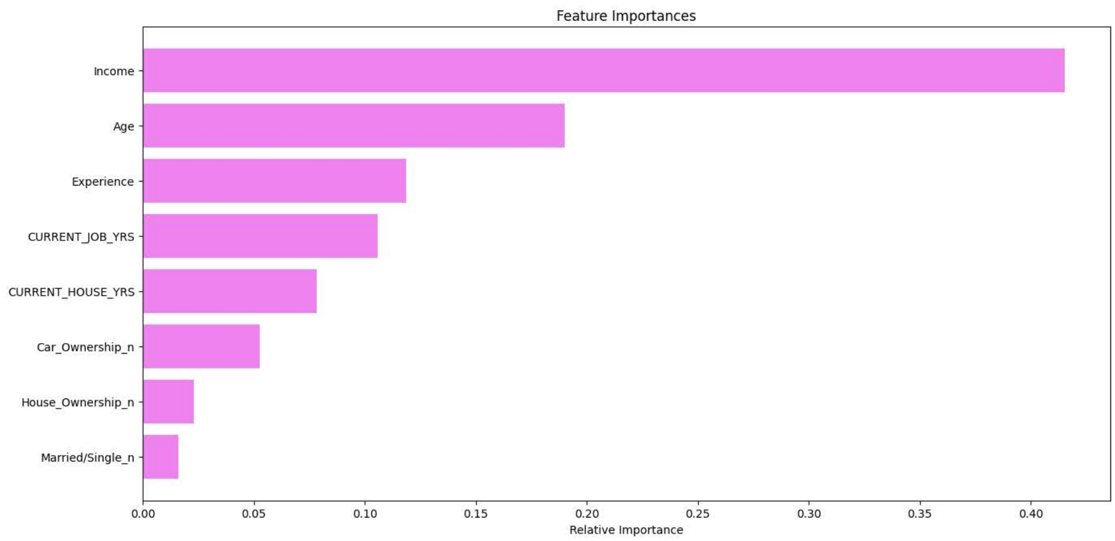



```
"E:\Users\15337\PycharmProjects\Hello python\pythonProject1\.venv\Scripts\python.exe" "E:\Users\15337\PycharmProjects\Hello python\pythonProject1\apple 2"
0.5301666666666667
[[2803 2482]
 [ 337  378]]
Recall on training set: 1.0
Recall on test set: 0.5286713286713287
Normalized Confusion Matrix:
[[0.53036897 0.46963103]
 [0.47132867 0.52867133]]
E:\Users\15337\PycharmProjects\Hello python\pythonProject1\.venv\Lib\site-packages\sklearn\base.py:486: UserWarning: X has feature names, but LogisticRegression was
  warnings.warn(
              precision    recall  f1-score   support

           0       0.89      0.53      0.67      5285
           1       0.13      0.53      0.21       715

    accuracy                           0.53      6000
   macro avg       0.51      0.53      0.44      6000
weighted avg       0.80      0.53      0.61      6000

Optimization terminated successfully.
         Current function value: 0.647318
         Iterations 6
                           Logit Regression Results
==============================================================================
Dep. Variable:              Risk_Flag   No. Observations:                24486
Model:                          Logit   Df Residuals:                    24477
Method:                           MLE   Df Model:                            8
Date:                Fri, 02 Aug 2024   Pseudo R-squ.:                 0.06612
Time:                        06:38:48   Log-Likelihood:                -15850.
converged:                       True   LL-Null:                       -16972.
Covariance Type:            nonrobust   LLR p-value:                     0.000
=====================================================================================
                        coef    std err          z      P>|z|      [0.025      0.975]
-------------------------------------------------------------------------------------
Income             2.922e-08   4.55e-09      6.417      0.000    2.03e-08    3.81e-08
Age                   0.0021      0.001      2.711      0.007       0.001       0.004
Experience           -0.0252      0.002    -10.663      0.000      -0.030      -0.021
CURRENT_HOUSE_YRS     0.0249      0.005      5.090      0.000       0.015       0.035
Married/Single_n      0.0142      0.040      0.356      0.722      -0.064       0.092
House_Ownership_n    -1.6908      0.102    -16.502      0.000      -1.892      -1.490
Car_Ownership_n      -0.9880      0.034    -29.288      0.000      -1.054      -0.922
Profession_n         -0.2313      0.052     -4.440      0.000      -0.333      -0.129
State_n              -1.6619      0.071    -23.413      0.000      -1.801      -1.523
=====================================================================================

Process finished with exit code 0
```

As the graph shown above, after balancing the "1" individuals and "0" individuals in the sampling dataset, we

