Analysis of Mortgage Applications

Markus Salomon 23.04.2019

Summary

This document describes the process and the results of the analysis of mortgage application data. The aim of the analysis is to establish a relationship between the data on the applicant and the loan (these will be called *features*) and the information whether the loan was accepted or declined (which will be called the *label*).

By the means of calculating statistics and visual inspection of plots, several correlations and patterns were identified in the data. These could point to causal relations or interactions between the features amongst each other and between the features and the label. Furthermore, these techniques revealed properties of the distribution of the data, which were used to identify outliers and clean the data, so the assumed relation between the features and the label could be made more distinct and visible the algorithms, that were used to determine it mathematically, as well as to the reader.

The most important information on whether a loan request is rejected or accepted is encoded in

- **Lender**: The institution that gives the loan
- **Purpose**: Information on what the money will be used for
- **Loan amount**: How big the loan is as well as its size relative to the financial resources of the applicant
- **Applicant Data**: Such as income, and if there is a co-applicant or not, and the race (note ethical considerations in practical use)

Geographical and census information on the applicant contain some insight as well.

Data Background

The data was published by the Federal Financial Institutions Examinations Council (FFIEC). It was gathered for the fulfillment of regulatory requirements (more exactly the Home Mortgage Disclosure Act). It contains information on the applicant and the respective loan, and the information whether the request for the loan was accepted or declined.

Environment and Tools

All technical accessories needed to execute the analysis were implemented in Python. For quick and interactive working with the data, the Jupyter Notebook was utilized.

The main packages that were used are:

- Pandas [1] for loading, storing and manipulating the dataset (and some basic plots)
- Numpy [2] for scientific computing
- Seaborn [3] for visualization
- Sklearn [4] for preprocessing, decomposition and machine learning prototyping

For an initial overview and very quick prototyping the data was loaded in WEKA [5], as this tool provides very fast initial plots and statistics in minutes without writing any code.

All work was performed on basic hardware (Windows 10, Intel I5-7300, 16GB Ram). No special high-performing hardware was leveraged.

The final data transformation pipeline and machine learning model was implemented and evaluated in an Azure Machine Learning Studio Experiment [6].

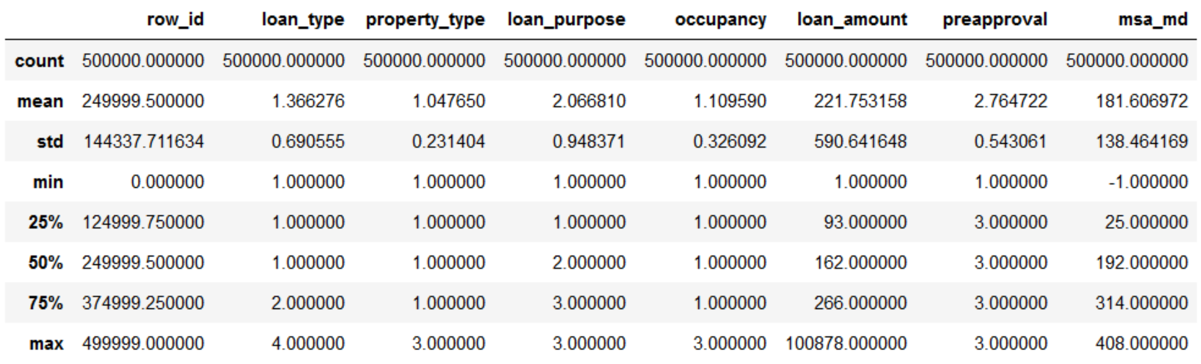
Data Exploration

Analysis of the feature distributions

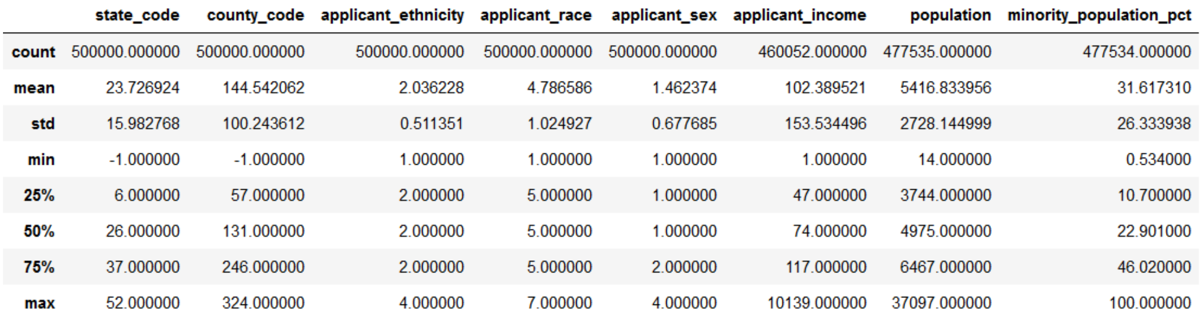
The file *train_values.csv* contains the features. Their meaning is described in detail in the Code Sheet provided by the FFIEC [7].

Including a unique identifier (*row_id*) the dataset contains 21 features. All of them are given as numeric values, but it is important to observe, that some of them are categorial features, which were already encoded as numbers. The feature *co_applicant* is encoded with (*True*, *False*).

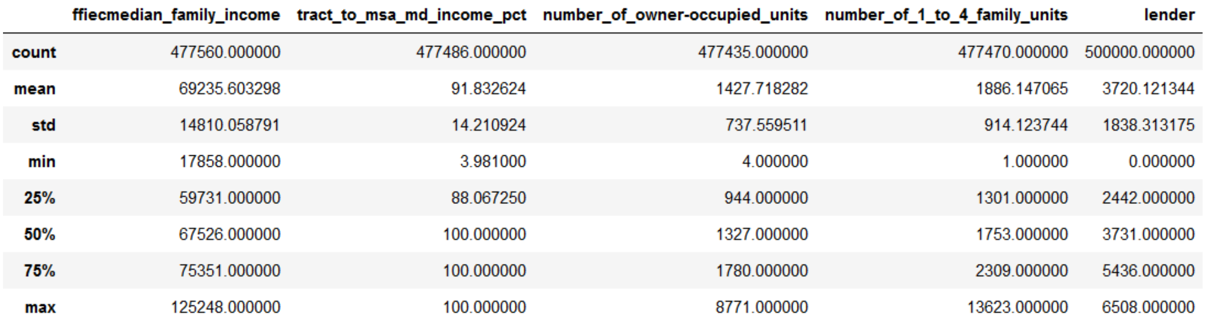
First, we calculate some basic column-wise summary statistics on the data.



	row_id	loan_type	property_type	loan_purpose	occupancy	loan_amount	preapproval	msa_md
count	500000.000000	500000.000000	500000.000000	500000.000000	500000.000000	500000.000000	500000.000000	500000.000000
mean	249999.500000	1.366276	1.047650	2.066810	1.109590	221.753158	2.764722	181.606972
std	144337.711634	0.690555	0.231404	0.948371	0.326092	590.641648	0.543061	138.464169
min	0.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	-1.000000
25%	124999.750000	1.000000	1.000000	1.000000	1.000000	93.000000	3.000000	25.000000
50%	249999.500000	1.000000	1.000000	2.000000	1.000000	162.000000	3.000000	192.000000
75%	374999.250000	2.000000	1.000000	3.000000	1.000000	266.000000	3.000000	314.000000
max	499999.000000	4.000000	3.000000	3.000000	3.000000	100878.000000	3.000000	408.000000



	state_code	county_code	applicant_ethnicity	applicant_race	applicant_sex	applicant_income	population	minority_population_pct
count	500000.000000	500000.000000	500000.000000	500000.000000	500000.000000	460052.000000	477535.000000	477534.000000
mean	23.726924	144.542062	2.036228	4.786586	1.462374	102.389521	5416.833956	31.617310
std	15.982768	100.243612	0.511351	1.024927	0.677685	153.534496	2728.144999	26.333938
min	-1.000000	-1.000000	1.000000	1.000000	1.000000	1.000000	14.000000	0.534000
25%	6.000000	57.000000	2.000000	5.000000	1.000000	47.000000	3744.000000	10.700000
50%	26.000000	131.000000	2.000000	5.000000	1.000000	74.000000	4975.000000	22.901000
75%	37.000000	246.000000	2.000000	5.000000	2.000000	117.000000	6467.000000	46.020000
max	52.000000	324.000000	4.000000	7.000000	4.000000	10139.000000	37097.000000	100.000000



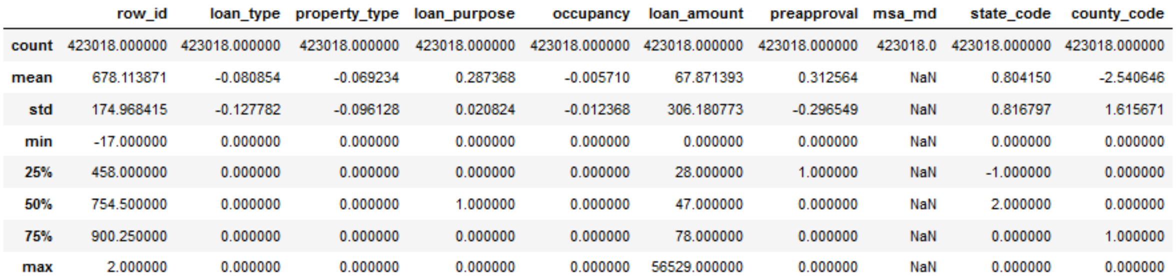
	ffiecmedian_family_income	tract_to_msa_md_income_pct	number_of_owner-occupied_units	number_of_1_to_4_family_units	lender
count	477560.000000	477486.000000	477435.000000	477470.000000	500000.000000
mean	69235.603298	91.832624	1427.718282	1886.147065	3720.121344
std	14810.058791	14.210924	737.559511	914.123744	1838.313175
min	17858.000000	3.981000	4.000000	1.000000	0.000000
25%	59731.000000	88.067250	944.000000	1301.000000	2442.000000
50%	67526.000000	100.000000	1327.000000	1753.000000	3731.000000
75%	75351.000000	100.000000	1780.000000	2309.000000	5436.000000
max	125248.000000	100.000000	8771.000000	13623.000000	6508.000000

We can immediately draw several conclusions from these:

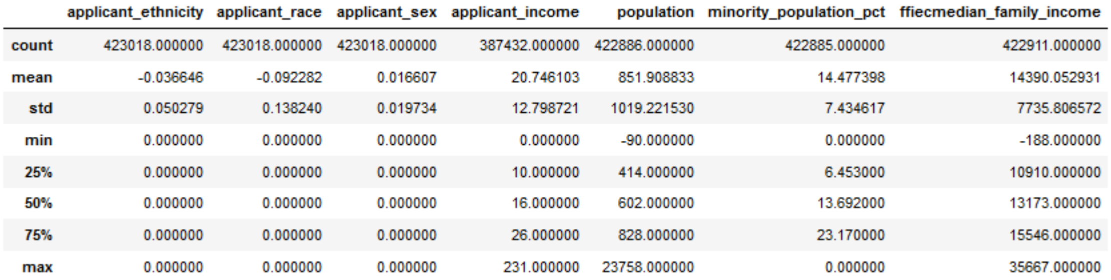
- The summary statistics on *row_id* show that this field contains indeed an incremental unique identifier that starts at 0 and ends at 499999
- From the *count* values we see that in some columns there are missing values, as the count is smaller than 500000. These columns contain census information (*applicant_income, population, minority_population_pct, ffiecmedian_family_income, tract_to_msa_md_income_pct, number_of_owner-occupied_units, number_of_1_to_4_family_units*). Missing values in the other fields were encoded by the value -*1*. We can identify columns that contain such values in the *min* statistic. Additional analysis of the rows with missing values shows, that the rows with missing values encoded as -1 are all a subset of the rows that miss the value of *msa_md.*
- The relatively small mean and mode, as well as standard deviation but in comparison large maximum value in *loan_amount, applicant_income* and *population* hint to outliers. This will become more obvious later during the visual inspection of the data.

It will be important to treat the missing values and outliers appropriately before building a machine learning model. At this point they are only revealed, and their existence is noted.

