A Case Study of US Mortgage Approvals

Max Melchior Lang

University of Oxford

Oxford, 2023

Contents

1 Introduction

This dataset assesses mortgage applications from a (unknown) 1990 U.S. city, focusing on the binary outcome of approval of the mortgage application. It includes financial ratios such as housing expenses to income (`hir`), other debts to income (`odir`), loan-to-value ratio (`lvr`), and a mortgage credit score (`lvr`) ranging from 1 (best) to 4. Demographic details cover self-employment (`self`), marital status (`single`), and `ethnicity` (white or black), alongside the 1989 state unemployment rate in the applicant's industry (`uria`). The analysis aims to deeply understand factors affecting mortgage approvals.

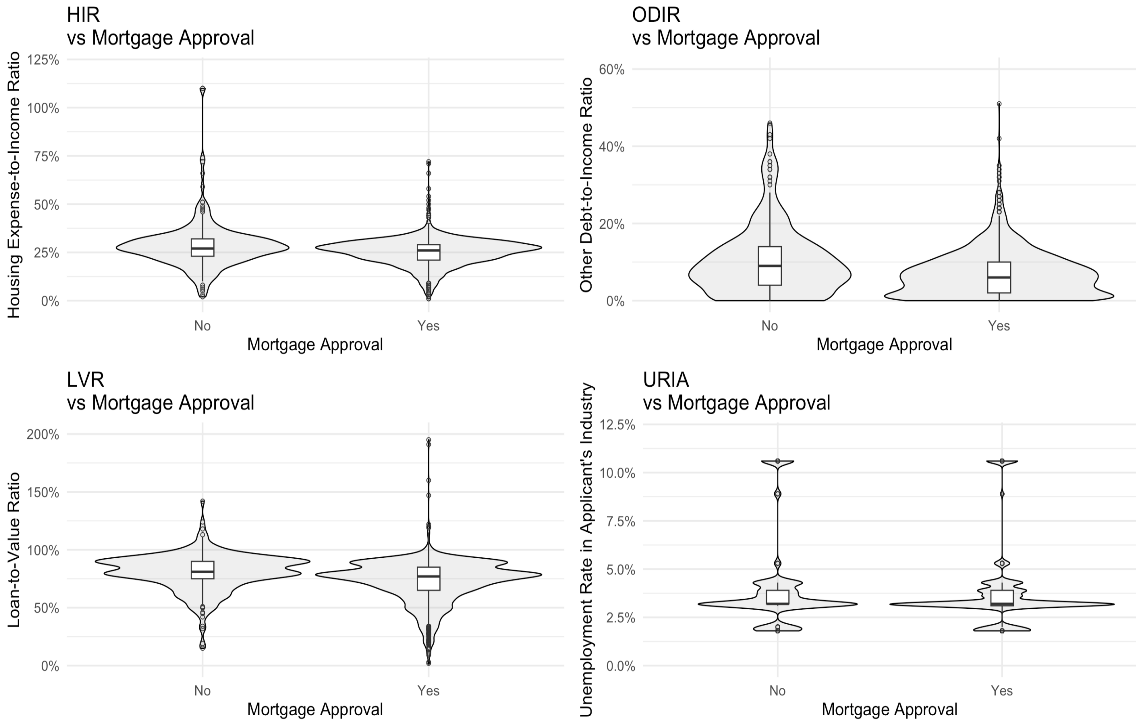

Figure 1: Violin and Box Plot Distribution of Financial Ratios and Unemployment Rate by Mortgage Approval Status

2 Exploratory Data Analysis

The dataset analyzed in this report is a collection of 1902 mortgage applications, with no missing entries, skewed towards approved outcomes (87.5%). For better interpretation, the ratios within the dataset have been transformed to a percentage scale. While the range of values for variables like the Housing Expense to Income Ratio (HIR) and Loan-to-Value ratio (LVR) is broad, they remain within plausible limits (see figure 1). The data reflects a predominance of white, non-single applicants, a lesser proportion of self-employed individuals, and a concentration of applicants with good credit scores. The HIR across applicants averages at 25.5%, with a wider spread of values beyond the interquartile range, suggesting varied housing expenses relative to income among applicants. The Other Debt-to-Income Ratio (ODIR) is skewed, with a median higher for declined (9%) than approved applications (6%). LVR, indicating borrowing risk, also displays asymmetry, with a lower median for approved loans (77%). Outliers on both ends of the spectrum do not imply data issues but rather high variability in loan amounts relative to property values. Similarly, the Unemployment Rate in Industry of Applicant (URIA) is comparable for both approved and denied applications (median for both 3.2 %), with only ten unique values within the dataset. Creditworthiness, as indicated by the Mortgage Credit Score (MCS), shows most applicants in the low-risk categories (1 and 2). Specifically, applicants with the most favorable credit score of 1 have the highest approval rate (92.6%), which marginally decreases with each subsequent credit score category, except from 3 to 4 (2: 85.5%, 3: 75.8%, 4: 78.9%) (see figure 2). Self-employed applicants have a lower approval rate (81.5%) than those not self-employed (88.3%), and single

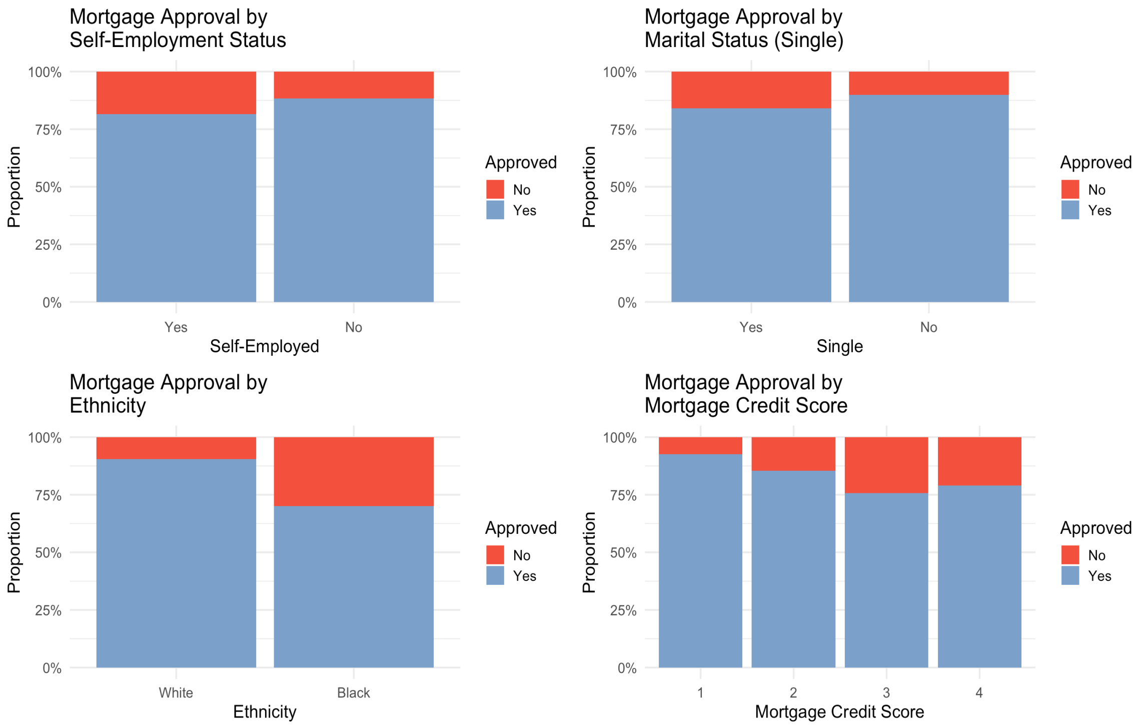

Figure 2: Stacked Bar Charts of Mortgage Approval Rates by Demographic and Financial Criteria.

applicants have a lower approval rate (84.1%) compared to those who are not single (89.8%). Notably, there's a stark contrast in approval rates between white applicants (90.5%) and non-white applicants (70.0%), highlighting potential disparities. The distribution of mortgage approval rates across different financial indicators shows that variables like HIR, LVR, and ODIR display variable patterns, with approval rates dropping at higher deciles (see figure 3). The ODIR plot demonstrates consistent approval rates across most deciles, with a modest downturn in the highest decile. HIR is characterized by a dynamic range, with approval rates peaking in the first decile and decreasing for higher ratio values. Detailed summary tables, in addition to the graphics in figure 1 and 2, can be found in the appendix (see A).