ISSN 2959-6130

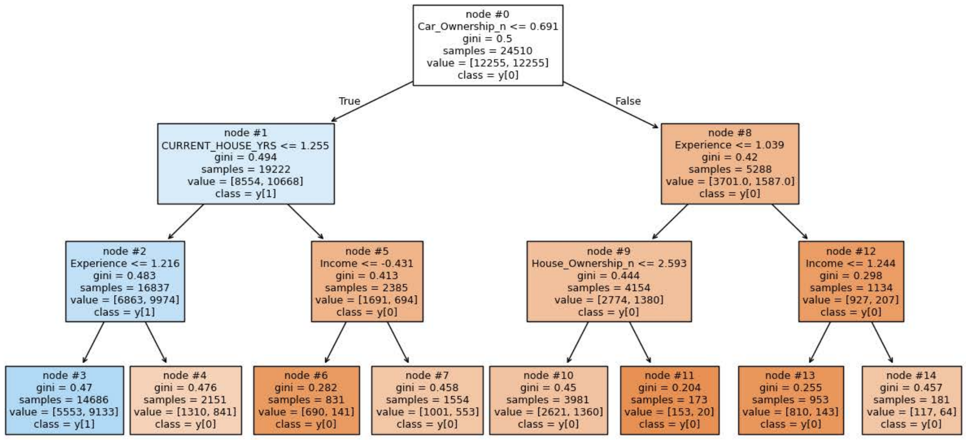
have deleted some of the atypia. Thus, it is obvious the total unique values in the statistic graph above is less than 20,000. Now there are 6,000 individuals available, the approximate accuracy of prediction is better than before. Although it does not seem a really good number, the accuracy is still good to use in most situations since the real data could not be perfect.



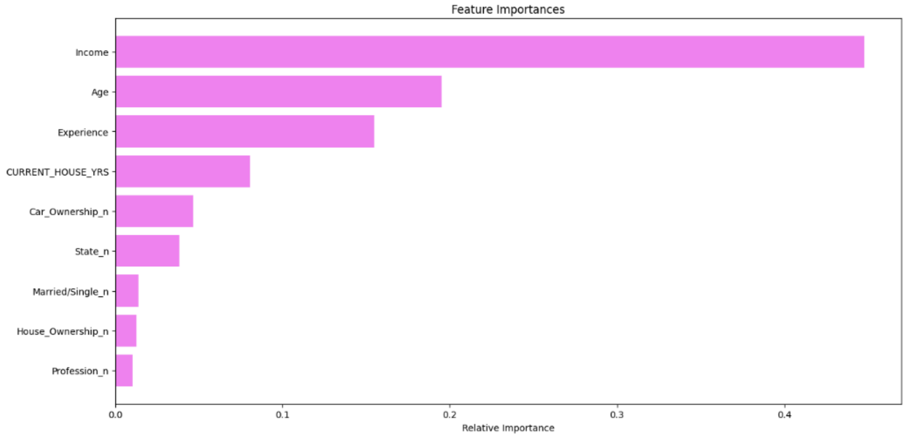
4.4.2 Decision Tree

Hyperparameter: Max_Depth = 4.

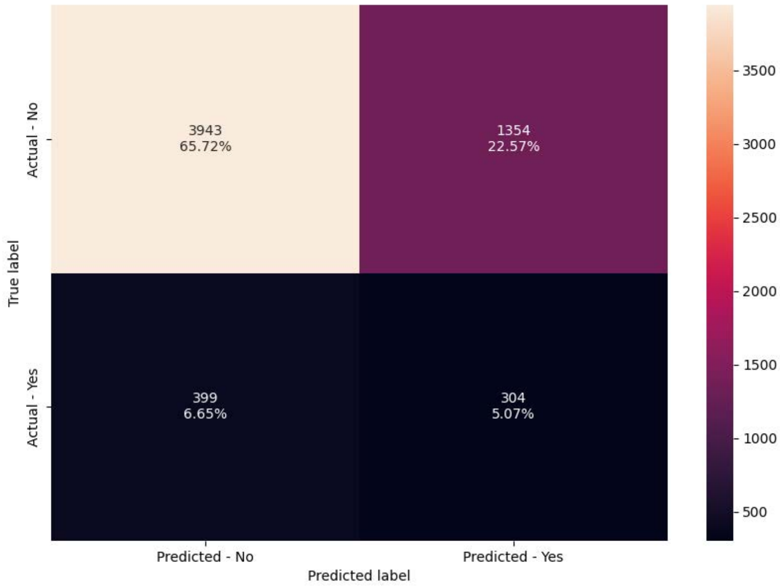
The tree after split:



The decision tree model is shown in the first graph. The class in the last row in each leaf could easily show the result of the application. And in the third row in each leaf, we can get the Gini Impurity calculated by Decision Tree model built by python.