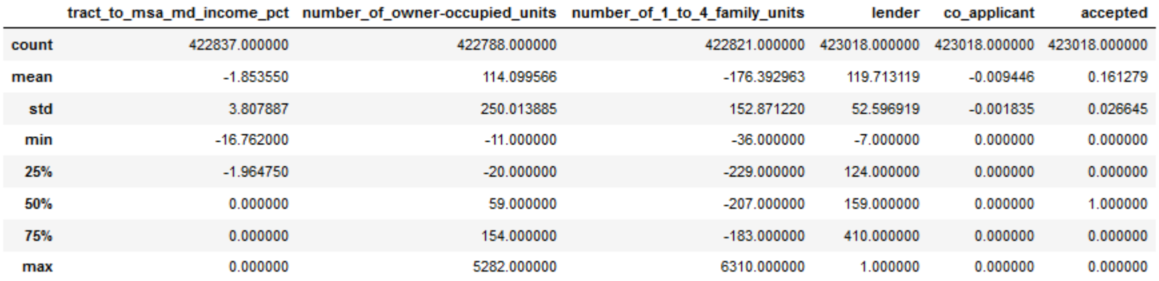
We can determine, how much the distribution of the part with the missing values differs from the distribution of the whole dataset. In order to quantify this, we subtract the summary statistics, calculated only on the subset with missing values from the summary statistics of the whole dataset. This way we can quantify the deviation. If the values are missing completely at random, the subset with missing values would have the same distribution as the whole dataset. And we could just drop the rows with missing values.



	row_id	loan_type	property_type	loan_purpose	occupancy	loan_amount	preapproval	msa_md	state_code	county_code
count	423018.000000	423018.000000	423018.000000	423018.000000	423018.000000	423018.000000	423018.000000	423018.0	423018.000000	423018.000000
mean	678.113871	-0.080854	-0.069234	0.287368	-0.005710	67.871393	0.312564	NaN	0.804150	-2.540646
std	174.968415	-0.127782	-0.096128	0.020824	-0.012368	306.180773	-0.296549	NaN	0.816797	1.615671
min	-17.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	NaN	0.000000	0.000000
25%	458.000000	0.000000	0.000000	0.000000	0.000000	28.000000	1.000000	NaN	-1.000000	0.000000
50%	754.500000	0.000000	0.000000	1.000000	0.000000	47.000000	0.000000	NaN	2.000000	0.000000
75%	900.250000	0.000000	0.000000	0.000000	0.000000	78.000000	0.000000	NaN	0.000000	1.000000
max	2.000000	0.000000	0.000000	0.000000	0.000000	56529.000000	0.000000	NaN	0.000000	0.000000



	applicant_ethnicity	applicant_race	applicant_sex	applicant_income	population	minority_population_pct	ffiecmedian_family_income
count	423018.000000	423018.000000	423018.000000	387432.000000	422886.000000	422885.000000	422911.000000
mean	-0.036646	-0.092282	0.016607	20.746103	851.908833	14.477398	14390.052931
std	0.050279	0.138240	0.019734	12.798721	1019.221530	7.434617	7735.806572
min	0.000000	0.000000	0.000000	0.000000	-90.000000	0.000000	-188.000000
25%	0.000000	0.000000	0.000000	10.000000	414.000000	6.453000	10910.000000
50%	0.000000	0.000000	0.000000	16.000000	602.000000	13.692000	13173.000000
75%	0.000000	0.000000	0.000000	26.000000	828.000000	23.170000	15546.000000
max	0.000000	0.000000	0.000000	231.000000	23758.000000	0.000000	35667.000000

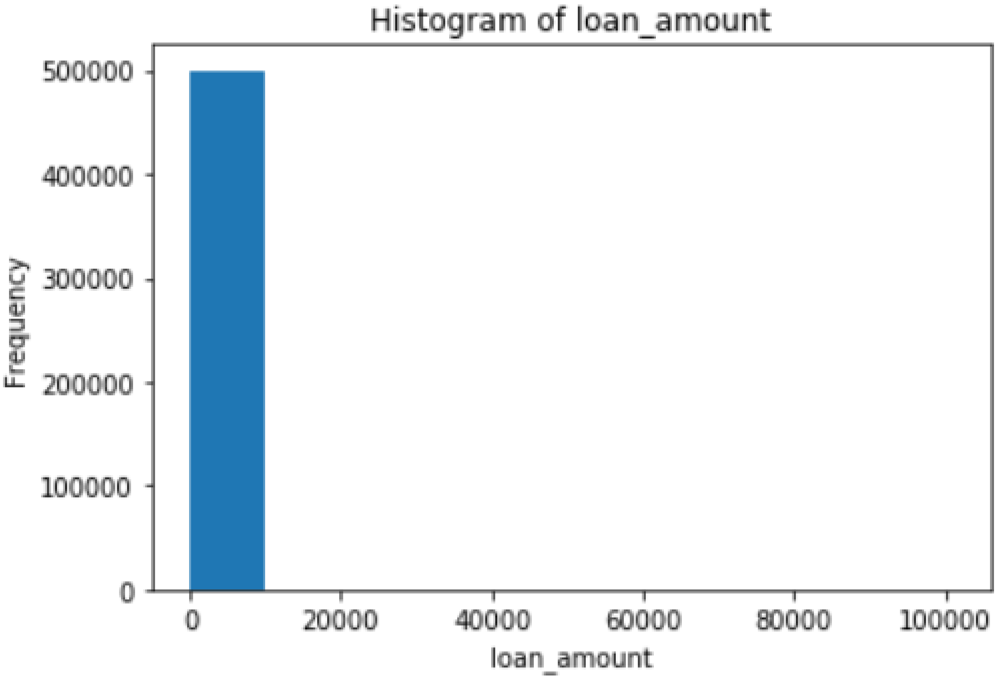


	tract_to_msa_md_income_pct	number_of_owner-occupied_units	number_of_1_to_4_family_units	lender	co_applicant	accepted
count	422837.000000	422788.000000	422821.000000	423018.000000	423018.000000	423018.000000
mean	-1.853550	114.099566	-176.392963	119.713119	-0.009446	0.161279
std	3.807887	250.013885	152.871220	52.596919	-0.001835	0.026645
min	-16.762000	-11.000000	-36.000000	-7.000000	0.000000	0.000000
25%	-1.964750	-20.000000	-229.000000	124.000000	0.000000	0.000000
50%	0.000000	59.000000	-207.000000	159.000000	0.000000	1.000000
75%	0.000000	154.000000	-183.000000	410.000000	0.000000	0.000000
max	0.000000	5282.000000	6310.000000	1.000000	0.000000	0.000000

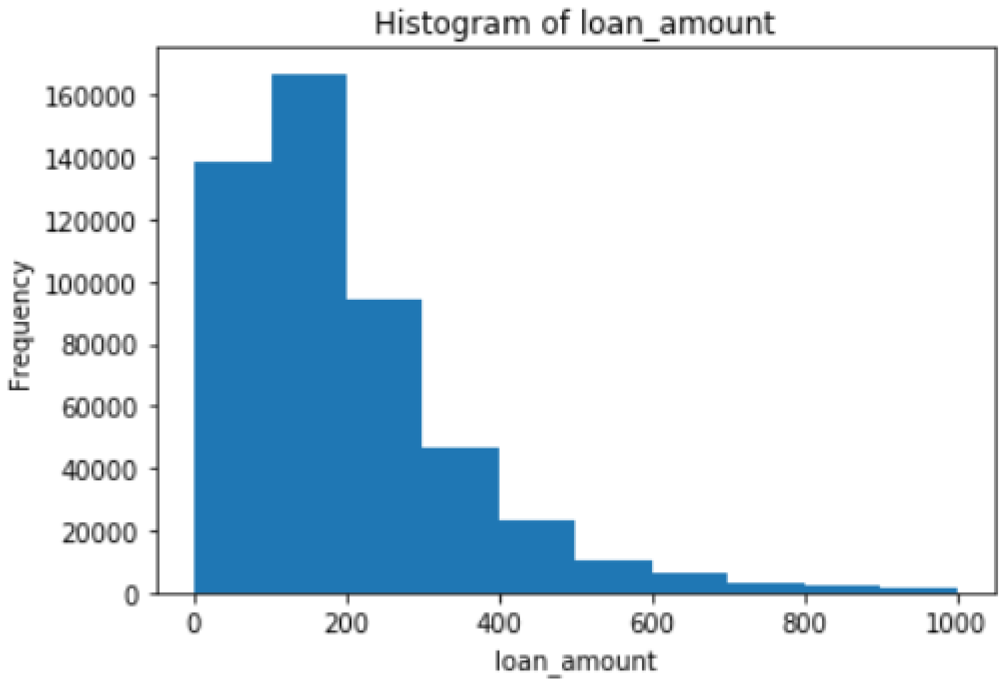
We see that the missing values are not evenly distributed, but the part with missing values has a notably different distribution than the whole dataset. Large differences are present in *loan_amount*, *applicant_income* and *population*. The differences in the census columns originate from the fact, that these are the rows, where most values are missing in the part of the population where any value is missing. This means that dropping rows with missing values would create a bias in our predictive model which we should avoid. We will impute them with the median of the respective column.

A short check of the labels in the file *train_labels.csv* (count aggregation by value) shows that the label data is complete (no missing values) and there are no erroneous rows with values that are not 0 or 1.

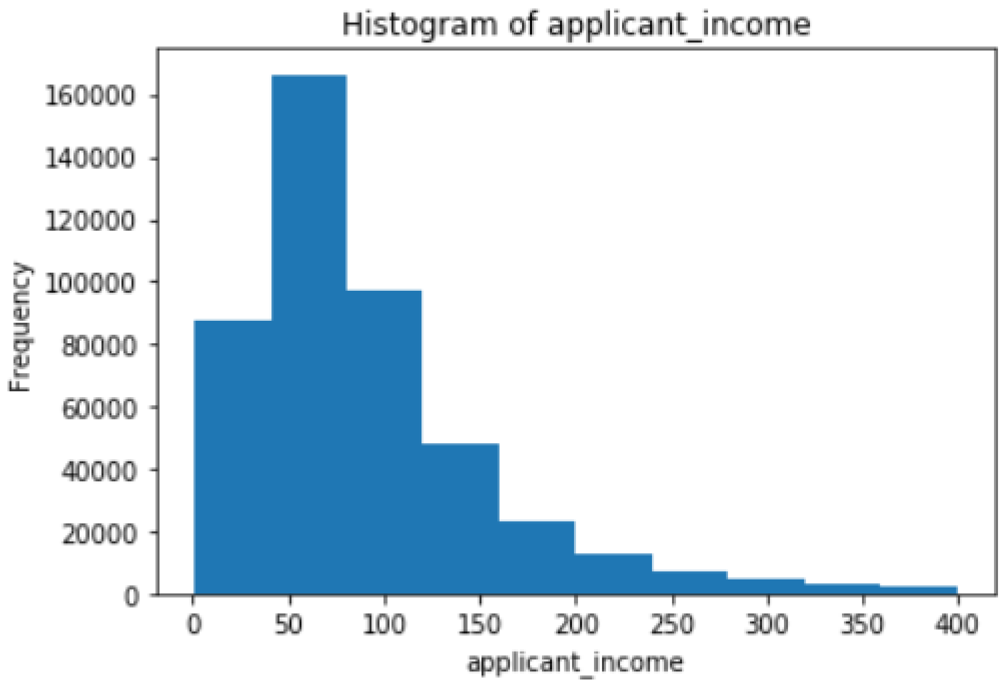
A Histogram plot of the feature *loan_amount* affirms the expectation, that there are outliers with very large values in this column present in the data.