3 Modelling

Given that our main objective is to better understand how the probability of mortgage approvals depends on the explanatory variables, we will leverage a logistic regression model with the logit link to model the binary outcome of mortgage approvals. As given by the task at hand, we constrained our scope to interactions with the *self* variable — denoting self-employment status.

3.1 Model selection

In refining the model selection, I applied the stepwise AIC algorithm that utilizes both forward and backward selection. This approach is applied to the model with all initial variables along with their interaction terms with `self` (see B.1). We are aiming to find a reduced model that balances complexity against its goodness of fit. We retrieve the following linear predictor:

$$\begin{aligned}\eta =& \beta_0 + \beta_1 x_{\text{hir}} + \beta_2 x_{\text{odir}} + \beta_3 x_{\text{lvr}} + \beta_4 x_{\text{mcs}} + \beta_5 x_{\text{self}} + \beta_6 x_{\text{single}} + \beta_7 x_{\text{white}} + \beta_8 x_{\text{uria}} + \\ & \beta_9 x_{\text{odir}} \times x_{\text{self}} + \beta_9 x_{\text{self}} \times x_{\text{white}} + \beta_{10} x_{\text{self}} \times x_{\text{uria}}.\end{aligned} \tag{1}$$

Further, I test whether or not the reduced model misfits the data compared to the initial model via the likelihood ratio test with test statistic

$$D_{\text{reduced}} - D_{\text{initial}} \sim \chi^2(p_{\text{initial}} - p_{\text{reduced}}) = \chi^2(20 - 14) = \chi^2(6). \tag{2}$$

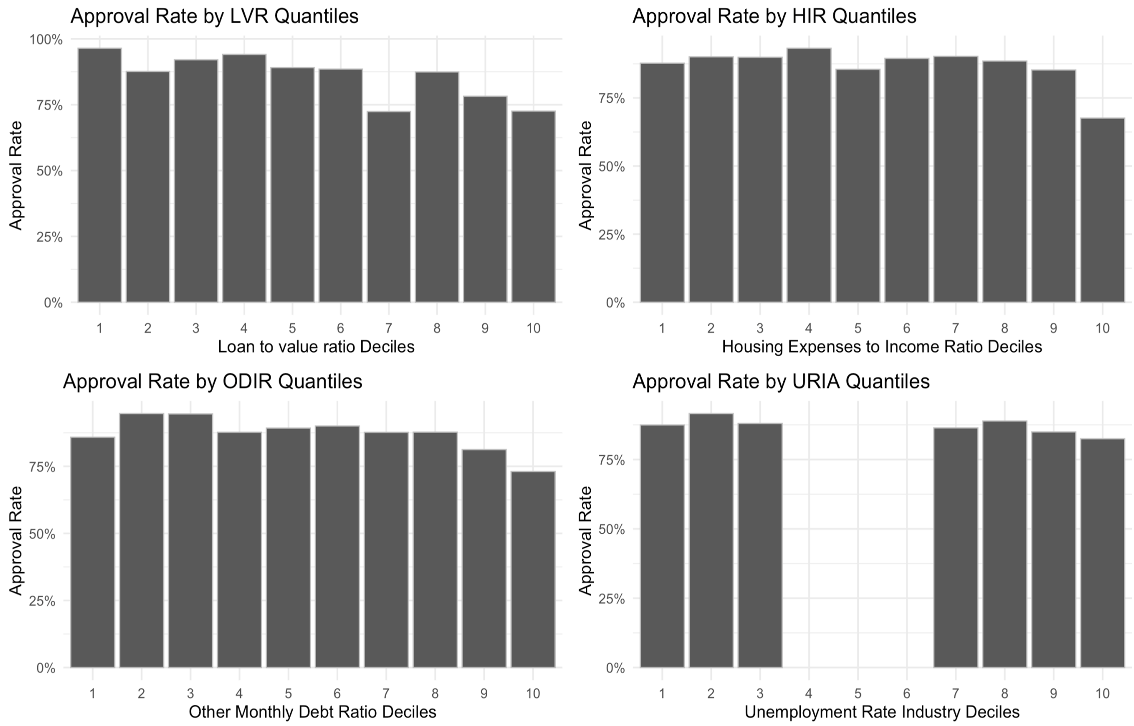

Figure 3: Mortgage Approval Rates across Financial Ratio and Unemployment Rate Deciles.

D_{reduced} is the residual deviance of our reduced model and D_{initial} is the residual deviance for the initial model with all variables and interactions of the self-employment status. We retrieve a p-value of 0.54 showing no significant change in deviance, such that we can simplify the model to our reduced model obtained through the stepwise AIC algorithm. In our models, we encoded the variable mortgage credit score (MCS) as a factor. This is due to the fact that our findings from our exploratory data analysis in 2 suggest that increasing the credit score from category 1 to category 2 is not as severe as increasing the credit score from 2 to 3 or 4. Therefore, we decided to include the variable as a categorical variable (factor) with dummy encoding (reference category MCS = 2) instead of a linear (numeric) effect, which would assume the same effect of changing credit score category between all four categories. Moreover, one can argue that because of the ordinal scale of the credit score, we can only make statements of higher, equal, or lower, but we cannot interpret the differences between categories in a meaningful way and, if included as numeric the relabelling of for instance category 4 to category 100 would influence the estimates and outcome, justifying the factor encoding in our model. The reference categories for the remaining binary/categorical variables are `single` = "Yes", `self` = "No" and `ethnicity` = "White".

3.2 Model diagnostics

In assessing model influence using Cook's Distance, as shown in Figure 4, we identified several influential observations using a soft threshold of 0.02, and the common hard threshold of $8/(n-2p)$. These points, notable enough to warrant scrutiny, were deemed legitimate data rather than errors and are detailed in Table C.1 in the appendix. The model fit visualized in Figure 5 and the associated generalized R-squared ($R^2_{KL} = 0.1542$) indicate a modest explanatory power of the model, typical for logistic regression analyses.

3.3 Results & Interpretation

The logistic regression model output in table 1 gives an overview of the estimated coefficients of our selected model in 3.1. As already pointed out in section 2, we scaled our ratio variables by a factor of 100 such that they are on a percentage point scale for better interpretation.

We observe that the housing expense to income ratio (HIR) shows a negative association with the log-odds of approval, with an estimate of -0.053, with all other variables held fixed (ceteris paribus

