Q2 Project Report

Chan, Jerry A15872251

Pochiraju, Apoorv A15494574

Wang, Zhendong A15605373

Zhang, Yujie A15620406

March 9, 2022

1. Introduction

Nowadays, the algorithmic decision-making system has been very common in people's daily lives. Gradually, some algorithms become too complex for humans to interpret, such as some black-box machine learning models and deep neural networks. In order to assess the fairness of the models and make them better tools for different parties, we need explainable AI (XAI) to uncover the reasoning behind the predictions made by those black-box models. In our project, we will be focusing on using different techniques from causal inferences and explainable AI to interpret various classification models across various domains. In particular, we are interested in three domains - healthcare, finance, and the housing market. Within each domain, we are going to train four binary classification models first, and we have four goals in general: 1) Explaining black-box models both globally and locally with various XAI methods. 2) Assessing the fairness of each learning algorithm with regard to different sensitive attributes; 3) Generating recourse for individuals - a set of minimal actions to change the prediction of those black-box models. 4) Explaining False Negative and False Positive predictions using Causal Inference.

2. Datasets

2.1 Description of Datasets

In our project, we will use datasets from three domains: **finance**, the **housing market**, and **healthcare**.
For the **finance** domain, we will use the loan defaulter dataset obtained from Kaggle. The loan defaulter dataset consists of information such as gender and income of 307,511 applicants. For this dataset, we will predict whether an applicant will be a defaulter.
For the **housing market** domain, we will use the Airbnb dataset obtained from Kaggle. The Airbnb dataset consists of basic information such as name and location for 3,818 Airbnb properties. For this dataset, we will predict the class that the price of an Airbnb property falls into.
For the **healthcare** domain, we will use the data on hospital readmission for diabetes patients obtained from Kaggle. The data was collected for the Hospital Readmission Reduction Program operated by Medicare & Medicaid Services and indicates whether diabetic patients were readmitted to the hospital and whether it was within 30 days or after beyond 30 days. This dataset contains records for 101,766 patients and includes attributes for each patient such as race, age, gender, insulin levels, type of admission, and other specific information about medical history. For this dataset, we will predict whether the patient will be readmitted.

2.2 Datasets Preprocessing

2.2.1 Loan Dataset

The original loan dataset has 307,511 rows and 122 columns. We first dropped the 49 columns that have more than 35% of missing values and are not relevant to the classification task. Then we did missing value imputation for categorical features using the mode and for numerical features using the mean value. Then we one-hot encoded 12 categorical features and created column names for the one-hot encoding results by combining the feature name and category it falls into. For example, for the 'CODE_GENDER' feature, the one-hot encoded columns are 'CODE_GENDER: F' (representing female) and 'CODE_GENDER: M' (representing male). Since the Target column is highly imbalanced, we sampled the dataset to make it balanced. The resulting dataset has 49,650 rows and 187 columns, where 186 columns are features, and 1 column is the label column.

2.2.2 Airbnb Dataset

For the Airbnb dataset, we only kept 35 columns that are relevant for predicting the listing price, including the number of bedrooms, amenities, locations, availability, etc. Then, we did some data cleaning, including striping string characters in numerical features, filling some null values with mean, mode, and string “None” accordingly, and dropping null values that are meaningless. The original dataset has 3818 records, and there were 2844 records left after the data cleaning. We further engineered the categorical features using one-hot encoding, and we binarized our label “Price” with the threshold of average price among all listings. Finally, the feature dataset is in the shape of 2844 × 263, and the label dataset is in the shape of 2844 × 1.

2.2.3 Healthcare Dataset

The healthcare dataset was filtered down to 10 relevant columns: race, gender, age, admission type, time in hospital, insulin level, diabetes, whether the patient takes a diabetic medication, and whether the patient was readmitted or not (the target variable). For race and gender, each category within the column was transformed into a new feature. For each category of race (white, black, Hispanic, Asian, Other) and each gender (Male and Female), a new binary feature was created to indicate whether the patient identified with that category. The age column had an age for each patient, so the column was binarized (1 or 0) to indicate whether the patient was older than 50. The column indicating insulin level contained four categories (No, Up, Steady, Down) and was transformed into 4 different binary features. Lastly, the target variable column for readmission was transformed into a binary column which included a 1 if the patient was readmitted before or after 30 days, and a 0 if the patient was never readmitted.