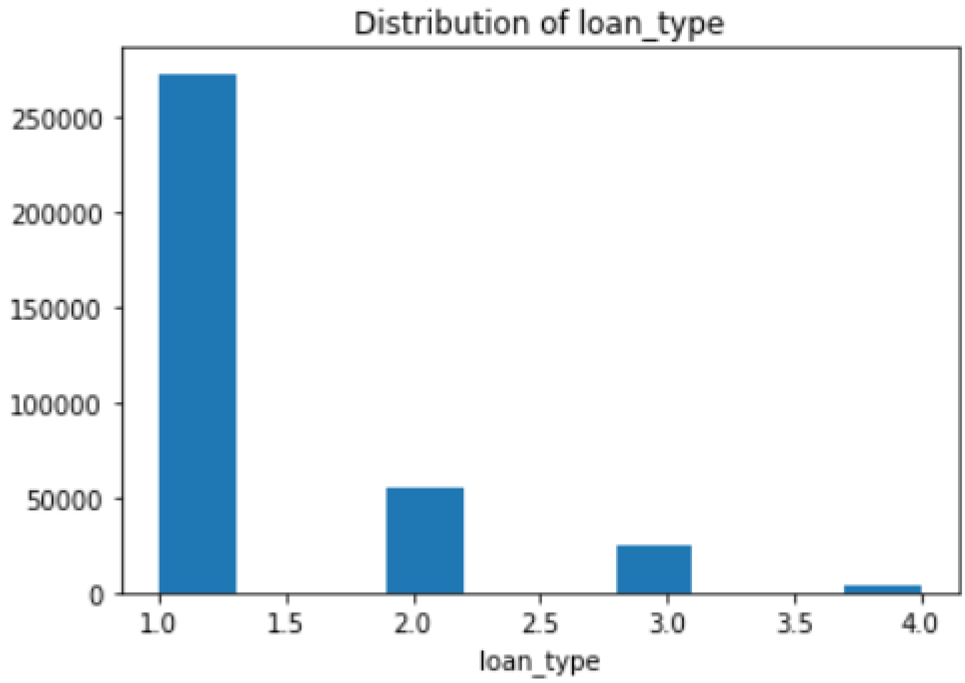
Removing all rows with values greater than 1000 shows a clearer picture. Even without the outliers there is a strong positive skew in this feature. This means that small loans are requested way more often than big ones.



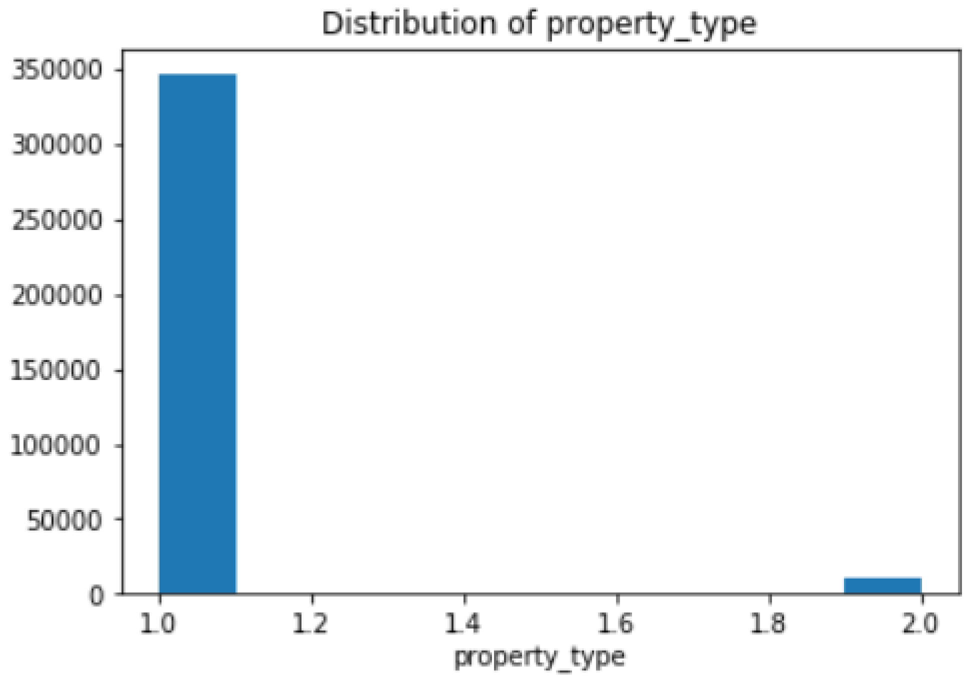
There are probably also outliers with high values in the column *applicant_income*. The histogram looks very much like the initial histogram of *loan_amount*. We remove all values above 400 and see that the distribution of the remaining data is positively skewed as well. More applications are filed by people with smaller incomes.



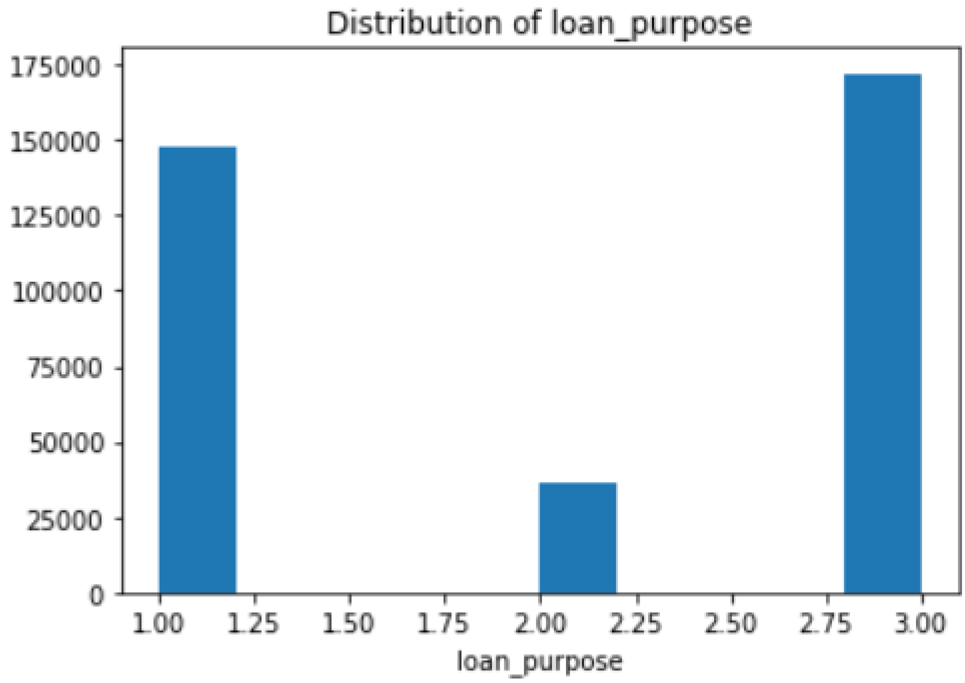
For the *loan_type*, the distribution is also positively skewed. However this must not be directly interpreted, as the feature is actually categorial and the value being smaller bears no meaning as the numbers assigned to the categories are arbitrary. Type 1 are conventional loans, as one would suspect, the majority of loans are conventional ones.



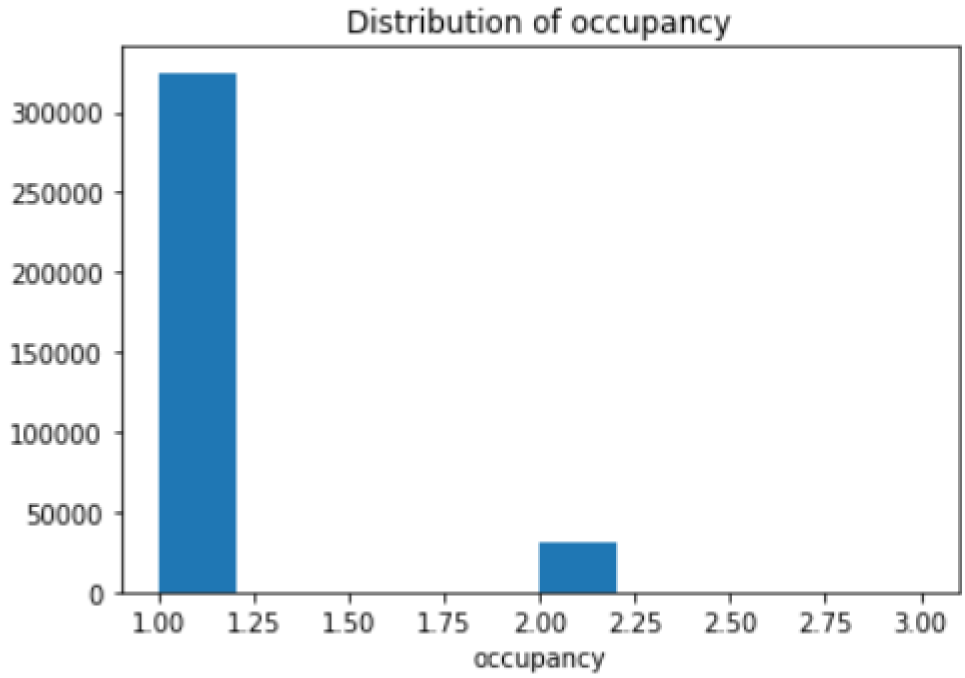
The property type is strongly imbalanced. By far the most common one is 1 (one to four-family) and there are some of type 2 (manufactured housing). Type 3 (multifamily) is not present.



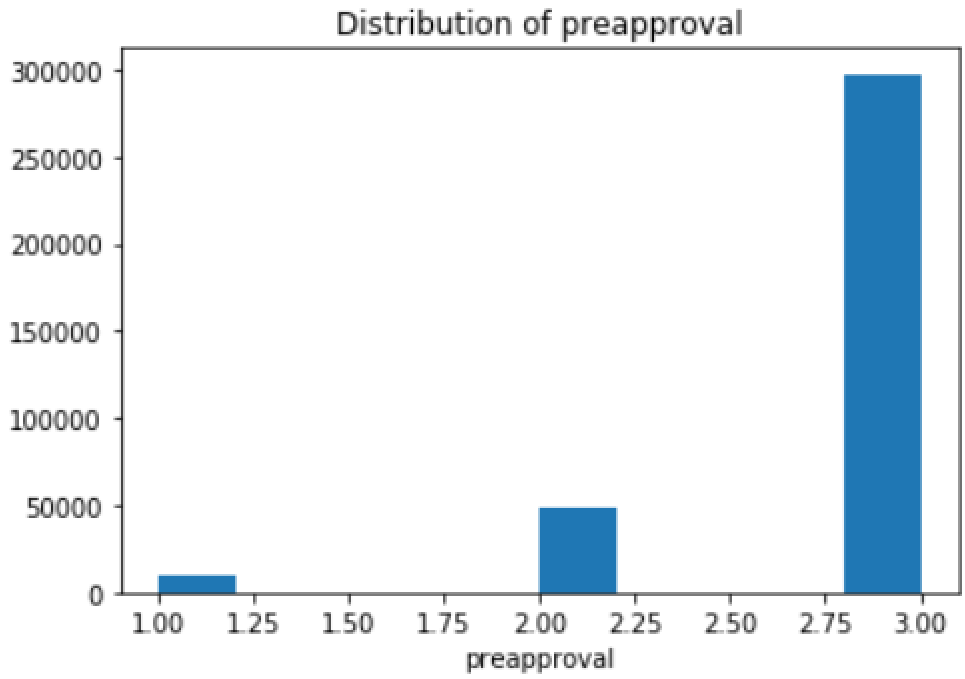
Concerning the purpose of the loan, the main purpose is 3 (refinancing), followed by 1 (home purchase). There are little of type 2 (home improvement).



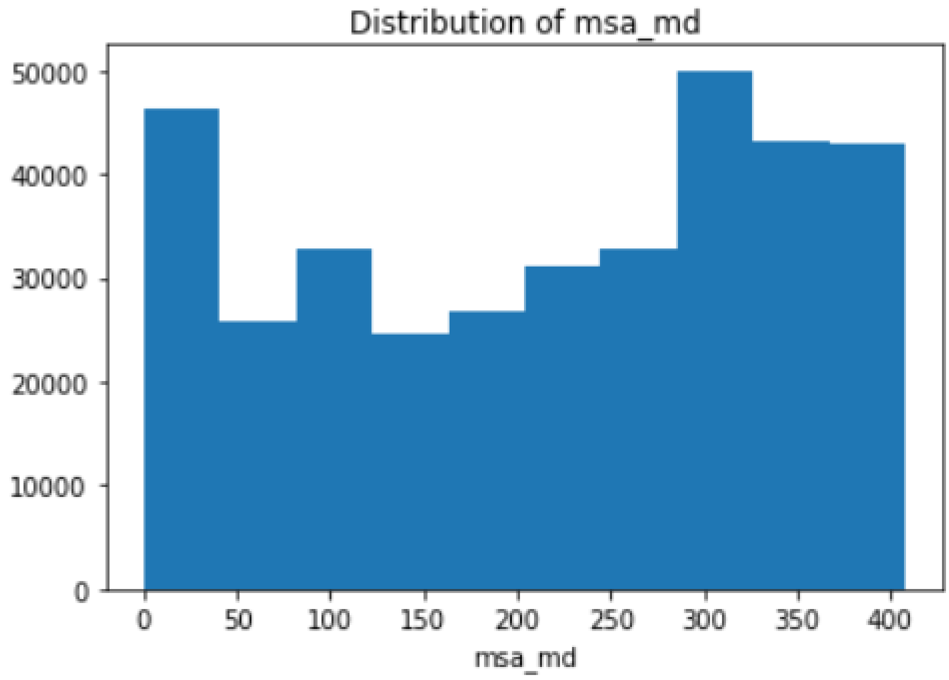
The occupancy is also strongly imbalanced. Most requests are for 1 (Owner-occupied), some but very little are 2 (not owner-occupied).