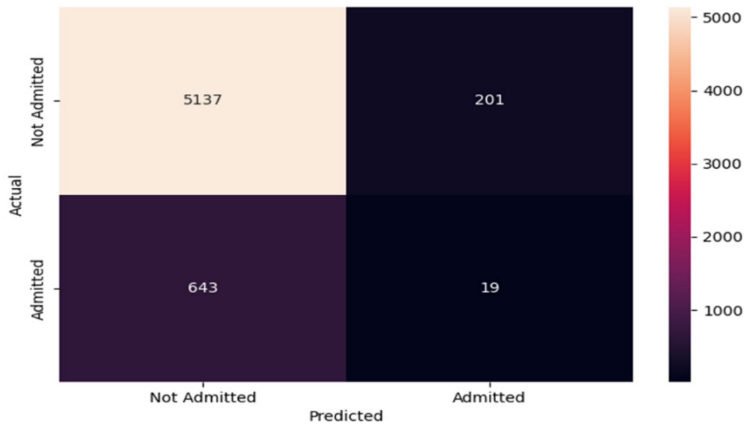
Accuracy calculated by Python:
Accuracy of test: 0.7078333333333333
Accuracy of train: 0.9792329661362709
Recall on training set: 0.9773969808241534
Recall on test set: 0.43243243243243246

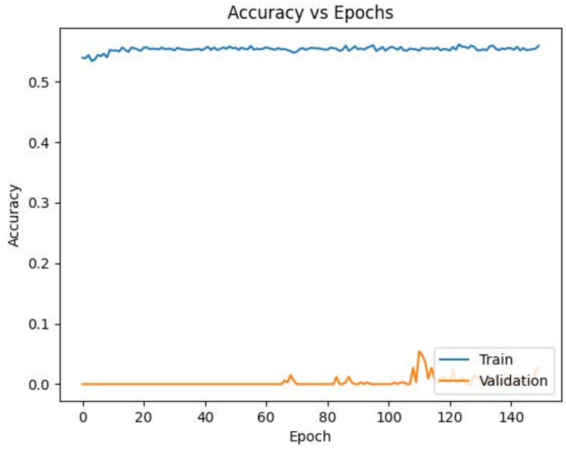


In the second graph, the total rate for predicting correctly is 65.72% + 5.07% = 70.79%, which is such a fantastic accuracy for Analysis and Prediction in Decision Tree model, using the real, normal data. The result shown above indicates the model predicts the people (both predicted and actual are NO), however, the people (both predicted and actual are yes) were predicted far less well. Finally, the decision tree gives the best matching analyzing.

ISSN 2959-6130

4.4.3 Neural Network





The accuracy of the Neural network in training is approximately 0.55 and the validation (Testing) is approximately. Even though the model fits the training data not bad, when it fits the testing data, the accuracy almost is 0. So, this model does not learn well.

Data Analyzing enhance

After analyzing the dataset, all the three chosen models indicate the income is the most effective variable among all. Then, it is necessary to check the conclusion before it comes to a theory that could be shown to the consumers.
As usual, P-value is used to measure the significant level of income to make sure it is totally independent.
However, after calculation, the significant level is too low (we can find through
P-value), much lower than the α = 0.05 and 0.01. Thus, the income is a proper independent variable which could be used in analysis and prediction.
To find the real determinable factors, logistic regression was used to calculate all
P-value of the relevant factors in the sampling dataset.
In the table, it is shown that the absolute value of P-value of the Car_Ownership is
0 and House_Ownership is 0, so we can make sure that Car_Ownership and House_Ownership are the features can really influence the result, but the P-value of Marriage/Single is too high to say it is not enough to affect the result. Then, we can see the coefficients of those factors to determine which one has the most influential to the result. For example, the coefficients of the House_Ownership has the most distance from 0, so we can say that House_Ownership has the largest impact on the result.
Logit Regression Results

Dep. Variable	y	No. Observations	8787
Model	Logit	Df Residuals	8776
Method	MLE	Df Model	10
Date	Wed, 31 Jul 2024	Pseudo R-squ	0.04260

Time	16:59:00	Log-Likelihood	-5830.8
Converged	True	LL-Null	-6090.2
Covariance Type	no robust	LLR p-value	4.129e-105