3. Description of Methods

3.1 Machine Learning Models

We trained four black-box classification models: **XGBoost**, which implements machine learning algorithms under the Gradient Boosting framework; **LightGBM**, which is a fast and high-performance gradient boosting framework based on decision tree algorithms; **TabNet**, which is a Deep Neural Network for tabular data; **SVM**, which does classification by constructing a hyperplane or set of hyperplanes in a high-dimensional space. All four models are capable of doing binary classification tasks and their parameters don’t have any physical significance in terms of equivalence to process parameters such as heat or mass transfer coefficients.

3.2 Model Explanations

We used both global model-agnostic and local model-agnostic methods to generate explanations for the four binary classification models.

3.2.1 Partial Dependence Plot (PDP)

The Partial Dependence Plot (PDP) is a global model-agnostic method that works by marginalizing the machine learning model output over the distribution of the features in set C so that the function shows the relationship between the features in set S we are interested in and the predicted outcome. In the case that features are uncorrelated, a PDP shows how the average prediction in the dataset changes when a feature changes.

3.2.2 Permutation Feature Importance

The Permutation Feature Importance is a global model-agnostic method that calculates the feature importance by permuting the feature in the dataset. A feature is important if the error of a model increases significantly after permuting the feature. A feature is not important if the error of a model does not change after shuffling the feature values. The Permutation Feature Importance method takes into account all interactions with other features by destroying the interaction effects with other features when permuting the feature.

3.2.3 Local Interpretable Model-agnostic Explanations (LIME)

As a local model-agnostic method, LIME tests what happens to the model output when we give variations of the model input data. LIME trains an interpretable model such as a decision tree on the perturbed data, which is a good approximation of the black box model predictions locally. Since LIME has the advantage of generating local explanations for different black-box models, we can use it as a good model explanation method to generate explanations for single instances.

3.2.4 Shapley Values

By calculating a feature's Shapley value, which is its average marginal contribution across all possible coalitions, we can generate local explanations for instance in interest. The sum of Shapley values for all features yields the difference between the actual prediction of a single instance and the average prediction. In other words, if we estimate Shapley values for all features, we will get a complete distribution of actual prediction minus the average prediction among the feature values.

3.3 Fairness Evaluations

In this section we conduct some common fairness tests on the models trained on the loan dataset and the healthcare dataset. We pick out sensitive attributes that should be independent of the target variable base on human knowledge. For each sensitive attribute, we run multiple fairness evaluation methods based on the models' accuracy, predictivity, and causal reasoning.

3.3.1 Sensitive Attributes

There were various sensitive attributes in both the loan dataset and healthcare dataset that had the potential to disrupt the fairness and accuracy of the models built. In the loan dataset, gender is the most sensitive attribute. In the healthcare dataset, race, gender, and age are the most sensitive attributes most susceptible to cause biased model predictions. In the loan dataset, gender should be independent of the target variable because the model may associate a specific gender with a greater likelihood of loan defaults simply due to the trends it learns in the sample of data used for training. For example, the model may predict female individuals as more likely to default on loans if they have lower incomes or no credit; this may only be the case because women may not be breadwinners if they are married and thus may have no credit. However, the model would indicate a high probability of default if an individual is female when in fact the two variables are not directly correlated and gender itself is not a predictor. The same phenomenon applies to race, gender, and age in the healthcare dataset. A sensitive attribute like race should be independent of the target variable representing hospital readmission because the model may associate a particular race as being more likely to be readmitted to a hospital. Other hidden factors such as insulin level, occupation, and level of physical activity may make it appear that a particular race has a higher risk of readmission when in fact race by itself is not a predictor of readmission.