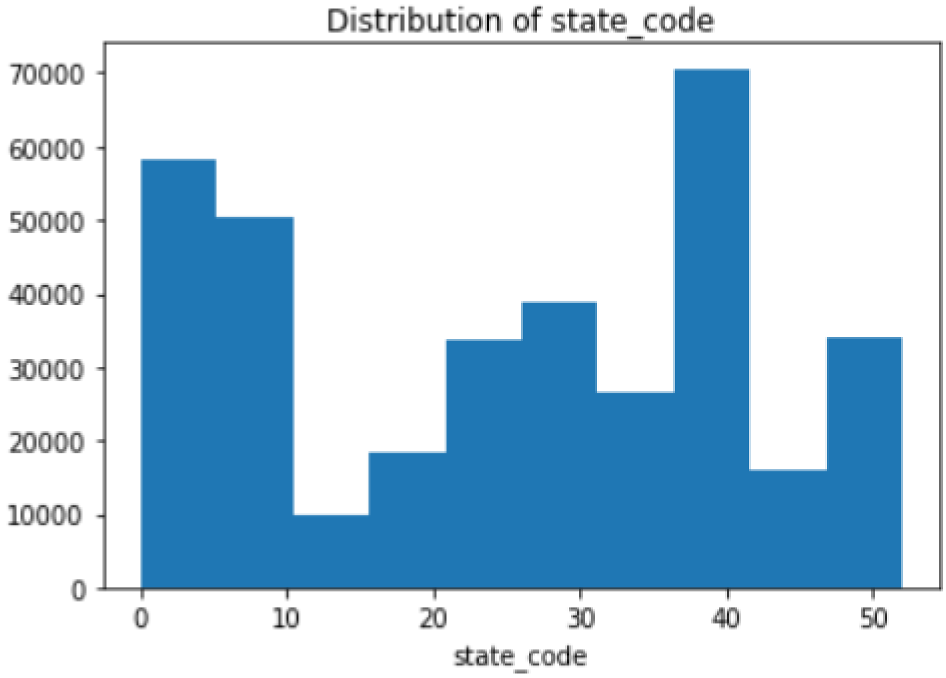


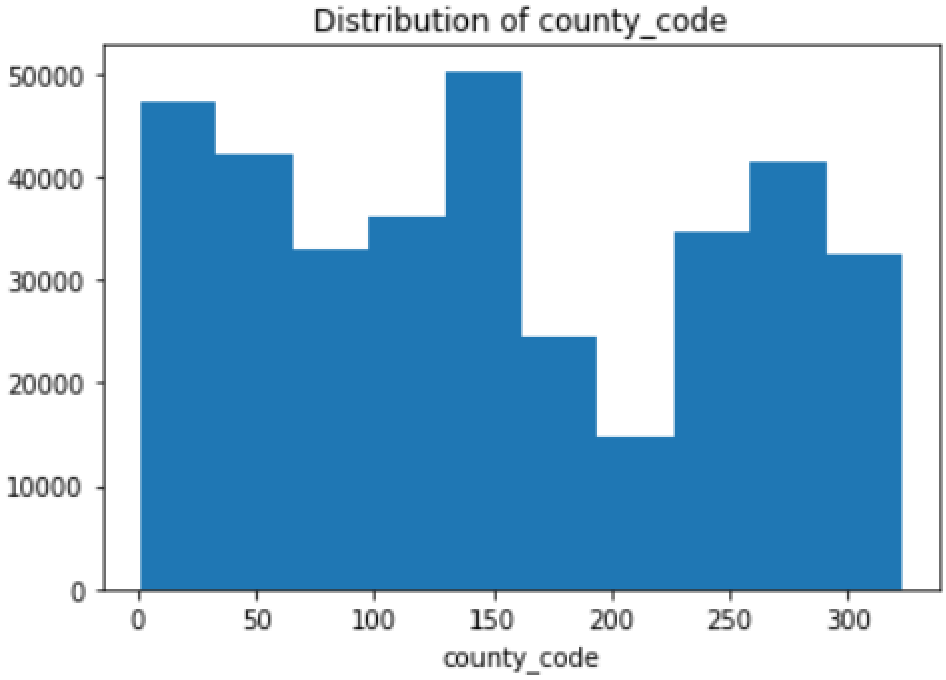
The preapproval is mostly 3 (not applicable).



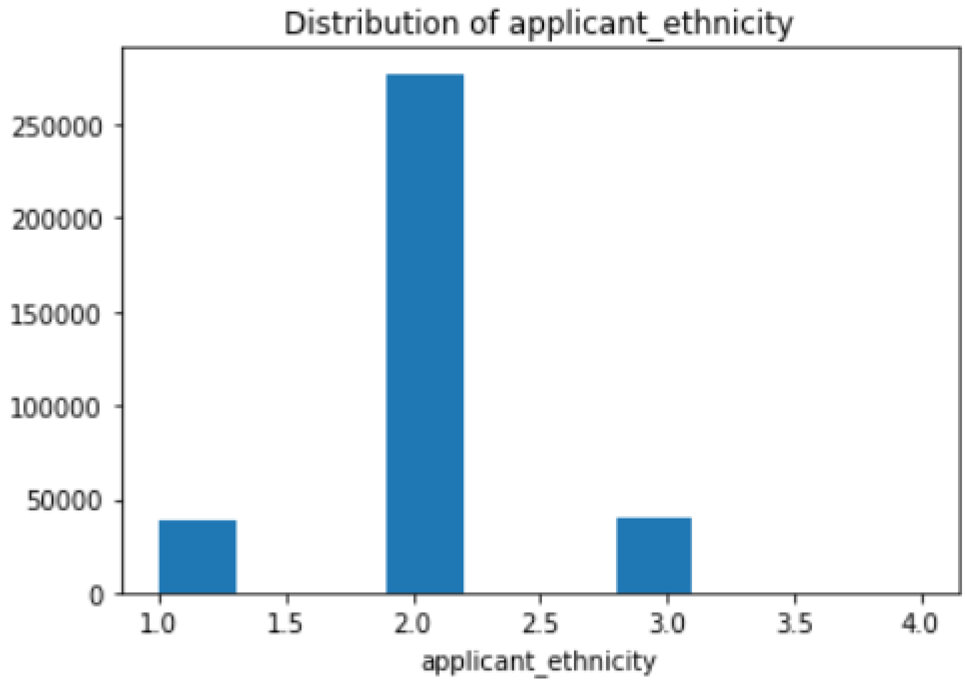
As *msa_md* is a categorical column with no inherent meaning behind the order of values, the order of frequencies cannot really be interpreted. There are no noticeable patterns in the distribution. The same applies to the other geographic indicators *state_code* and *county_code*.



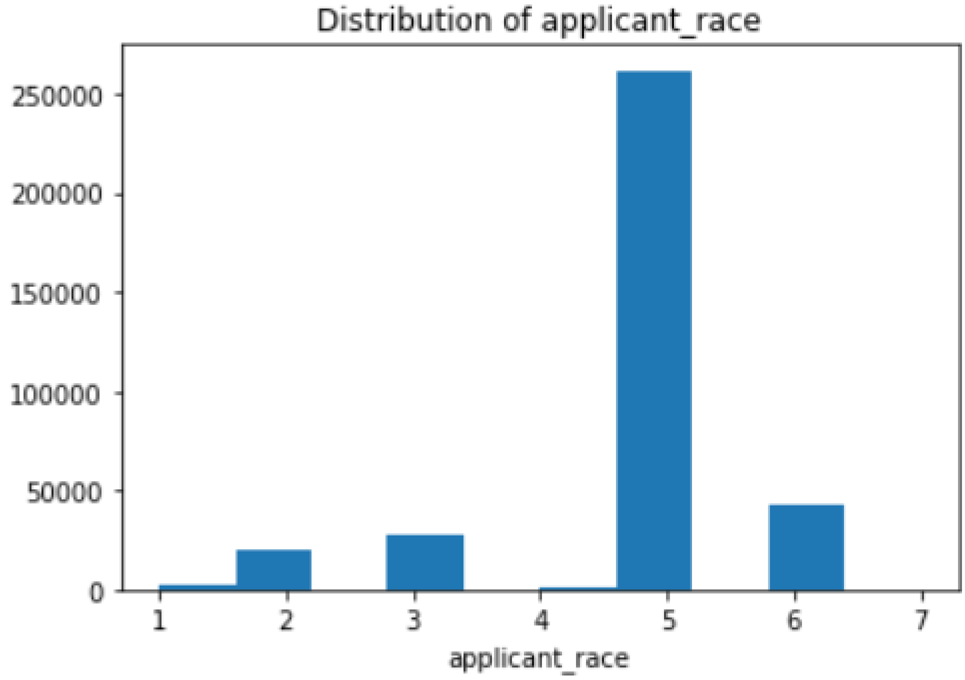




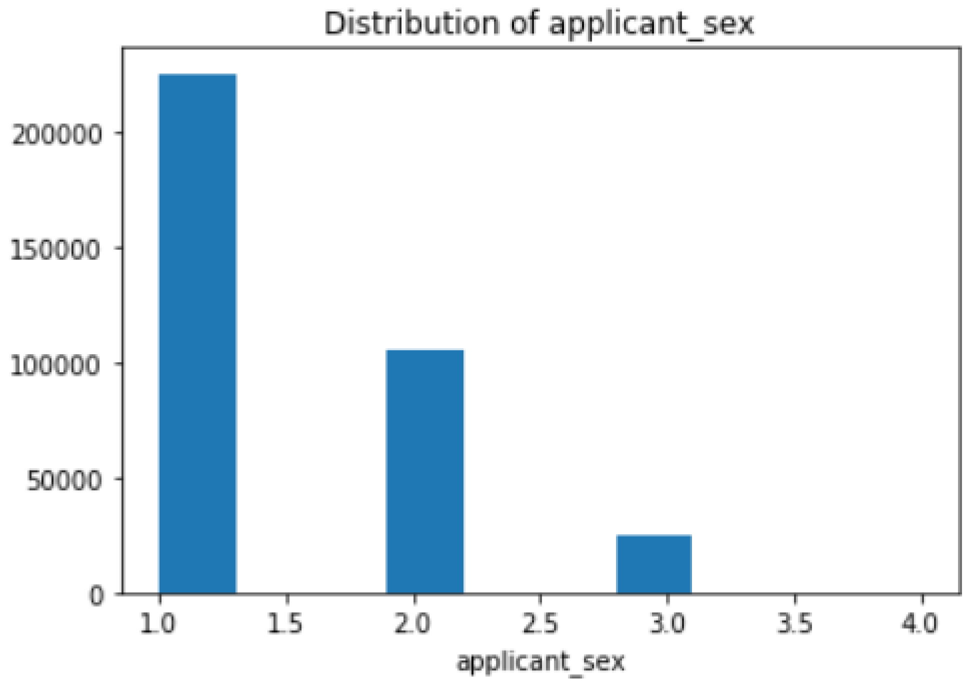
Most applicants belong to the group 2 (Not Hispanic or Latino). There are some of group 1 (Hispanic or Latino) and 3 (Not provided). Very few have group 4 (not applicable) and there are none of group 5 (No co-applicant).