=== ==============================

	coef	Std err	z	P>|z|	[0.025	0.975]
Income	2.922e-08	4.55e-09	6.417	0.000	2.03e-08	3.81e-08
Age	0.0021	0.001	2.711	0.007	0.001	0.004
Experience	-0.0252	0.002	-10.663	0.000	-0.030	-0.021
CURRENT_HOUSE_YRS	0.0249	0.005	5.090	0.000	0.015	0.035
Marriage/Single_n	0.0142	0.040	0.356	0.722	-0.064	0.092
House_Ownership_n	-1.6908	0.102	-16.502	0.000	-1.892	-1.490
Car_Ownership_n	-0.9880	0.034	-29.288	0.000	-1.054	-0.922
Profession_n	-0.2313	0.052	-4.440	0.000	-0.333	-0.129
State_n	-1.6619	0.071	-23.413	0.000	-1.801	-1.523

5. Conclusion

This article explores decisive factors for house loans and compares methods to assist customers in securing loans. The study employs decision trees, neural networks, and logistic regression. The methodology involves acquiring consumer financial data, segmenting it to create a preliminary financial profile, reducing noise, and analyzing the data to identify key loan factors. Key features analyzed include income, age, professional experience, marital status, and house/car ownership. Results indicate that decision tree models have the highest accuracy (70.79%) respectively. The study highlights the importance of these factors in loan approval and provides a precise, multi-dimensional understanding of loan decision-making, ultimately aiding consumers in obtaining loans more effectively.

Finally, the State_n and House_Ownership is determined to be the most effective factors (independent variables) for this mortgage loans analysis for consumers because they have the highest coefficients which means they are most relevant largely influential variables in the research. Combined with financial analyzing by, the banks are more willing to lend the house loans to people who have houses, but different area may have different policy about loaning [7]. Therefore, the result suggests if customers have car and house, bank is willing to lend you money. But if you do not have those things, put more concentration on the policy of different area about your loan and staying in one job or field more years to get more experience.

Acknowledgement

Lingxiao He, Zhekai Yu, Chenshirui Wang, and Yue'er Yang contributed equally to this work and should be considered co-first authors.

References

[1] What Is Logistic Regression? | IBM. (n.d.). https://www.ibm.com/topics/logistic-regression#:~:text=Logistic%20regression%20estimates%20the%20probability,data%20set%20of%20independent%20variables.

[2] Tolles, Juliana, and William J. Meurer. "Logistic regression: relating patient characteristics to outcomes." Jama 316.5 (2016): 533-534.

[3] Cramer, Jan Salomon. "The origins of logistic regression." (2002): 4.

[4] Maimon, Oded Z., and Lior Rokach. Data mining with decision trees: theory and applications. Vol. 81. World scientific, 2014.

[5] Quinlan, J. Ross. "Induction of decision trees." Machine learning 1 (1986): 81-106.

[6] Hardesty L (14 April 2017). "Explained: Neural networks". MIT News Office. Archived from the original on 18 March 2024. Retrieved 2 June 2022

[7] Bishop, Christopher M., and Nasser M. Nasrabadi. Pattern recognition and machine learning. Vol. 4. No. 4. New York: springer, 2006.

[8] Uddin, Nazim, et al. "An ensemble machine learning based bank loan approval predictions system with a smart application." International Journal of Cognitive Computing in Engineering 4 (2023): 327-339.

[9] Li, Xingyun, et al. "Prediction of loan default based on multi-model fusion." Procedia Computer Science 199 (2022): 757-764.

[10] Li, Huan, and Weixing Wu. "Loan default predictability

ISSN 2959-6130

with explainable machine learning." Finance Research Letters 60 (2024): 104867.
[11] Ayberk, Idil, and Zeynep Önder. "House prices and bank loan portfolios in an emerging market: The role of bank ownership." Economic Modelling 106 (2022): 105683.
[12] Cortés, David, and Pilar Soriano. "Analysis of macroeconomic determinants of non-performance in consumer and mortgage loans." Finance Research Letters 61 (2024): 104939.
[13] Baumöhl, Eduard, Štefan Lyócsa, and Petra Vašaničová. "Macroeconomic environment and the future performance of loans: Evidence from three peer-to-peer platforms." International Review of Financial Analysis 95 (2024): 103416.
[14] Zeng, Guoping. "A Comprehensive Study of Coefficient Signs in Weighted Logistic Regression." Heliyon (2024).
[15] Alvarez Melis, David, and Tommi Jaakkola. "Towards robust interpretability with self-explaining neural networks." Advances in neural information processing systems 31 (2018).