3.3.2 Fairness Evaluate Methods

In this section, we use the following definitions:

X: All features given to the model

T: True value of the target variable (binary variable)

Y: Model prediction (binary variable)

p: Model predicted probability (continuous variable in [0~1])

S: Sensitive attribute

The following method are the methods we use to evaluate fairness:

1) Group fairness:

   A fair model's prediction should be independent of the sensitive attribute. Therefore, it should have the same probability of giving a positive prediction for individuals of different protected classes. In this test we check if $P(Y = 1|S = s_i) = P(Y = 1|S = s_j)$. Notice that this test does not require the actual value of the target variable. In other words, the test is independent of whether the model makes the correct predictions.

2) Predictive parity:

   This test measures the model's predictive power on different groups of the protected class. The probability of the model making the correct prediction should be the same across different groups. In this test, we check if the true positive rates are the same among groups: $P(T = 1|Y = 1, S= s_i) = P(T = 1|Y = 1, S = s_j)$. The method can be also be applied with different prediction evaluation metrics.

3) Matching conditional frequencies:

   This test is similar to the predictive parity test, except we consider the distribution of predicted probabilities rather than the binarized prediction. We binned the predicted probabilities and compare the frequencies of each bin across different groups. For each bin, We check if $P(T = 1|Y \in bin_k, S= s_i) = P(T = 1|Y \in bin_k, S = s_j)$

4) Causal discrimination

The model is counterfactually fair if its prediction is independent of the change of sensitive variable. We conduct the test by flipping or randomly shuffling the sensitive attribute of the test set and checking if the prediction remains the same.

3.4 Explaining False Positive (FP) and False Negative (FN) Predictions

In this section, we are interested in explaining both false positive and false negative predictions made by different black-box models. In other words, we would like to explore reasons why a particular model predicts positive outcomes whereas the true label is negative and why a model predicts negative outcomes whereas the true label is positive. To analyze FP and FN predictions, we will be using causal inferences in general, and the details of each step are described below.

3.4.1 Changing Labels

To begin with, we are going to change the label of a dataset to denote whether a prediction is FP or FN. For example, in the original Airbnb dataset, the label encodes whether a price of a listing is above or below average with values 0 and 1. In the case of FP analysis, we will be changing the label to 0 and 1, where value 1 denotes that a prediction of the model is FP and value 0 denotes that a prediction of the model is not FP.

3.4.2 Modeling

To do causal inference, we first need to model the assumptions and create causal graphs that explain the causal relationships between variables, including the treatment and outcome that we are interested in.

3.4.3 Identification

After reasoning about the problem with causal graphs, we need to identify an estimable quantity that represents $P(Y|do(T))$ generated by the intervention graph. Also, the desired quantity should be calculated using statistical observations alone. We call it an identification step.

3.4.4 Estimation

After identifying the estimable quantity, we use the observed data to compute the target probability expression. For binary treatments, the causal effect is $E[Y|T = 1, W = w] - E[Y|T = 0, W = w]$. The ultimate goal of estimation is to estimate $p(Y|T = t)$ when all confounders W are kept constant.

3.4.5 Robustness Check

After conducting estimation, we need to test the robustness of the estimate to the violation of assumptions. One example is using a Placebo Treatment Refuter by replacing the treatment variable with a randomly-generated variable and seeing if the estimate is zero to check if the treatment really causes the outcome. To check the sensitivity of an estimate to a new confounder, we can use an Unobserved Confounder Refuter by simulating a confounder based on a given correlation with both treatment and outcome and rerun the analysis to see if the direction of the estimate flips.

3.5 Counterfactual Recourse

For the counterfactual recourse, we plan to use the LEWIS algorithm that is developed by Sainyam Galhotra, Romila Pradhan, and Babak Salimi. For an individual who got an undesired outcome from the model, LEWIS can provide minimal interventions on a set of actionable variables that have a high sufficiency score for the individual to get a different outcome (Galhotra, Pradhan, Salimi, 7).