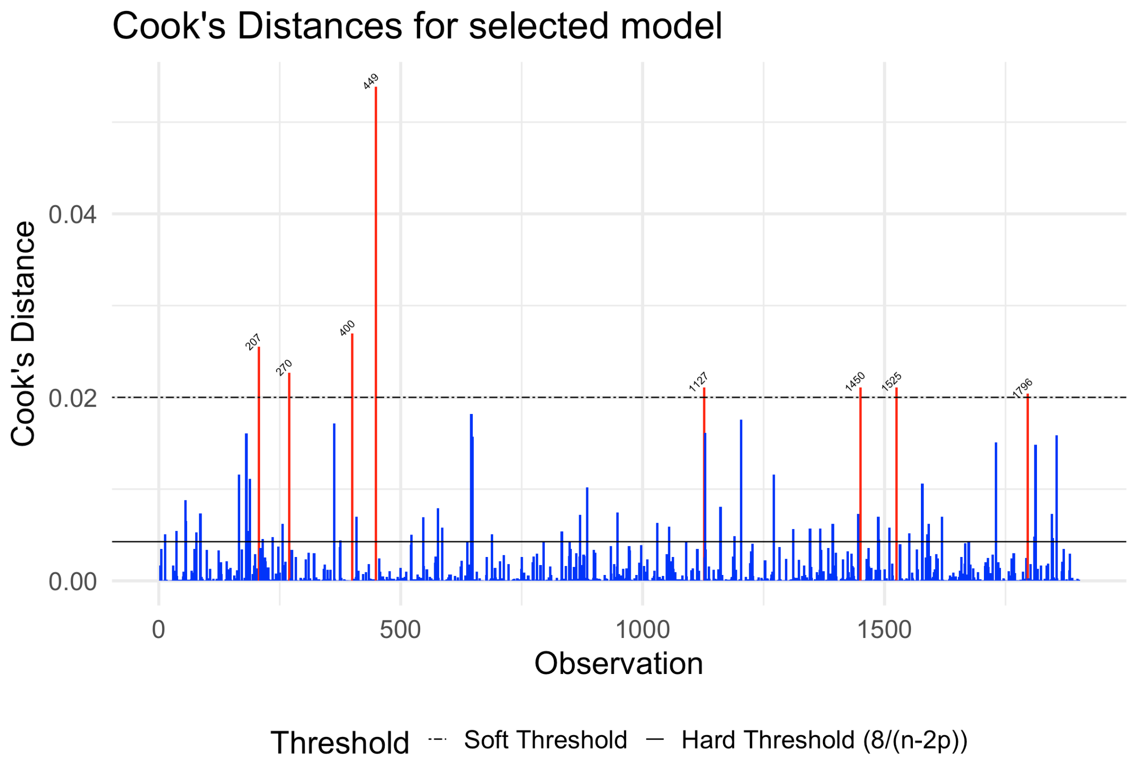

Figure 4: Cook's Distance for Influence Diagnostics in selected model. Vertical bars representing each observation's Cook's distance. The dashed lines indicate the soft and hard thresholds for identifying potentially influential points, with those exceeding the soft threshold highlighted in red.

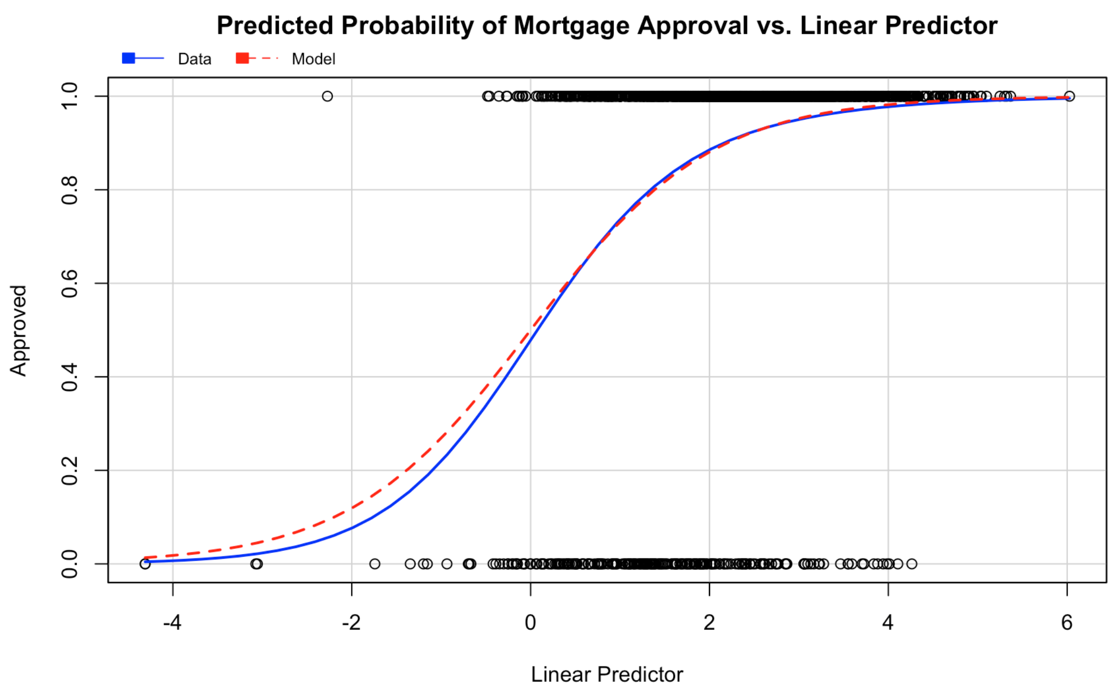

Figure 5: Calibration Curve for Mortgage Approval Predictions. Predicted probability of mortgage approval against linear predictor with the actual data points are marked in blue, while the red dashed line represents the model's predictions.

(c.p)). If the HIR increases by one percentage point, the odds of approval decrease on average by an estimated multiplicative factor of 0.95 c.p.. By considering the confidence interval, we can also make statements about the significance, as if the 95% confidence interval covers the value zero (or one if exponentiated), our effect is not significant at α-level 0.05.

For the mortgage credit score (`mcs`) variables, we exhibit varying influences. While category 1 shows a positive relationship with the odds for approval, showing that if the credit score category is 1 the estimated odds of approval increase by multiplicative factor of 1.72 (on average) compared to our reference category 2, while the odds for MCS3 and MCS4 decrease with factors 0.63 and 0.59.

For demographic variables, not being single slightly increases the odds of approval, with an OR of 1.39. It is worth mentioning that for the variables included in the interaction terms seen in table 1 we have to distinguish between interpreting effects for self-employed applicants and non self-employed applicants. We included enhancing effects on, for example, the ODIR through our interaction terms. The effect of the ODIR for self-employed applicants is therefore $\beta_{\text{odir}}+\beta_{\text{selfYes:odir}}$ and confidence interval

$$\left(\hat{\beta}_{\text{odir}}+\hat{\beta}_{\text{selfYes:odir}}\right) \pm z_{\alpha/2}\sqrt{\widehat{var}_{\text{odir}}+\widehat{var}_{\text{selfYes:odir}}+2\cdot\widehat{\text{Cov}}\left(\hat{\beta}_{\text{odir}}, \hat{\beta}_{\text{selfYes:odir}}\right)} \tag{3}$$

Note that we need to consider the estimated covariance of the coefficients when calculating this interval. The results for the coefficients of self-employed applicants can be seen in table 2, including the odds ratios, which are simply $exp(\beta_{\text{odir}}+\beta_{\text{selfYes:odir}})$. We notice that if the applicant is self-employed, the increase of one percentage point in the ODIR does have a multiplicative effect closer to an OR of one (OR = 0.969) than for applicants that are self-employed (OR = 0.917). For ethnicity, we observe that self-employed black individuals have a point estimate of 0.05 with an odds ratio of 1.051, indicating a minimal effect on approval. For non-self-employed black individuals, the estimate is notably negative (-1.340), with a very low odds ratio of 0.262. This indicates a substantial decrease in the odds of approval by an estimated multiplicative factor of 0.262 if the individual is not self-employed and black compared to not self-employed whites.

For self-employed individuals, the estimate of the effect of URIA corresponds to an odds ratio of 1.113, therefore suggesting that an increase of one percentage point in URIA results in an increase in the odds for approval by an estimated average multiplicative of 1.113. The estimated coefficient associated with self-employment alone represents the unique contribution of being self-employed to the log-odds of the outcome when all other variables in the model are held constant at their reference levels. However, interpreting the 'selfYes' effect is challenging due to its involvement in several interaction terms with continuous variables like URIA and ODIR, hence implying that the impact of being self-employed changes with varying unemployment rates and debt ratios. In this context, marginal effects become particularly insightful.