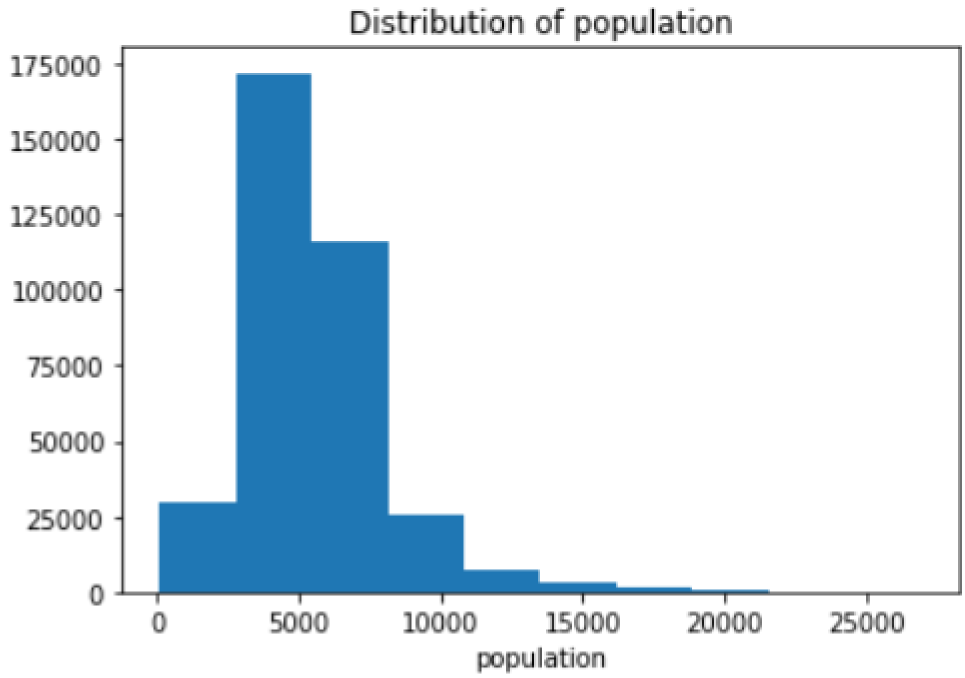
The majority of requests were filed by white people (group 5), followed by Not-Provided (group 6), Black (3) and Asian (2). There are practically no requests by American Indian (1) or with value not applicable (7). The value 8 (no co-applicant) is not present in the data.



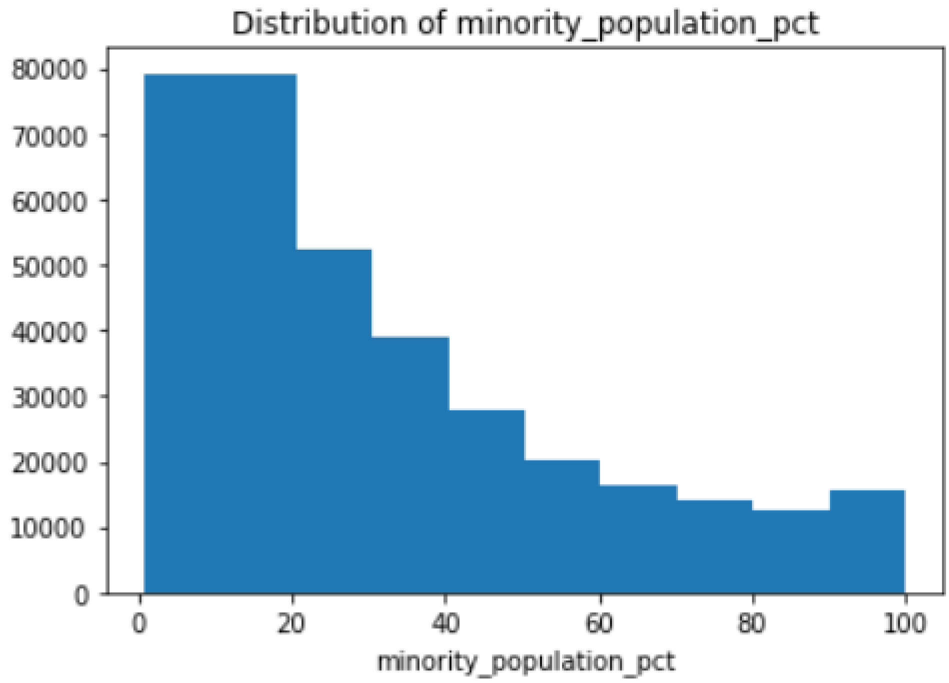
Most requests are made by males (1), about half as many by females (2) and some few are not provided or not applicable.



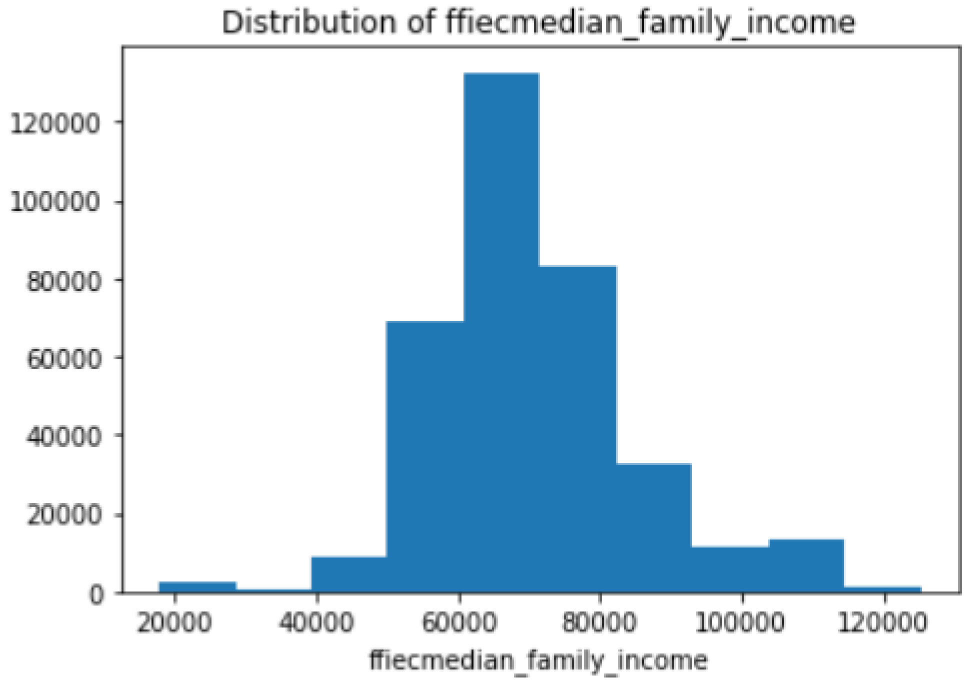
The population distribution is positively skewed. The right tail is very flat.



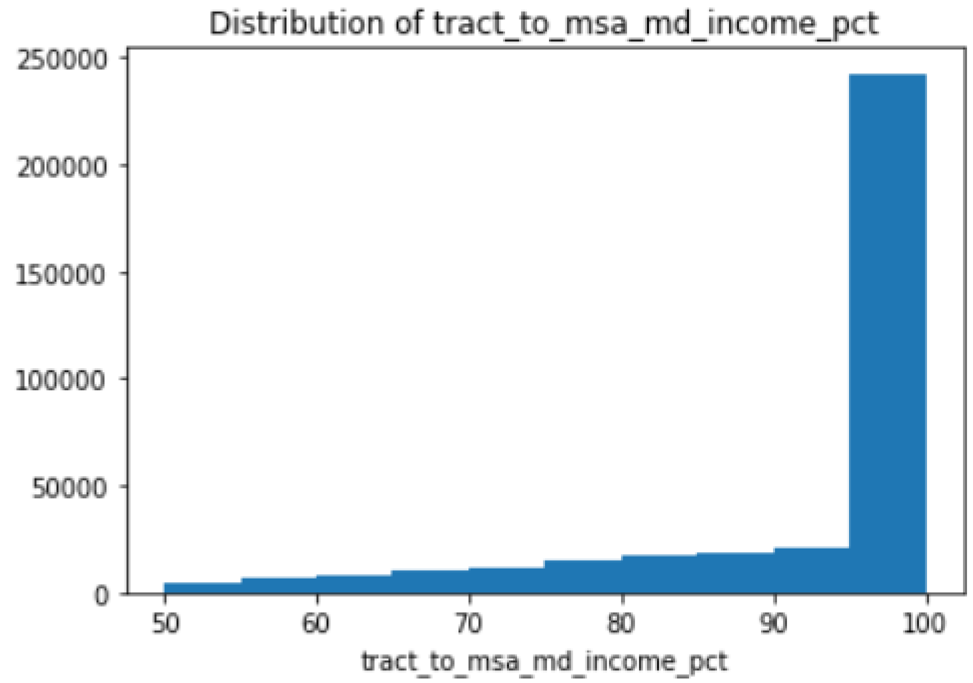
The distribution of the share of minorities among the population is also positively skewed.



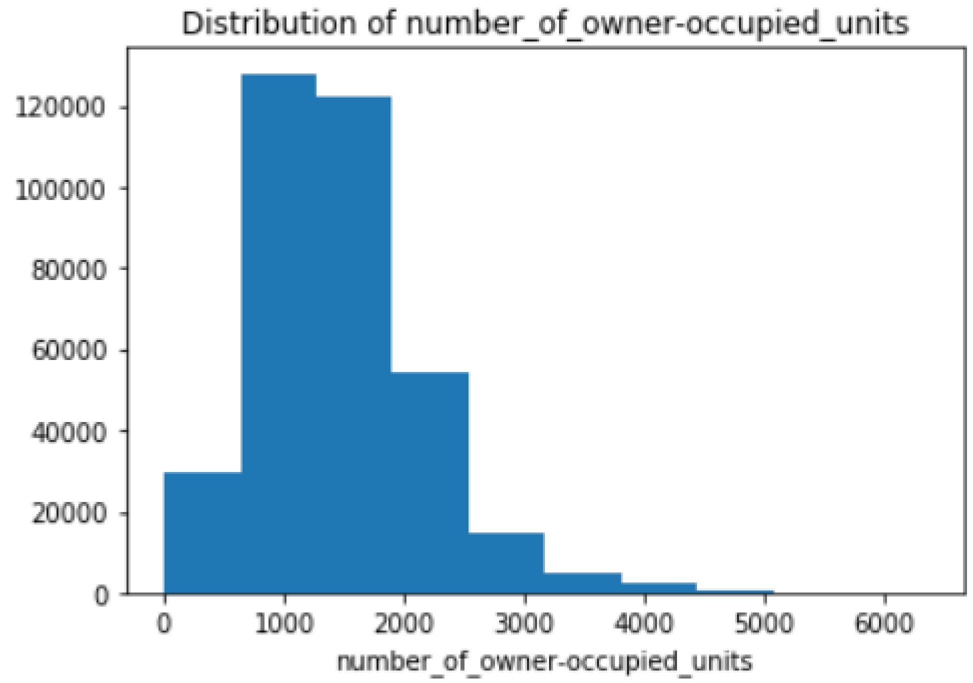
The median family income distribution looks reasonably close to a normal distribution, however it deviates from the normal distribution by having a heavier right tail.



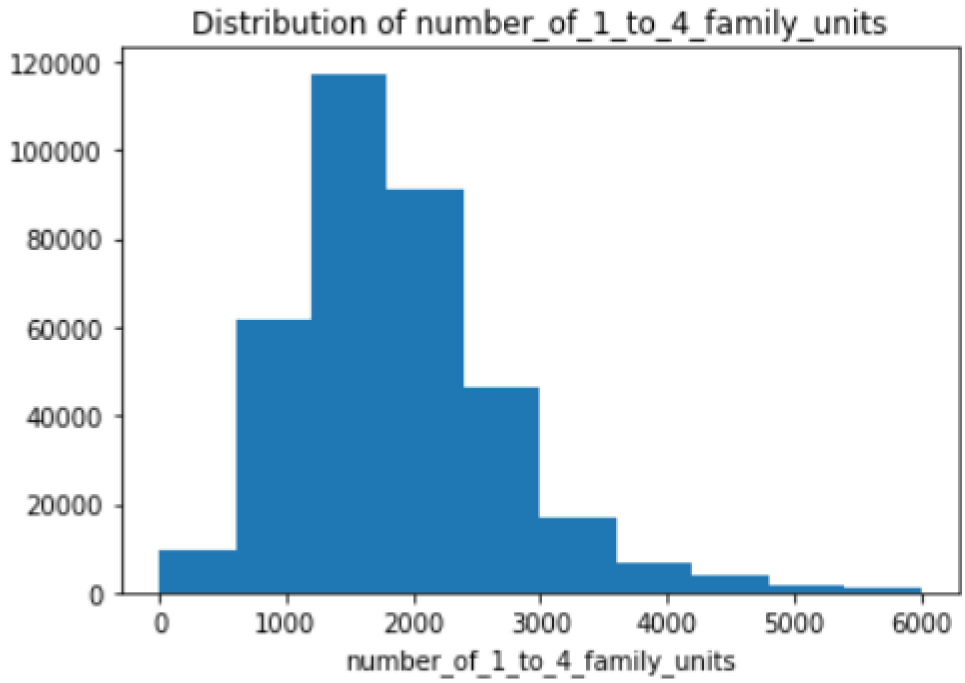
In the plot of the distribution of the percentage of tract median income to MSA/MD median family income we see that for almost all rows the percentage is 100% or close to it. The frequency increases slightly from 50% to 90%, but the values are overall very low compared to the 100% frequency.