4. Results

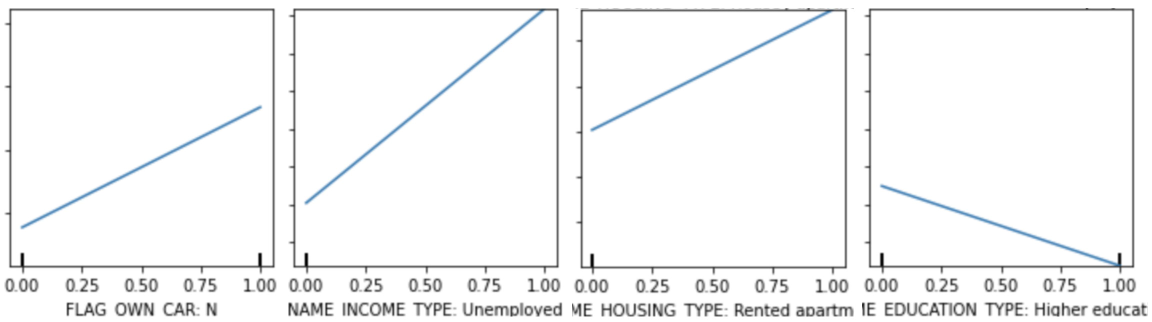
4.1 Model Explanations

4.1.1 Global Explanations

In this section, we will present several interesting examples of PDP and Permutation Feature Importance.

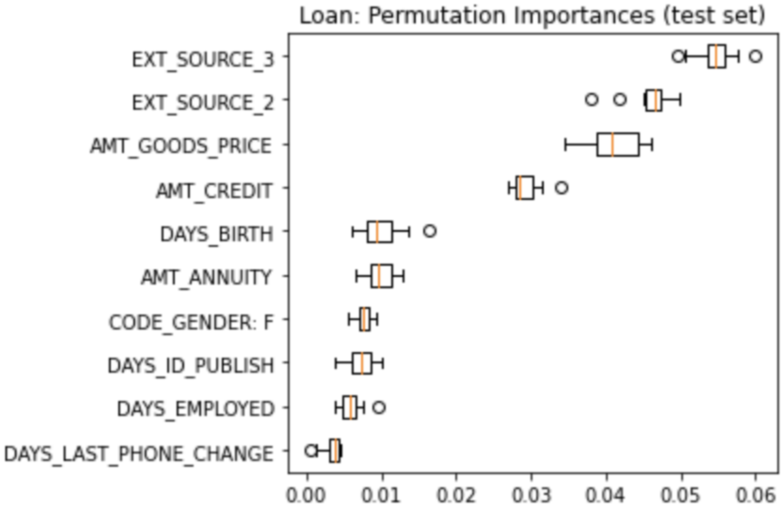
Example from loan dataset

Below is the PDPs of ‘FLAG_OWN_CAR: N’, ‘NAME_INCOME_TYPE: Unemployed’, ‘NAME_HOUSING_TYPE: Rented apartment’, ‘NAME_EDUCATION_TYPE: Higher Education’ from the **loan** dataset that uses the XGBoost model.



Based on the partial dependence plots, we can tell that when the value of ‘FLAG_OWN_CAR: N’, ‘NAME_INCOME_TYPE: Unemployed’, and ‘NAME_HOUSING_TYPE: Rented apartment’ changes from 0 to 1, the average prediction in the dataset increases. And when the value of ‘NAME_EDUCATION_TYPE: Higher Education’ changes from 0 to 1, the average prediction decreases. Combining the domain knowledge, we can tell that the four plots show an accurate relationship: in reality, a person is more likely to be a defaulter if the person doesn’t have a car, doesn’t have a job, or is renting apartments. And a person with higher education is more likely to get a loan.