Variable	Estimate	Std. Error	95% CI	OR ($exp(\hat{\beta})$)	95% CI (OR)
(Intercept)	6.186***	0.551	(5.106, 7.266)	485.86	(164.61, 1437.82)
HIR	-0.053***	0.010	(-0.073, -0.033)	0.95	(0.93, 0.97)
ODIR	-0.087***	0.013	(-0.111, -0.062)	0.92	(0.89, 0.94)
LVR	-0.021***	0.005	(-0.030, -0.011)	0.98	(0.97, 0.99)
MCS1	0.540**	0.195	(0.157, 0.923)	1.72	(1.17, 2.52)
MCS3	-0.454	0.474	(-1.384, 0.475)	0.63	(0.25, 1.61)
MCS4	-0.529	0.604	(-1.715, 0.656)	0.59	(0.18, 1.93)
Single No	0.331*	0.153	(0.031, 0.631)	1.39	(1.03, 1.88)
Ethnic Black	-1.340***	0.179	(-1.690, -0.989)	0.26	(0.18, 0.37)
URIA	-0.105**	0.038	(-0.180, -0.030)	0.90	(0.83, 0.97)
Self Yes	-2.352***	0.497	(-3.326, -1.378)	0.095	(0.036, 0.252)
Self Yes:odir	0.054*	0.025	(0.006, 0.103)	1.06	(1.01, 1.11)
Self Yes:Black	1.389*	0.656	(0.103, 2.675)	4.01	(1.11, 14.48)
self Yes:URIA	0.212*	0.087	(0.042, 0.383)	1.24	(1.04, 1.47)

$^{***}p < 0.001$; $^{**}p < 0.01$; $^{*}p < 0.05$

Table 1: Model coefficients, standard errors, 95% confidence intervals, exponentiated coefficients (OR), and their 95% confidence intervals with significance codes

Variable	Employment	Estimate	Std. Error	95% CI	OR ($exp(\hat{\beta})$)
ODIR	*Self* = "Yes"	-0.032	0.021	(-0.074, 0.010)	0.969
	Self = "No"	-0.087***	0.013	(-0.111 -0.062)	0.917
Ethnicity Black	*Self* = "Yes"	0.05	0.633	(-1.192, 1.291)	1.051
	Self = "No"	-1.340***	0.179	(-1.690, -0.989)	0.262
URIA	*Self* = "Yes"	0.107	0.078	(-0.046, 0.261)	1.113
	Self = "No"	-0.105**	0.038	(-0.180, -0.030)	0.900

$^{***}p < 0.001$; $^{**}p < 0.01$; $^{*}p < 0.05$

Table 2: Model coefficients, standard errors, 95% confidence intervals, exponentiated coefficients (OR) for variables included in interaction terms.

3.3.1 Marginal Effects

Marginal effects are particularly valuable when interpreting complex models with interaction terms between categorical and continuous variables, hence for the interpretation of the role of self-employment, they seem worth exploring. Table 3 presents the average marginal effects (AME) for each variable in the logistic regression model estimated via the delta method, which quantifies the expected average change in the log odds of mortgage approval associated with a one-unit change in the predictor variable, while other variables are held at their fixed average values. We can now obtain the average marginal effect (AME) of being self-employed of -0.9425, which can be interpreted as follows for self-employed applicants, the odds of an approved mortgage application decrease multiplicatively by factor $exp(-0.9425) = 0.3897$ c.p. compared to a person who is not self-employed. The set of plots in figure 6 illustrates the marginal effects of the interactions in our reduced model on the probability scale, which I want to touch on shortly, in addition to our odds-scale interpretations. In the first plot, we see the predicted probability of mortgage approval as a function of ODIR, stratified by self-employment status. For non-self-employed individuals ('No'), there is a steeper decline in approval probability as ODIR increases.

Factor	AME	exp(AME)	SE	p	CI
HIR	-0.0532	0.9483	0.0101	<0.0001	(-0.073, -0.033)
LVR	-0.0206	0.9795	0.0048	<0.0001	(-0.030, -0.011)
MCS1	0.5399	1.7161	0.1954	0.0057	(0.157, 0.923)
MCS3	-0.4544	0.6348	0.4742	0.3380	(-1.384, 0.475)
MCS4	-0.5294	0.5891	0.6048	0.3815	(-1.715, 0.656)
ODIR	-0.0803	0.9230	0.0114	<0.0001	(-0.103, -0.058)
Self Yes	-0.9425	0.3897	0.2195	<0.0001	(-1.373, -0.512)
Single No	0.3307	1.3922	0.1530	0.0307	(0.031, 0.631)
URIA	-0.0806	0.9226	0.0352	0.0220	(-0.150, -0.012)
Ethnicity Black	-1.1818	0.3072	0.1748	<0.0001	(-1.524, -0.839)

Table 3: Average Marginal Effects (AME) and Exponentiated AME for Mortgage Approval Factors. AME of various factors on mortgage approval, their exponentiated forms representing odds ratios, standard errors (SE), p-values, and 95% confidence intervals (CI). Negative AME values suggest a decrease, and positive values suggest an increase in the probability of mortgage approval.

The second plot depicts the effect of the URIA on approval probability, again differentiated by self-employment status. For those not self-employed, a higher unemployment rate in the applicant's industry corresponds to a lower probability of approval.

Finally, the third plot shows the approval probability by ethnicity. Non-self-employed black individuals have a lower predicted probability of approval than their white counterparts, which is less the case for self-employed individuals.

3.4 Dispersion

We obtain a dispersion parameter estimate of 1.102 by using the formula $\widehat{\phi} = \frac{1}{n-p}\sum_{i=1}^{n}\frac{(y_i - \widehat{\mu}_i)^2}{V(\widehat{\mu}_i)}$. This difference from the assumed $\phi = 1$ suggests minor overdispersion, which means that the observed variation in the response variable is greater than the variation predicted and assumed

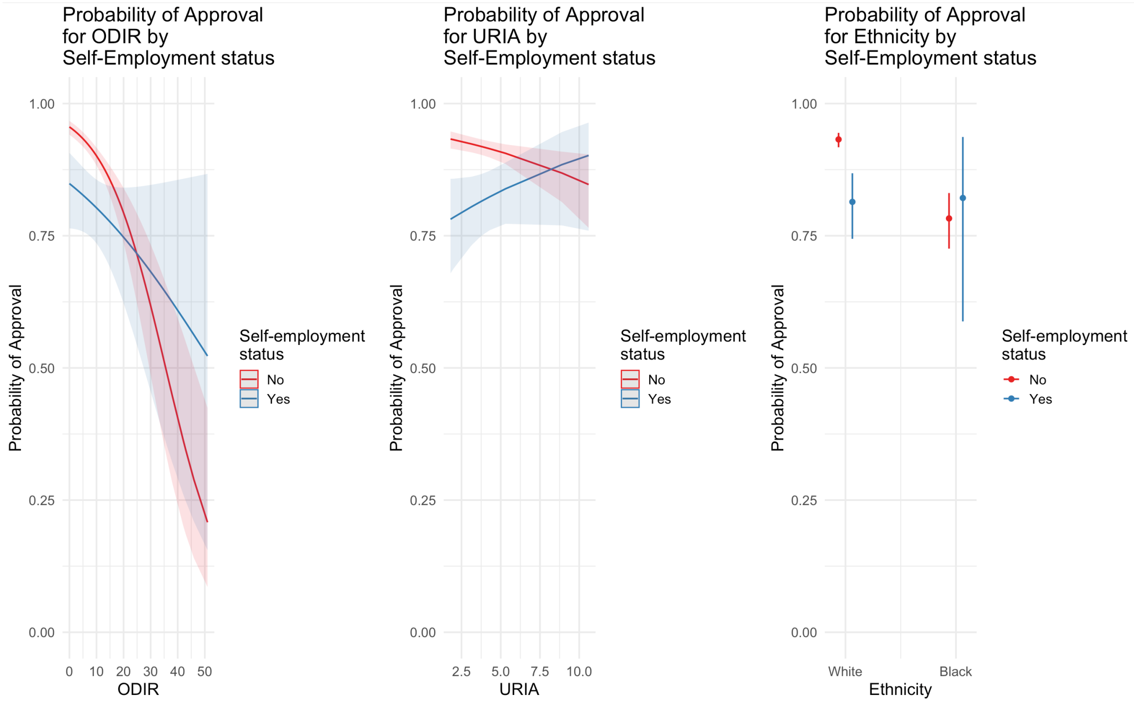

Figure 6: Conditional Effects of ODIR, URIA, and Ethnicity on Mortgage Approval Probability by Self-employment status. Lines show the predicted probability of mortgage approval at values of ODIR, URIA, and Ethnicity—focal terms in the statistical model—while holding non-focal variables constant. Ribbons show the 95% confidence intervals.

in fitting the model, leading to potentially attenuated variance estimates. It would be possible to tackle overdispersion by using a quasibinomial model, which adjusts the variance to account for the extra variation observed in the response variable.