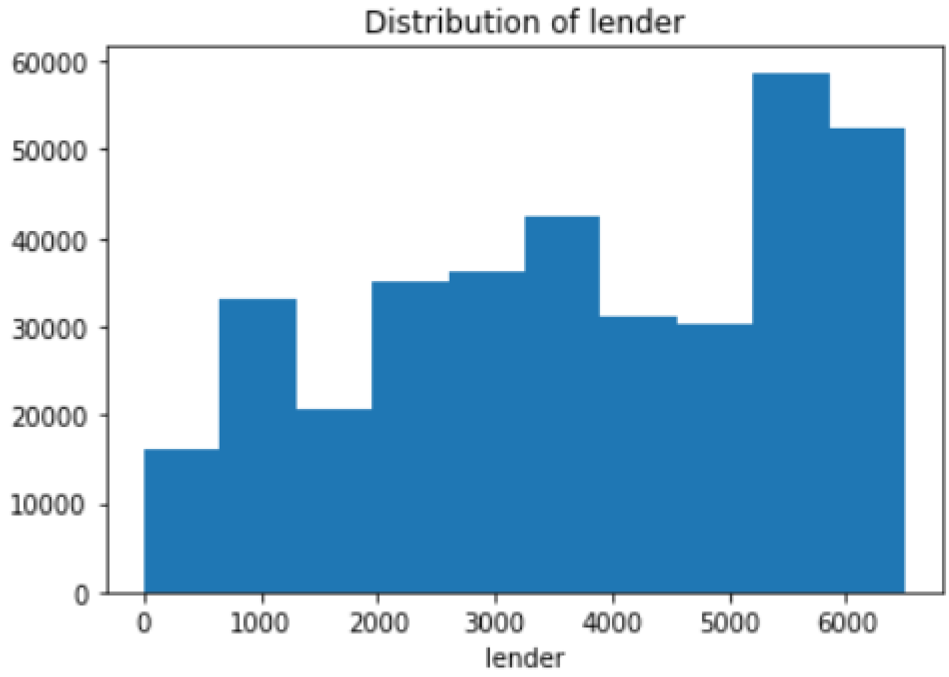
The number of owner-occupied units is centered around a value close to 1500 with a slight positive skew.



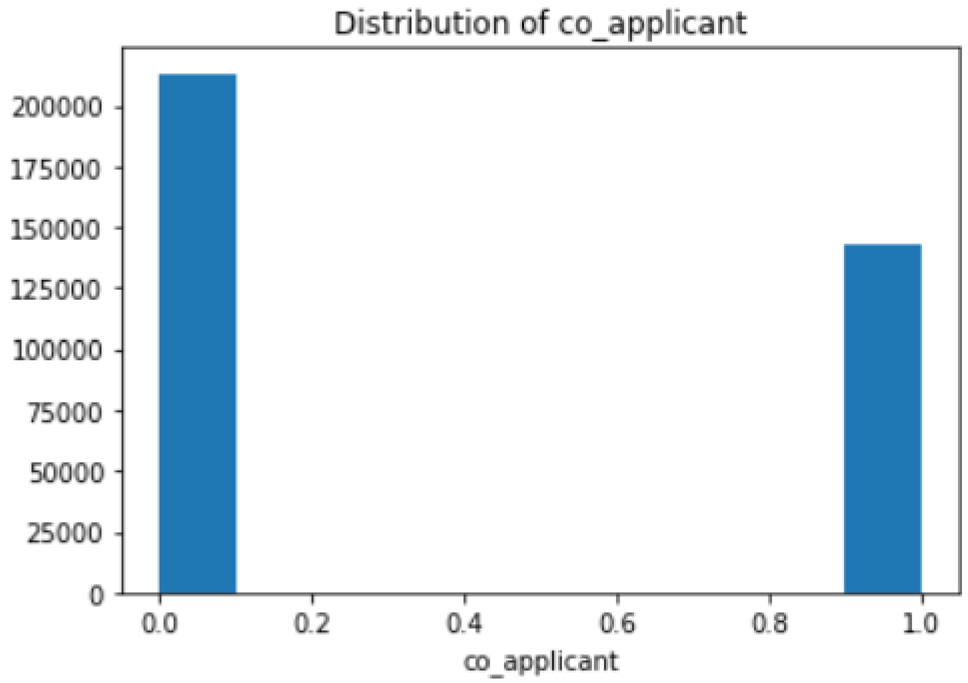
The shape of the number of 1 to 4 family units has a similar shape to that of the number of owner-occupied units. It has a bigger standard deviation.



Similar to the geographical features, the distribution of lenders cannot really be interpreted. The frequencies vary from about 15000 to 60000.



There are more requests without a co-applicant.



Relations between the features

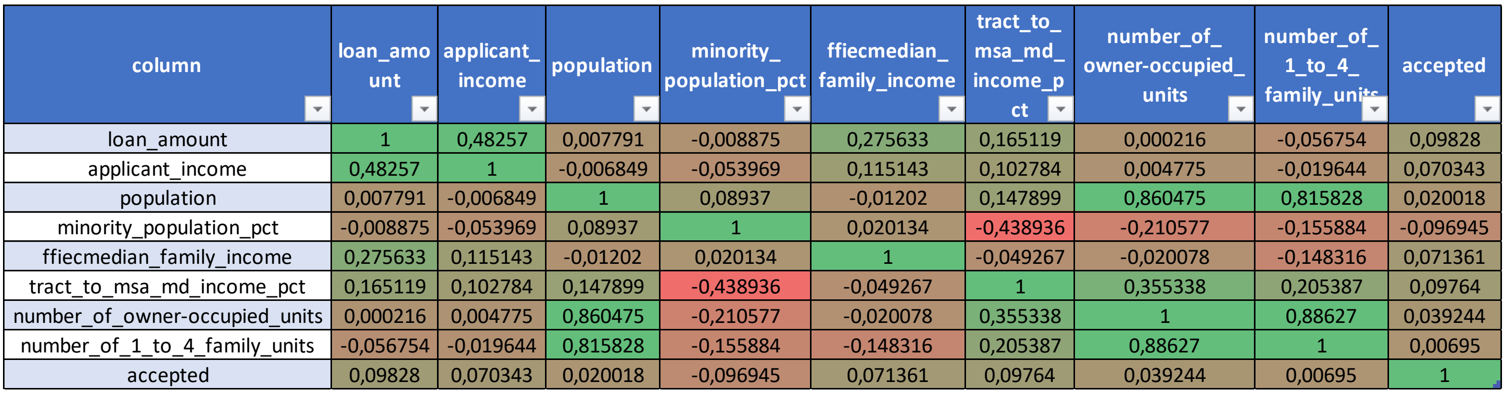
We first calculate the Pearson product-moment correlation coefficients for all columns. This coefficient measures linear correlation between two vectors. A value of +1 means perfect positive linear correlation, a value of -1 perfect negative linear correlation. In the table below values are marked green if they are closer to 1 and red if they are closer to -1. We can identify some correlations among the columns from this table:

Positive:

- *applicant_income* and *loan_amount* (0.48)
- *ffiecmedian_family_income* and *loan_amount* (0.28)
- ffiecmedian_family_income and msa_md (0.27)
- *number_of_owner-occupied_units* and *population* (0.86)
- *number_of_owner-occupied_units* and *tract_to_msa_md_income_pct* (0.36)
- *number_of_1_to_4_family_units* and *population* (0.81)
- *number_of_1_to_4_family_units* and *number_of_owner-occupied_units* (0.89)

Negative:

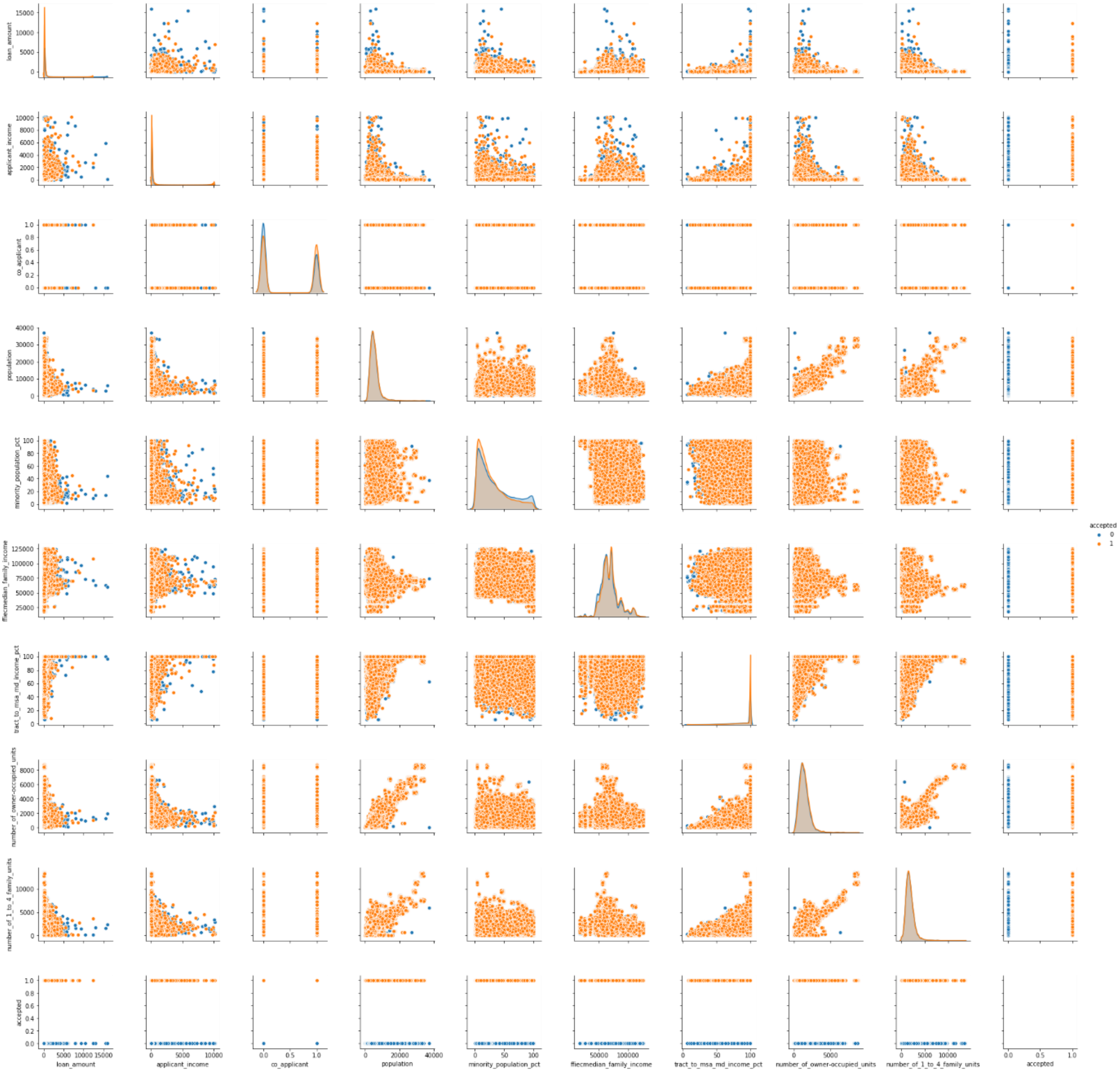
- *tract_to_msa_md_income_pct* and *minority_population_pct* (0.43)
- *number_of_owner_occupied_units* and *minority_population_pct* (0.21)



column	loan_amount	applicant_income	population	minority_population_pct	ffiecmedian_family_income	tract_to_msa_md_income_pct	number_of_owner-occupied_units	number_of_1_to_4_family_units	accepted
loan_amount	1	0,48257	0,007791	-0,008875	0,275633	0,165119	0,000216	-0,056754	0,09828
applicant_income	0,48257	1	-0,006849	-0,053969	0,115143	0,102784	0,004775	-0,019644	0,070343
population	0,007791	-0,006849	1	0,08937	-0,01202	0,147899	0,860475	0,815828	0,020018
minority_population_pct	-0,008875	-0,053969	0,08937	1	0,020134	-0,438936	-0,210577	-0,155884	-0,096945
ffiecmedian_family_income	0,275633	0,115143	-0,01202	0,020134	1	-0,049267	-0,020078	-0,148316	0,071361
tract_to_msa_md_income_pct	0,165119	0,102784	0,147899	-0,438936	-0,049267	1	0,355338	0,205387	0,09764
number_of_owner-occupied_units	0,000216	0,004775	0,860475	-0,210577	-0,020078	0,355338	1	0,88627	0,039244
number_of_1_to_4_family_units	-0,056754	-0,019644	0,815828	-0,155884	-0,148316	0,205387	0,88627	1	0,00695
accepted	0,09828	0,070343	0,020018	-0,096945	0,071361	0,09764	0,039244	0,00695	1