4 Limitations & Outlook

This analysis, while detailed, encounters limitations such as slight overdispersion and the absence of variables like wealth, which may influence the precision of effect estimation. Addressing non-linear predictor-outcome relationships through generalized additive models could also refine the understanding of the data. For those seeking predictive approaches, the appendix presents methodologies like oversampling and LASSO regression as alternatives.

5 Conclusion

Our logistic regression model discerns key determinants of mortgage approval. Financial metrics like the housing expense-to-income ratio (HIR) and loan-to-value ratio (LVR) negatively impact approval odds, whereas a superior mortgage credit score markedly boosts them. Demographics influence outcomes too; singles and people of black ethnicity face disadvantages, hinting at broader issues not encapsulated by the model. The inclusion of additional data and the application of alternative modeling techniques could enhance the robustness of future models, offering a more comprehensive view of the factors influencing mortgage approvals.

A Supplementary EDA Tables

Continuous Variables				
Variable	Mean	St. Dev.	Min	Max
hir	0.255	0.079	0.010	1.100
odir	0.075	0.066	0.000	0.510
lvr	0.737	0.180	0.020	1.950
uria	3.763	2.015	1.800	10.600

Table 4: Summary of Continuous Variables

Categorical/Binary Variables			
Variable	Levels	n	%
approved	Yes	1665	87.5
	No	237	12.5
mcs	1	609	32.0
	2	1241	65.2
	3	33	1.7
	4	19	1.0
self	Yes	216	11.4
	No	1686	88.6
single	Yes	753	39.6
	No	1149	60.4
white	White	1625	85.4
	Black	277	14.6

Table 5: Summary of Categorical/Binary Variables

B Modelling

B.1 Model summary of initial model

	Estimate	Std. Error	z value	Pr(>|z|)
(Intercept)	6.0374	0.6087	9.92	<0.0001
HIR	-0.0530	0.0117	-4.53	<0.0001
ODIR	-0.0851	0.0127	-6.71	<0.0001
LVR	-0.0186	0.0054	-3.45	0.0006
MCS1	0.5379	0.2206	2.44	0.0148
MCS3	-0.6383	0.5399	-1.18	0.2371
MCS4	-0.9631	0.6376	-1.51	0.1309
Self Yes	-1.6640	1.2988	-1.28	0.2001
Single No	0.2708	0.1676	1.62	0.1061
Ethnic Black	-1.3599	0.1794	-7.58	<0.0001
URIA	-0.1015	0.0384	-2.64	0.0082
hir:selfYes	-0.0069	0.0242	-0.29	0.7742
odir:selfYes	0.0517	0.0258	2.00	0.0451
lvr:selfYes	-0.0098	0.0119	-0.82	0.4105
MCS1:selfYes	0.0613	0.4812	0.13	0.8987
MCS3:selfYes	0.7142	1.1123	0.64	0.5208
MCS4:selfYes	14.0039	428.5445	0.03	0.9739
singleNo:selfYes	0.3862	0.4269	0.90	0.3656
ethnBlack:selfYes	1.5179	0.6829	2.22	0.0262
uria:selfYes	0.2022	0.0898	2.25	0.0243

Table 6: Model summary of initial model before applying stepwise AIC selection

C Outlier Table

C.1 Soft Threshold

	approved	hir	odir	lvr	mcs	self	single	white	uria
207	No	10.00	0.00	67.00	2	Yes	Yes	White	10.60
270	No	18.00	5.00	61.00	1	Yes	No	White	10.60
400	No	31.00	25.00	57.00	3	Yes	No	White	10.60
449	Yes	44.00	51.00	55.00	3	Yes	No	White	10.60
1127	No	13.00	8.00	80.00	2	Yes	Yes	Black	3.20
1450	No	11.00	36.00	90.00	1	Yes	No	White	3.20
1525	No	6.00	12.00	80.00	4	No	No	Black	3.20
1796	No	30.00	11.00	72.00	4	No	No	White	3.20

Table 7: Observation with Cooks distance of over 0.02 in the selected model

D Further Modelling Approaches

D.1 Oversampling

Oversampled the minority class of unapproved applications, however, results should be interpreted with caution as this assumes a representative representation of the minority class in the data.

	Estimate	Std. Error	z value	Pr(>|z|)
(Intercept)	3.7498	0.2993	12.53	<0.001
hir	-0.0411	0.0060	-6.80	<0.001
odir	-0.0794	0.0068	-11.67	<0.001
lvr	-0.0184	0.0028	-6.60	<0.001
mcs1	0.4437	0.1054	4.21	<0.001
mcs3	-0.4961	0.3134	-1.58	0.1134
mcs4	-0.3584	0.4228	-0.85	0.3967
selfYes	-0.8089	0.7745	-1.04	0.2963
singleNo	0.4007	0.0858	4.67	<0.001
BlackYes	-1.3740	0.1041	-13.20	<0.001
uria	-0.1266	0.0207	-6.13	<0.001
hir:selfYes	-0.0333	0.0143	-2.34	0.0195
odir:selfYes	0.0395	0.0155	2.54	0.0110
lvr:selfYes	-0.0104	0.0078	-1.33	0.1835
mcs1:selfYes	0.2287	0.2699	0.85	0.3969
mcs3:selfYes	0.3197	0.7075	0.45	0.6513
mcs4:selfYes	14.4137	250.4227	0.06	0.9541
selfYes:singleNo	0.7702	0.2524	3.05	0.0023
selfYes:BlackYes	1.2852	0.4043	3.18	0.0015
selfYes:uria	0.1727	0.0506	3.41	0.0006

Table 8: Results for model fitted with oversampled / balanced data set

D.2 Weighted GLM

$$w(x) = \begin{cases} \frac{N}{\sum_{i=1}^{N} 1_{\{approved_i = \text{Yes}\}}} & \text{if } approved = \text{Yes} \\ \frac{N}{\sum_{i=1}^{N} 1_{\{approved_i = \text{No}\}}} & \text{if } approved = \text{No} \end{cases} \tag{4}$$

	Estimate	Std. Error	z value	Pr(>|z|)
(Intercept)	6.1859	0.1945	31.80	<0.001
hir	-0.0532	0.0036	-14.87	<0.001
odir	-0.0865	0.0044	-19.54	<0.001
lvr	-0.0206	0.0017	-12.19	<0.001
mcs1	0.5399	0.0690	7.83	<0.001
mcs3	-0.4544	0.1674	-2.71	0.0066
mcs4	-0.5294	0.2135	-2.48	0.0132
selfYes	-2.3519	0.1754	-13.41	<0.001
singleNo	0.3307	0.0540	6.12	<0.001
whiteBlack	-1.3395	0.0631	-21.22	<0.001
uria	-0.1047	0.0135	-7.74	<0.001
odir:selfYes	0.0545	0.0087	6.27	<0.001
selfYes:whiteBlack	1.3892	0.2316	6.00	<0.001
selfYes:uria	0.2122	0.0307	6.91	<0.001

Table 9: Results for weighted GLM with weights calculated as described in 4.

D.3 LASSO Regression and n-fold Cross-Validation

20-fold cross-validation to select the optimal regularization parameter, lambda. Figure D.3 visualizes the cross-validation process with the model's binomial deviance (a goodness-of-fit measure) against different values of log(lambda).