These correlations will not impact the performance of the predictive algorithm very much, because a boosted decision tree model will be used, which is quite robust to correlations in the features. For alternative algorithms like logistic regression, additional cleaning steps to eliminate correlations would be necessary.

To further investigate the interaction of the numerical features we use a pair plot matrix. In the diagonal we see a kernel density estimation of the distribution of the features. In cell (*i, j*) we see a scatter plot with feature *i* on the x-axis and feature *j* on the y-axis. Refused requests are marked with a blue dot, accepted requests are in yellow. We can see the linear correlations that the correlation coefficient indicates in these plots.

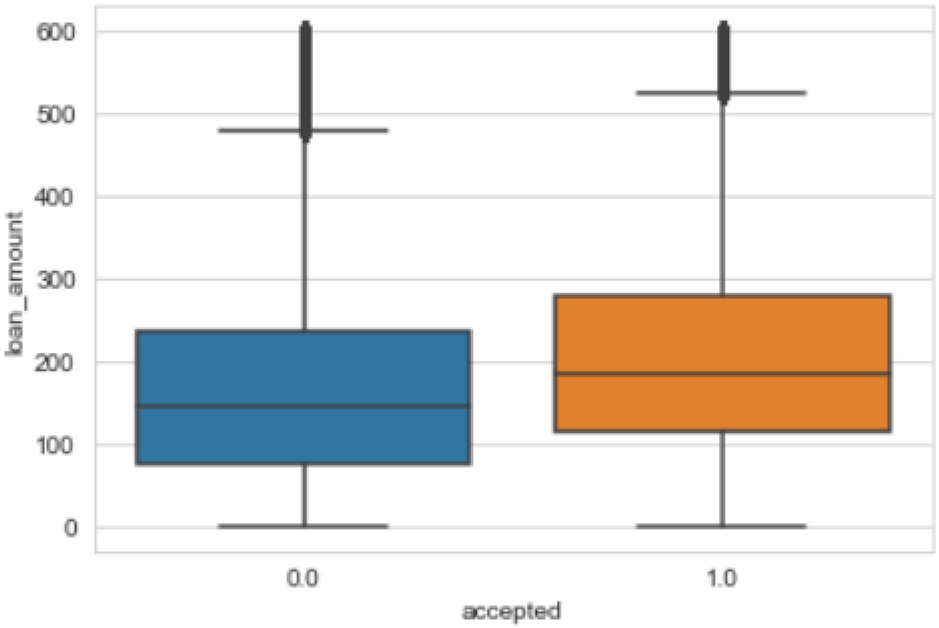


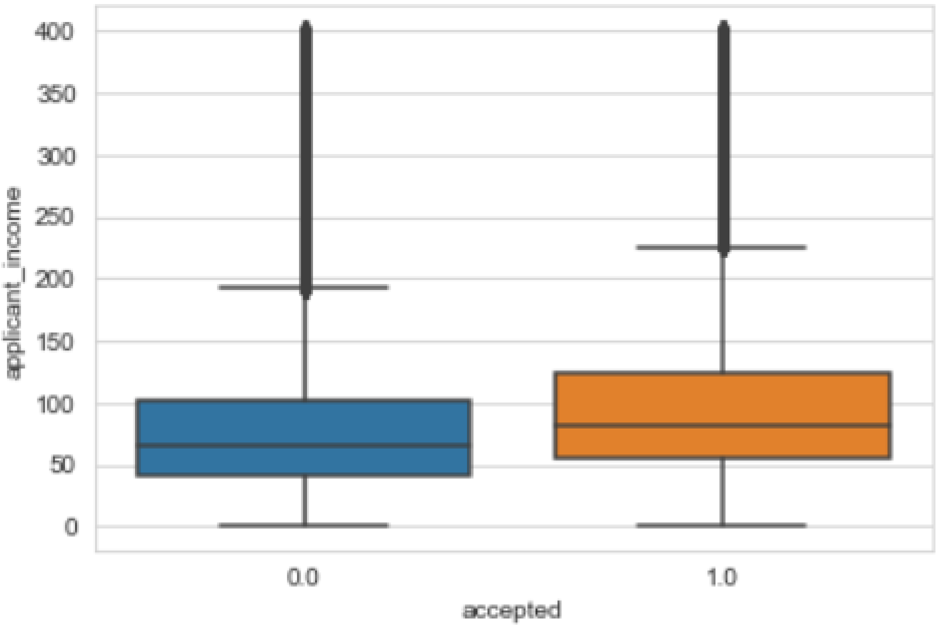
Some of these correlations seem quite likely. For example, if the total population is higher, then there are higher numbers of units, so there are probably also a higher number of owner-occupied and 1-to-4-family units.

Relation with the label

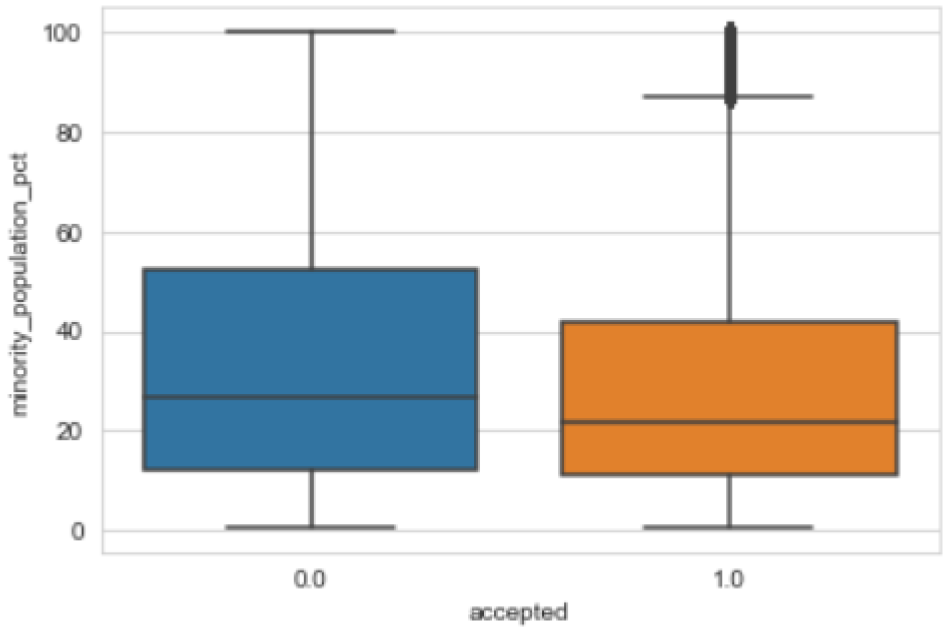
We now investigate how the features are correlated with the label. In this case correlations can hint to a causal relationship, that could be built into our predictive model.

We see that the *loan_amount* for accepted rows is higher than for not accepted. The same is visible for *applicant_income*.

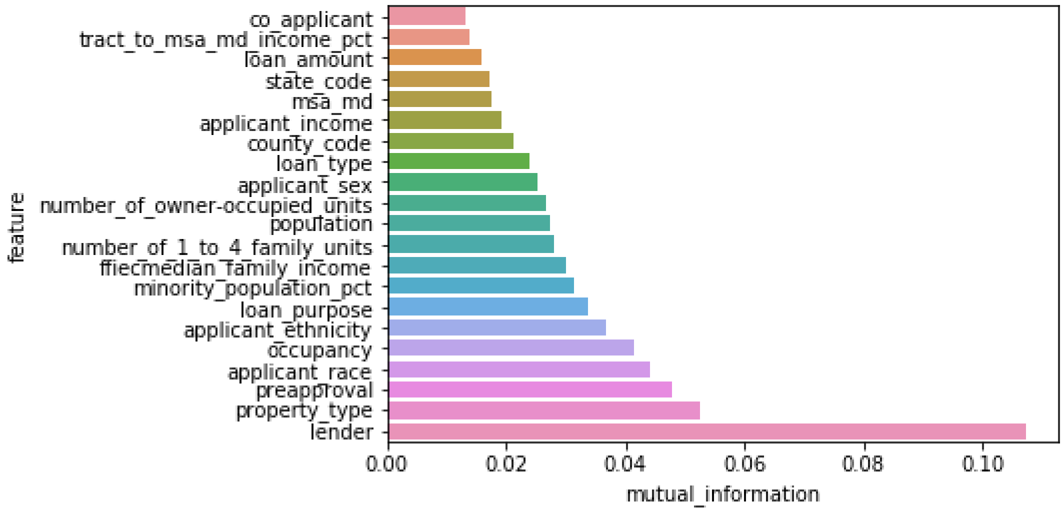




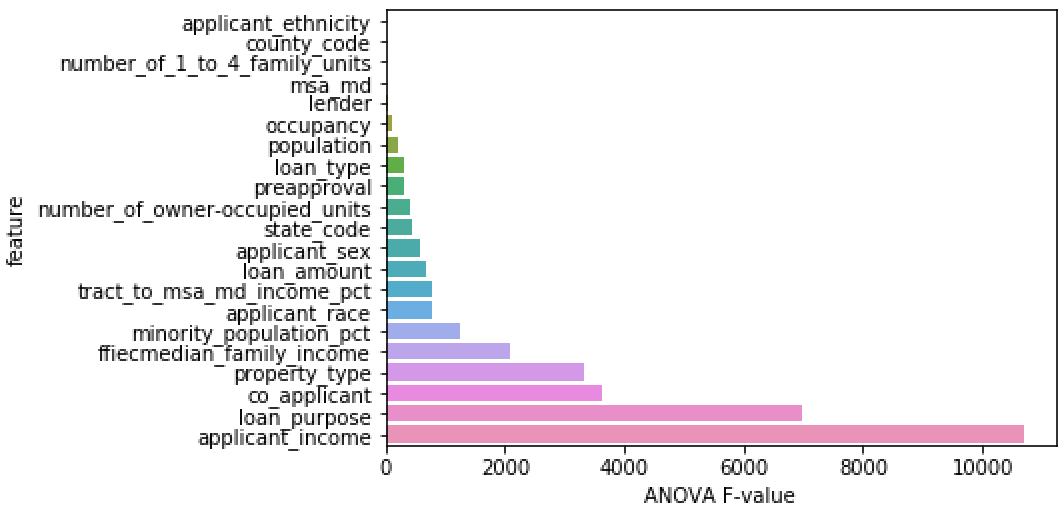
Accepted rows have a slightly lower minority population percentage as accepted ones.



For each feature we calculate the mutual information [8] with the label. We see that the features *lender*, *property_type*, preapproval, *applicant_race* and occupancy score the highest. If regarded in isolation, these features have a strong relation to the label. However, this measure does not account for interactions between the features.