	Feature	Coefficient
1	hir	-0.02
2	odir	-0.03
3	lvr	-0.01
4	whiteBlack	-0.84
5	hir.selfYes	-0.01

Table 10: Non-zero coefficients for lambda which is one standard error above the minimum cross-validated error

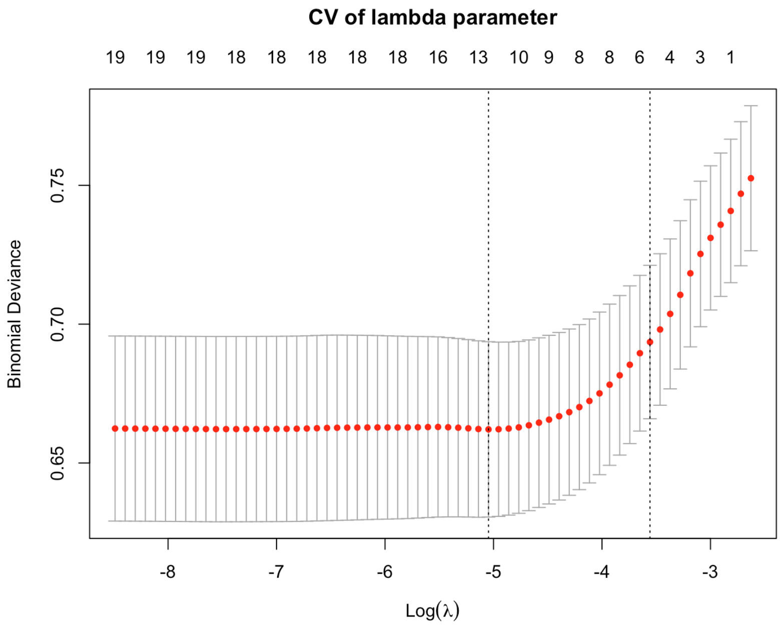

Figure 7: Cross-Validation of Lambda in LASSO Regression. Dotted lines represent the lambda values yielding minimum deviance and the most regularized model within one standard error of the minimum.

E R Code

```r
# Libraries ----
set.seed(42)
library(ggplot2)
library(gridExtra)
library(dplyr)
library(car)
library(rsq)
library(gbm)
library(boot)
library(xgboost)
library(margins)
library(ggeffects)
library(stargazer)
# Data read-in and data defintion ----
## Read and definition ----
mortg_raw <- read.csv("data/mortg.csv")
mortg <- read.csv("data/mortg.csv")
mortg$approved <- factor(mortg$approved,
                         levels = c("0", "1"),
                         labels = c("No", "Yes"))
mortg$mcs <- factor(mortg$mcs)
mortg$self <- factor(
  as.character(mortg$self),
  levels = c("1", "0"),
  labels = c("Yes", "No")
)
mortg$single <- factor(
  as.character(mortg$single),
  levels = c("1", "0"),
  labels = c("Yes", "No")
)
```

```r
mortg$white <- factor(
  as.character(mortg$white),
  levels = c("1", "0"),
  labels = c("White", "Black")
)

##Scale ratios to percentage point scale for interpretability ----
mortg$hir <- mortg$hir * 100
mortg$odir <- mortg$odir * 100
mortg$lvr <- mortg$lvr * 100

# Exploratory Data Analysis ----
## Overview ----
summary(mortg)
table(mortg$approved, mortg$self)
table(mortg$approved, mortg$single)
table(mortg$approved, mortg$white)
prop.table(table(mortg$approved, mortg$self), 1)
prop.table(table(mortg$approved, mortg$single))
prop.table(table(mortg$approved, mortg$white), 2)
prop.table(table(mortg$white, mortg$self), 2)

## Bivarite Plots ----
### Boxplots (continous data) ----
create_vio_box_plot <-
  function(data,
           variable,
           title,
           xlab = "",
           limits = c(0, 100)) {
    ggplot(data, aes_string(x = variable, y = "approved")) +
      geom_violin(
        width = 1,
        color = "black",
        fill = "darkgrey",
        alpha = 0.2
      ) +
      geom_boxplot(width = 0.1,
                   outlier.size = 1.4,
                   outlier.shape = 1) +
      labs(title = title,
           x = xlab,
           y = "Mortgage Approval") +
      coord_flip() +
      theme_minimal(base_size = 15) +
      scale_x_continuous(labels = scales::percent_format(scale = 1),
                         limits = limits)
  }
create_scaled_stacked_bar_plot <-
  function(data,
           category,
           xlab = category,
           ylab = "Percentage",
           fill = c("#F05039", "#7CA1CC"),
           title =  paste("Mortgage Approval by", category, "Status")) {
    data %>%
      group_by(!!sym(category), approved) %>%
      summarise(Count = n()) %>%
```

```
      mutate(Total = sum(Count)) %>%
      mutate(Percent = (Count / Total) * 100) %>%
      ggplot(aes_string(x = category,
                        y = "Percent", fill = "factor(approved)")) +
      geom_bar(stat = "identity", position = "fill") +
      labs(
        title = title,
        x = xlab,
        y = ylab,
        fill = "Approved"
      ) +
      scale_fill_manual(values = fill) +
      scale_y_continuous(labels = scales::percent_format()) +
      theme_minimal(base_size = 15)
  }


plot_hir <-
  create_vio_box_plot(mortg,
                      "hir",
                      "HIR\nvs Mortgage Approval",
                      xlab = "Housing Expense-to-Income Ratio",
                      limits = c(0, 120))

plot_odir <-
  create_vio_box_plot(
    mortg,
    "odir",
    "ODIR\nvs Mortgage Approval",
    xlab = "Other Debt-to-Income Ratio",
    limits = c(0, 60)
  )
plot_lvr <-
  create_vio_box_plot(mortg,
                      "lvr",
                      "LVR\nvs Mortgage Approval",
                      xlab = "Loan-to-Value Ratio",
                      limits = c(0, 200))
plot_uria <-
  create_vio_box_plot(
    mortg,
    "uria",
    "URIA\nvs Mortgage Approval",
    xlab = "Unemployment Rate in Applicant's Industry",
    limits = c(0, 12)
  )
plot_self <-
  create_scaled_stacked_bar_plot(mortg,
                                 "self",
                                 xlab = "Self-Employed",
                                 ylab = "Proportion",
                       title = "Mortgage Approval by\nSelf-Employment Status")
plot_single <-
  create_scaled_stacked_bar_plot(mortg,
                                 "single",
                                 xlab = "Single",
                                 ylab = "Proportion",
                       title = "Mortgage Approval by\nMarital Status (Single)")
```

```
plot_white <-
  create_scaled_stacked_bar_plot(mortg,
                                 "white",
                                 xlab = "Ethnicity",
                                 ylab = "Proportion",
                                 title = "Mortgage Approval by\nEthnicity")
plot_mcs <-
  create_scaled_stacked_bar_plot(mortg,
                                 "mcs",
                                 xlab = "Mortgage Credit Score",
                                 ylab = "Proportion",
                         title = "Mortgage Approval by\nMortgage Credit Score")

grid.arrange(plot_hir, plot_odir, plot_lvr, plot_uria, ncol = 2)
grid.arrange(plot_self, plot_single, plot_white, plot_mcs, ncol = 2)

## Decile Plots ----
mortg_unscaled <- mortg_raw
plot_approval_rate_by_quantiles <-
  function(data,
     variable_name,
     xlab = paste(variable_name, "Deciles"),
     title = paste("Approval Rate by", toupper(variable_name), "Quantiles")) {
    # Compute the quantile bins for the variable
    data <- data %>%
      mutate(quantile_bin = findInterval(.data[[variable_name]],
                                   quantile(.data[[variable_name]],
                                            probs = seq(0.1, 1, 0.1)),
                                   left.open = TRUE) + 1)

    # Calculate the mean approval rate by quantile bin
    approval_rate <- data %>%
      group_by(quantile_bin) %>%
      summarise(approval_rate = mean(approved, na.rm = TRUE)) %>%
      ungroup()

    # Create the bar plot
    ggplot(approval_rate, aes(x = as.factor(quantile_bin), y = approval_rate)) +
      geom_bar(stat = "identity",
               colour = "grey",
               position = position_dodge()) +
      scale_x_discrete(limits = as.character(1:10), drop = FALSE) +
      labs(title = title,
           x = xlab,
           y = "Approval Rate") +
      scale_y_continuous(labels = scales::percent_format()) +
      theme_minimal(base_size = 15)
  }



decile_approval_lvr <-
  plot_approval_rate_by_quantiles(mortg_raw, "lvr",
                              xlab = "Loan to value ratio Deciles")
decile_approval_hir <-
  plot_approval_rate_by_quantiles(mortg_raw, "hir",
                              xlab = "Housing Expenses to Income Ratio Deciles")
decile_approval_odir <-
```

```r
  plot_approval_rate_by_quantiles(mortg_raw, "odir",
                              xlab = "Other Monthly Debt Ratio Deciles")
decile_approval_uria <-
  plot_approval_rate_by_quantiles(mortg_raw, "uria",
                              xlab = "Unemployment Rate Industry Deciles")

grid.arrange(
  decile_approval_lvr,
  decile_approval_hir,
  decile_approval_odir,
  decile_approval_uria,
  ncol = 2
)

# Modelling ----
## Set reference categories ----
mortg$mcs <-
  relevel(mortg$mcs, 2) # MCS 2 as reference category (rc)
mortg$self <- relevel(mortg$self, 2) # Not self-employed as rc
mortg$single <- relevel(mortg$single, 1) # Single Yes as rc
mortg$white <- relevel(mortg$white, 1) # white as rc

## Variable selection stepwise AIC ----
full_model <-
  glm(approved ~ . * self,
      family = binomial(link = "logit"),
      data = mortg)
summary(full_model)
full_model_summary <- summary(full_model)
reduced_model <- step(full_model, direction = "both")
summary(reduced_model)
reduced_model_summary <- summary(reduced_model)
## Model selection Analysis of Deviance ----
### LRT ----
D_full <- full_model$deviance
D_reduced <- reduced_model$deviance
p_full <- length(full_model$coefficients)
p_reduced <-
  length(reduced_model$coefficients)  # number of parameters in reduced model
deviance_change <-  D_reduced - D_full
pchisq(deviance_change,
       df = p_full - p_reduced,
       lower.tail = FALSE)
### via ANOVA function ----
anova(reduced_model, full_model, test = "Chisq")

# Model Diagnosis ----
# Cooks Distance ----
plot_cooks_distance <-
  function(model,
           model_name,
           alpha = 0.1,
           threshold = NULL) {
    # Calculate Cook's distance
    cooks_d <- cooks.distance(model)

    # If threshold is not set, use the default value
    if (is.null(threshold)) {
```

```r
      threshold <- 4 / length(cooks_d)
    }

    # Create a data frame for plotting
    plot_data <- data.frame(
      Observation = 1:length(cooks_d),
      CooksDistance = cooks_d,
      AboveThreshold = cooks_d > threshold
    )

    # Create the plot using ggplot2
    ggplot(plot_data, aes(x = Observation, y = CooksDistance)) +
      geom_point(aes(color = AboveThreshold), alpha = alpha) +
      geom_point(
        data = subset(plot_data, AboveThreshold),
        aes(color = AboveThreshold),
        alpha = 1
      ) + # Points above threshold with full opacity
      scale_color_manual(values = c("FALSE" = "blue", "TRUE" = "red"),
                         guide = FALSE) +
      ggtitle(paste("Cook's Distance for", model_name)) + # Title
      xlab('Observation') + # X-axis label
      ylab("Cook's Distance") + # Y-axis label
      geom_text(
        data = subset(plot_data, AboveThreshold),
        aes(label = Observation),
        vjust = -0.5,
        hjust = "inward",
        size = 3,
        angle = 45
      ) + # Label high points
      geom_hline(yintercept = threshold,
                 linetype = 'dotted',
                 col = 'black') +
      geom_hline(
        yintercept = 4 / length(cooks_d),
        linetype = 'dotted',
        col = 'pink'
      )
  }

plot_cooks_distance_barplot <-
  function(model,
           model_name,
           threshold = 4 / length(cooks.distance(model))) {
    # Calculate Cook's distance
    cooks_d <- cooks.distance(model)

    # Create a data frame for plotting
    plot_data <- data.frame(Observation = 1:length(cooks_d),
                            CooksDistance = cooks_d)

    # Create the bar plot using ggplot2