Using the ANOVA F-value [9] as a measure for feature importance, we get a different picture. Here, the applicant income and *loan_purpose* are much more important. In both cases the geographical features like *county_code*, *state_code* or population receive a low value in comparison to the loan- and applicant-specific features.



Data preparation for machine learning

To enable the machine learning algorithms to perform better on the data, some preparation steps are taken:

Feature engineering

As not only the total amounts of the loan and the applicant income are important for the acceptance of the loan from a business perspective, but also the relative sizes, a new feature *loan_income_ratio* is created as the quotient of *loan_amount* and *applicant_income*.

Imputation of missing values

All missing values (completely missing, as well as encoded by -1) are replaced with the respective column median. As described above, removing the rows would introduce a bias into the training data distribution and thus into the prediction.

Binning

The categorical features *lender*, *county_code*, *state_code* and *msa_md* have a high number of distinct values. The rows were binned in the following way:

For each of these features:

- For each distinct value of the feature the quota of accepted rows to not accepted rows was calculated
- The interval [0,1] was divided into bins. The number of bins was chosen to be approximately the square root of the number of distinct values of the feature
- For each row, the original feature was removed and the number of the bin in which the value of the original feature fell, was added as a new feature
- The size of the bins is chosen, so that in each bin contains approximately the same number of rows

In this way, new features are created that have a stronger relation with the label were created from the old ones. Note that, while the ordering of the original values does not bear any inherent interpretation, the numbering of the bins (which are the values of the new features) is monotonously increasing with the percentage of acceptance. Still no interpretable metric is induced.

Encoding

Categorical features with less than 5 distinct values were one-hot encoded. The others were left in label encoding. With too many distinct values, the data matrix becomes too sparse and algorithms will perform worse (*curse of dimensionality*, cf. [10], p.22-26).

Normalization

All numerical features (not the label-encoded) were normalized (Z-score).

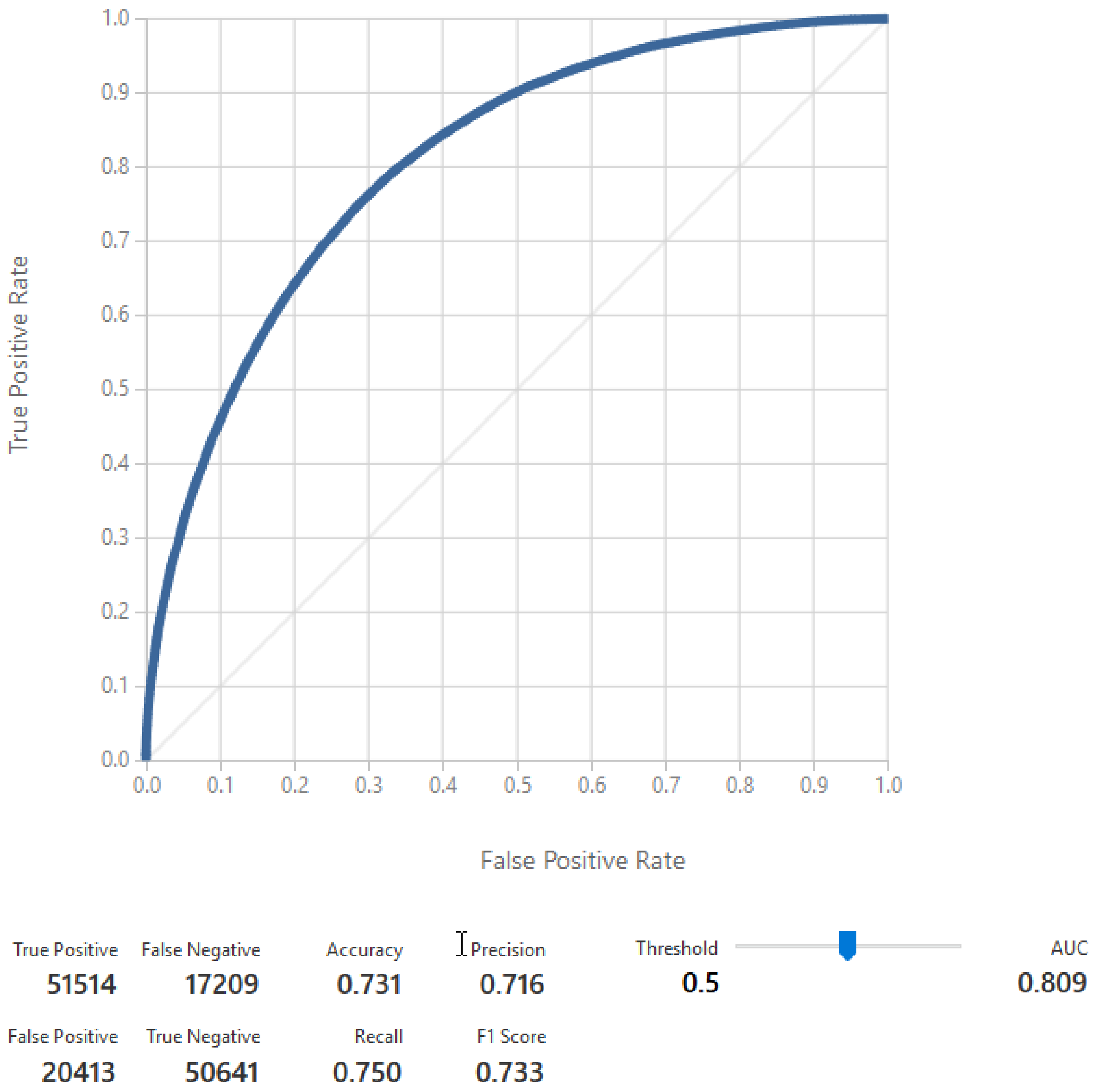
Model building

A model was built to predict the approval from the information in the other features. For this, the data was transformed using the preparation steps described above. To thirds of the data were used for training and validation (using 10-fold cross validation), the remaining third was used for final evaluation and estimation of the generalization error.

A model of boosted decision trees on the whole dataset could not strongly classify rows for which preapproval is not applicable (preapproval = 3). The scores the model gave for these rows was often very close to 0.5. Because of this, a distinct model decision tree-based model was built for these rows.

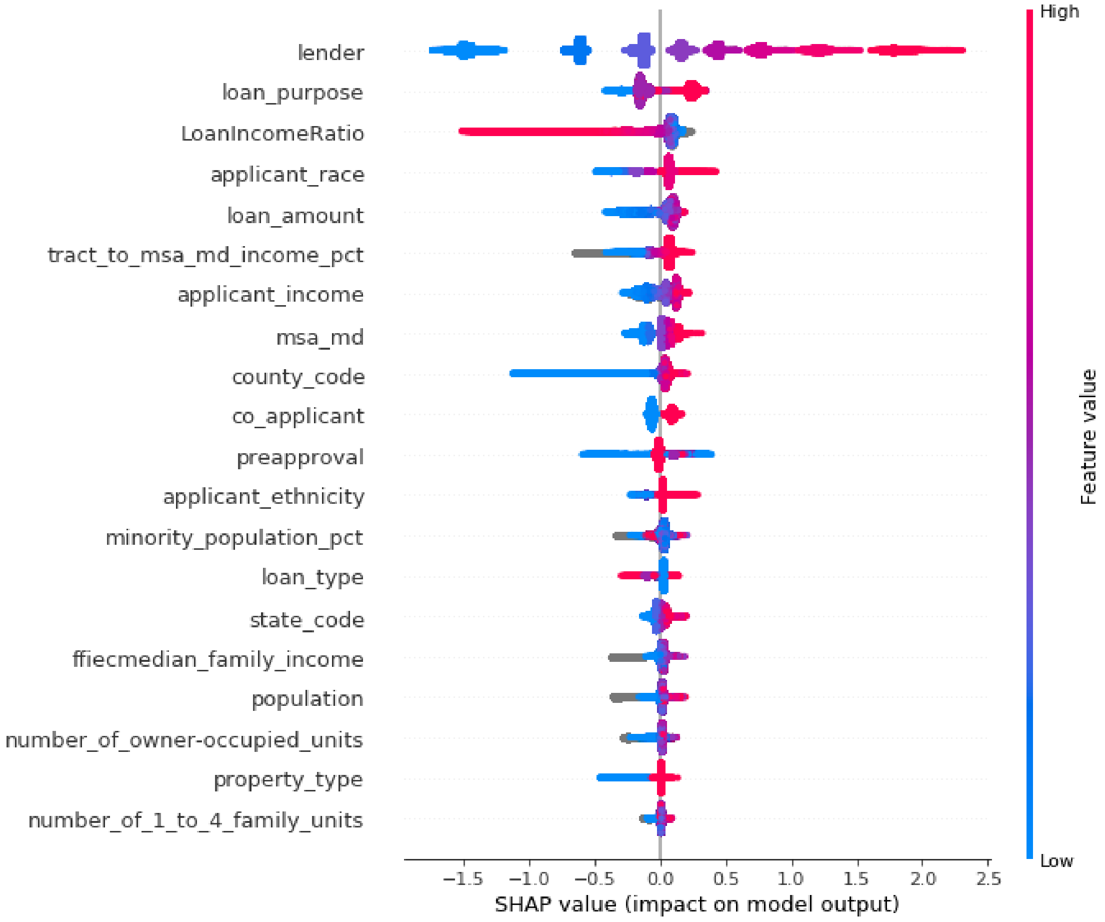
Model evaluation

The resulting model, consisting of two boosted decision tree models, reaches about 73% accuracy on the test set. The model seems to have a slight bias towards the positive class, because the number of false positives is approximately 18% higher than the number of false negatives, even though the number of positive and negative labels is uniformly distributed (at least in the dataset provided).



True Positive	False Negative	Accuracy	Precision	Threshold	AUC
51514	17209	0.731	0.716	0.5	0.809
False Positive	True Negative	Recall	F1 Score		
20413	50641	0.750	0.733		

Using SHAP [11] we can extract the information, which features the decision tree ensemble uses for classification. We see that here *lender*, *loan_purpose* and *loan_income_ratio* and *applicant_race* have the highest importance to the algorithm. However, in comparison to the ANOVA F-scores and the mutual information measures, especially *msa_md* and *county_code* have a higher impact on the model output, than the ANOVA and mutual information scores imply.



Conclusion

This analysis shows, that the acceptance of a loan can be predicted moderately well from information on the loan and the applicant. An accuracy of close to 74% can be useful for practical purposes to support decision making in the lending business, but the model is not strong enough to be relied on completely. However, it could be added to support the manual decision process made by business experts.

Especially useful for the prediction are information on

- The lender
- The purpose of the loan
- The absolute and relative values of the loan amount and the applicant income
- The applicant's race and ethnicity (note that ethical considerations must be made when using race and ethnic information in a predictive model. To prevent a racist bias these features can be excluded or its importance in the model can be scaled down by giving it a smaller weight.)

Geographic and census data are not as important but can also provide a notable amount of information to the prediction.

References

[1] "https://pandas.pydata.org/," April 2019. [Online].

[2] "https://www.numpy.org/," April 2019. [Online].

[3] "https://seaborn.pydata.org/," April 2019. [Online].

[4] "https://scikit-learn.org," April 2019. [Online].

[5] "https://www.cs.waikato.ac.nz/~ml/weka/," April 2109. [Online].

[6] "https://studio.azureml.net/," April 2019. [Online].

[7] FFIEC, "https://www.ffiec.gov/hmdarawdata/FORMATS/2015HMDACodeSheet.pdf," 2015. [Online].

[8] "http://www.info612.ece.mcgill.ca/lecture_02.pdf," [Online].

[9] "https://www.sciencedirect.com/topics/medicine-and-dentistry/analysis-of-variance," 2019. [Online].

[10] T. Hastie, R. Tibshirani and J. Friedman, The Elements of Statistical Learning, Springer Series in Statistics, 2009.

[11] "https://github.com/slundberg/shap," 2019. [Online].

[12] M. J. Azur, M. A. Stuart, C. Frangakis and P. J. Leaf, "Multiple Imputation by Chained Equations: What is it and how does it work?," 1 March 2011. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3074241/pdf/nihms267760.pdf. [Accessed April 2019].