    ggplot(plot_data, aes(x = Observation, y = CooksDistance)) +
      geom_bar(stat = "identity",
               aes(fill = CooksDistance > threshold),
               width = 5) +
      scale_fill_manual(values = c("blue", "red"), guide = FALSE) +
```

```r
      ggtitle(paste("Cook's Distances for selected model")) +
      xlab("Observation") +
      ylab("Cook's Distance") +
      geom_text(
        data = subset(plot_data, CooksDistance > threshold),
        aes(label = Observation),
        vjust = -0.5,
        size = 3,
        angle = 45
      ) +
      geom_hline(aes(yintercept = threshold, linetype = "Soft threshold"),
                 color = 'black') +
      geom_hline(aes(
        yintercept = 8 / (length(cooks_d) - 2 * length(coef(reduced_model))),
        linetype = "Hard threshold"
      ),
      color = 'black') +
      scale_linetype_manual(
        name = "Threshold",
        values = c("Soft threshold" = "twodash", "Hard threshold" = "solid"),
        breaks = c("Soft threshold", "Hard threshold"),
        labels = c("Soft Threshold", "Hard Threshold (8/(n-2p))")
      ) +
      theme_minimal(base_size = 25) +
      theme(legend.position = "bottom")
  }

plot_cooks_distance(reduced_model, "Reduced", 0.1, 0.02)
plot_cooks_distance_barplot(reduced_model, "Reduced", 0.02)
cooks_d <- cooks.distance(reduced_model)


# Calibration Curve ----
marginalModelPlot(reduced_model,
                  ylab = "Approved",
                  cex.lab = 15)
title(main = "Predicted Probability of Mortgage Approval vs. Linear Predictor")
# Model results ----
# Model table ----
summary(reduced_model)

# Confidence Intervals ----
confint.default(reduced_model)
exp(confint.default(reduced_model))
## Multiple Confidence Intervals -----
calculate_interac_confint <-
  function(model,
           var1,
           var2 = "selfYes",
           sig.level = 0.05) {
    # Extract variance-covariance matrix from the model
    coefs_var <- vcov(model)
    # Extract the coefficients
    coefs <- coef(model)
    # Calculate the z-value based on the significance level
    z <- qnorm(1 - sig.level / 2)

    # Calculate the standard error for the interaction term
```

```r
    # considering the covariance
    se <-
      sqrt(coefs_var[var1, var1] + coefs_var[var2, var2]
           + 2 * coefs_var[var1, var2])
    ci_lower <- coefs[var1] + coefs[var2] - z * se
    ci_upper <- coefs[var1] + coefs[var2] + z * se
    # Return the confidence interval

    return(list(
      "confint" = c(lower = ci_lower, upper = ci_upper),
      "se" = se
    ))
  }

calculate_interac_pv <- function(model, var1, var2) {
  # Extract variance-covariance matrix from the model
  coefs_var <- vcov(model)
  # Extract the coefficients
  coefs <- coef(model)

  # Calculate the standard error for the interaction term
  # considering the covariance
  se <-
    sqrt(coefs_var[var1, var1] + coefs_var[var2, var2]
         + 2 * coefs_var[var1, var2])

  p <- 2 * (1 - pnorm(abs((coefs[var1] + coefs[var2]) / se)))
  names(p) <- paste(var1, var2, sep = "+")
  # Return the confidence interval
  return(p)
}

### odir:selfYes ----
round(sum(coef(reduced_model)[c("odir", "odir:selfYes")]), 3)
calculate_interac_confint(
  model = reduced_model,
  var1 = "odir",
  var2 = "odir:selfYes",
  sig.level = 0.05
)
calculate_interac_pv(model = reduced_model,
                     var1 = "odir", var2 = "odir:selfYes")
### selfYes:whiteBlack ----
round(sum(coef(reduced_model)[c("whiteBlack", "selfYes:whiteBlack")]), 3)
calculate_interac_confint(
  model = reduced_model,
  var1 = "whiteBlack",
  var2 = "selfYes:whiteBlack",
  sig.level = 0.05
)
calculate_interac_pv(model = reduced_model,
                     var1 = "whiteBlack", var2 = "selfYes:whiteBlack")
### selfYes:uria ----
round(sum(coef(reduced_model)[c("uria", "selfYes:uria")]), 3)
calculate_interac_confint(
  model = reduced_model,
  var1 = "uria",
  var2 = "selfYes:uria",
```

```r
  sig.level = 0.05
)
calculate_interac_pv(model = reduced_model,
                     var1 = "uria", var2 = "selfYes:uria")



# Marginal Effects ----
marginal_effects <-
  margins(
    reduced_model,
    vce = "delta",
    vcov = vcov(reduced_model),
    type = "link"
  )

## Summary of marginal effects ----
summary(marginal_effects) %>% as.data.frame()


marginal_effects_df <- summary(marginal_effects) %>% as.data.frame()

marginal_effects_df <- marginal_effects_df %>%
  mutate(AME_exp = exp(AME))



marginal_effects_df <- marginal_effects_df %>%
  mutate(p = 2 * (1 - pnorm(abs(z))),
         p_rounded = round(2 * (1 - pnorm(abs(
           z
         ))), 4))

## AME Plot ----
ggplot(marginal_effects_df, aes(x = factor, y = AME)) +
  geom_point() +
  geom_errorbar(aes(ymin = lower, ymax = upper), width = 0.2) +
  coord_flip() +  # Flip coordinates to have factors on the y-axis
  labs(title = "Average Marginal Effects (AME) with Confidence Intervals",
       x = "Average Marginal Effect",
       y = "Factor") +
  theme_minimal()


## Effects on Prob Scale----
### Single effects ----
meff_self <- ggpredict(reduced_model, terms = c("self")) |>
  plot() +
  ylim(c(0, 1))
meff_white <- ggpredict(reduced_model, terms = c("white")) |>
  plot() +
  ylim(c(0, 1))
meff_single <- ggpredict(reduced_model, terms = c("single")) |>
  plot() +
  ylim(c(0, 1))
meff_mcs <- ggpredict(reduced_model, terms = c("mcs")) |>
  plot() +
  ylim(c(0, 1))
```

```r
meff_odir <- ggpredict(reduced_model, terms = c("odir[all]")) |>
  plot() +
  ylim(c(0, 1))
meff_lvr <- ggpredict(reduced_model, terms = c("lvr[all]")) |>
  plot() +
  ylim(c(0, 1))
meff_hir <- ggpredict(reduced_model, terms = c("hir[all]")) |>
  plot() +
  ylim(c(0, 1))
meff_uria <- ggpredict(reduced_model, terms = c("uria[all]")) |>
  plot() +
  ylim(c(0, 1))

grid.arrange(
  meff_odir,
  meff_lvr,
  meff_hir,
  meff_uria,
  meff_self,
  meff_white,
  meff_single,
  meff_mcs,
  ncol = 4
)

## Interaction Plots ----
meff_odir_self <-
  ggeffect(reduced_model, terms = c("odir[all]", "self")) |>
  plot() +
  ylim(c(0, 1)) +
  geom_line() +
  xlab("ODIR") +
  ylab("Probability of Approval") +
  labs(colour = "Self-employment\nstatus") +
  ggtitle("Probability of Approval\nfor ODIR by\nSelf-Employment status") +
  theme_minimal(base_size = 15)


meff_uria_self <-
  ggeffect(reduced_model, terms = c("uria[all]", "self")) |>
  plot() +
  ylim(c(0, 1)) +
  xlab("URIA") +
  ylab("Probability of Approval") +
  labs(colour = "Self-employment\nstatus") +
  ggtitle("Probability of Approval\nfor URIA by\nSelf-Employment status") +
  theme_minimal(base_size = 15)


meff_white_self <-
  ggeffect(reduced_model, terms = c("white[all]", "self")) |>
  plot() +
  ylim(c(0, 1)) +
  xlab("Ethnicity") +
  ylab("Probability of Approval") +
  labs(colour = "Self-employment\nstatus") +
  ggtitle("Probability of Approval\nfor Ethnicity by\nSelf-Employment status") +
  theme_minimal(base_size = 15)
```

```r

grid.arrange(meff_odir_self, meff_uria_self, meff_white_self, ncol = 3)

# Dispersion Parameter ----
E2 <- resid(reduced_model, type = "pearson")
N  <- nrow(mortg)
p  <- length(coef(reduced_model))
sum(E2 ^ 2) / (N - p)

check_overdispersion <- function(logit_model) {
  residual_df <- df.residual(logit_model)
  pearson_resid <- residuals(logit_model, type = "pearson")
  chi_squared <- sum(pearson_resid ^ 2)
  dispersion_ratio <- chi_squared / residual_df
  p_value <-
    pchisq(chi_squared, df = residual_df, lower.tail = FALSE)
  c(
    chi_sq = chi_squared,
    disp_ratio = dispersion_ratio,
    res_df = residual_df,
    p_val = p_value
  )
}

round(check_overdispersion(reduced_model), 5)

#Appendix ---
## Oversampling ----
# Splitting the data into majority and minority

minority_data <- mortg[mortg$approved == "No", ]
majority_data <- mortg[mortg$approved == "Yes", ]

# Oversampling minority class
oversampled_minority <-
  minority_data[sample(nrow(minority_data), nrow(majority_data),
                       replace = TRUE), ]

# Combine back with majority class
balanced_data <- rbind(majority_data, oversampled_minority)
balanced_full_model <-
  glm(approved ~ . * self,
      family = binomial(link = 'logit'),
      data = balanced_data)
summary(balanced_full_model)
balanced_reduced_model <-
  glm(reduced_model$formula,
      family = binomial(link = 'logit'),
      data = balanced_data)
summary(balanced_reduced_model)

## Lasso Regression ----
model_lasso <- glmnet::glmnet(
  x = model.matrix( ~ . * self, data = mortg[, -1]),
  y = model.frame(mortg) |> model.response(),
  alpha = 1,
  family = "binomial"
)
```

```r

lasso_cv <- glmnet::cv.glmnet(
  x = model.matrix( ~ . * self, data = mortg[, -1]),
  y = model.frame(mortg) |> model.response(),
  alpha = 1,
  nfolds = 20,
  family = "binomial"
)
plot(lasso_cv)
lasso_coef <-
  coef(model_lasso, s = lasso_cv$lambda.1se , digits = 0.3)
lasso_coef

lasso_coef <- coef(model_lasso, s = lasso_cv$lambda.1se)

coef_df <- as.data.frame(Matrix::as.matrix(lasso_coef))
names(coef_df) <- c("Coefficient")
coef_df$Feature <- row.names(coef_df)
coef_df <- coef_df[-c(1:2),]  # remove the intercept row

# Filter out zero coefficients for a cleaner plot
coef_df <- coef_df[coef_df$Coefficient != 0,]
row.names(coef_df) <- NULL

ggplot(coef_df, aes(x = Feature, y = Coefficient)) +
  geom_hline(yintercept = 0,
             color = "red",
             linetype = "dashed") +
  geom_point() +
  coord_flip() +  # Flip the axes to make it horizontal
  labs(x = "Features", y = "LASSO Coefficients",
       title = "Non-zero LASSO Coefficients at lambda.1se") +
  theme_minimal()

## Weighted Logistic Regression ----
weights <- ifelse(
  mortg$approved == 1,
  nrow(mortg) / sum(mortg$approved == "Yes"),
  nrow(mortg) / sum(mortg$approved == "No")
)

weighted_full_model <-
  glm(
    approved ~ . * self,
    family = binomial(link = "logit"),
    data = mortg,
    weights = weights
  )
summary(weighted_full_model)
weighted_reduced_model <-
  glm(
    approved ~ hir + odir + lvr + mcs + self + single +
      odir * self + self * white + self * uria,
    family = binomial(link = "logit"),
    data = mortg,
    weights = weights
  )
summary(weighted_reduced_model)
```