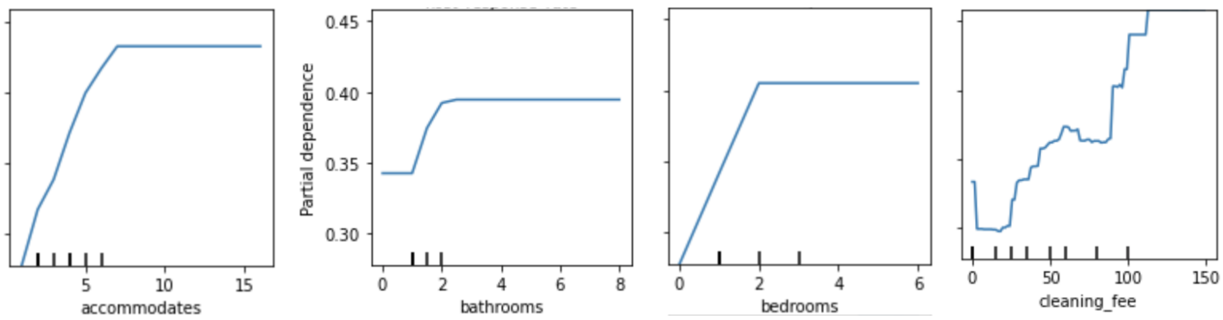
The graph below shows the ten most important features generated by permutation feature importance on the test set of loan data that uses the XGBoost Model.



According to the permutation importance graph, we can see that except for the features that are determined by an external source (‘EXT_SOURCE_3’ and ‘EXT_SOURCE_2’), ‘AMT_GOODS_PRICE’ and ‘AMT_CREDIT’ are the two most important features. So we can see that the price of the goods for which the loan is given and the credit amount of the loan are very important in a loan application. ‘DAYS_BIRTH’, which represents the applicant’s age in days at the time of application is also important. To evaluate the feature importances using domain knowledge, we can tell that the ranking of features based on their importance is quite accurate as attributes like the amount of money a person is applying for, the age of the applicant, and the applicant's days of employment are definitely heavily considered during the loan application process.

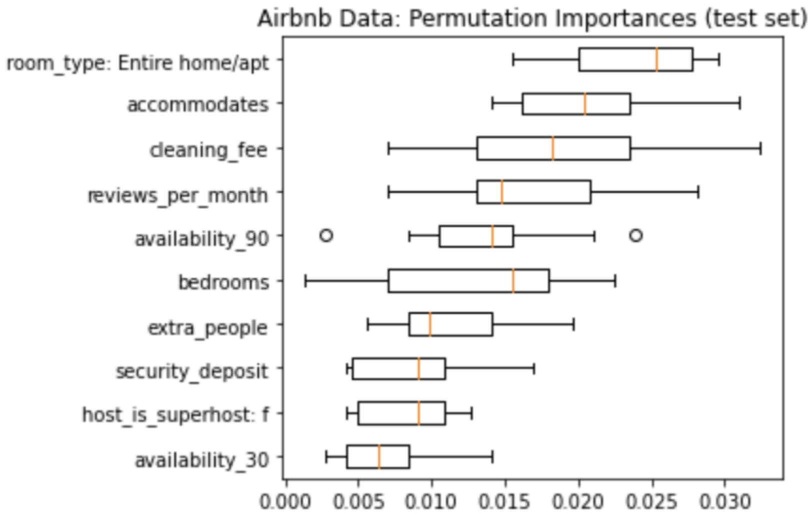
Example from Airbnb dataset

Below are the PDPs of ‘accommodates’, ‘bathrooms’, ‘bedrooms’, and ‘cleaning_fee’ in the **Airbnb** dataset that uses the XGBoost model.



From the partial dependence plots, we can see that when the number of accommodates, bathrooms, bedrooms, or cleaning fees increases, the average prediction increase (higher price). Combining domain knowledge, we can say that the PDPs are accurate because usually, Airbnb homes with more accommodates, bathrooms, bedrooms, higher cleaning fees charge a higher price.

The graph below shows the ten most important features in the test set of the Airbnb dataset.



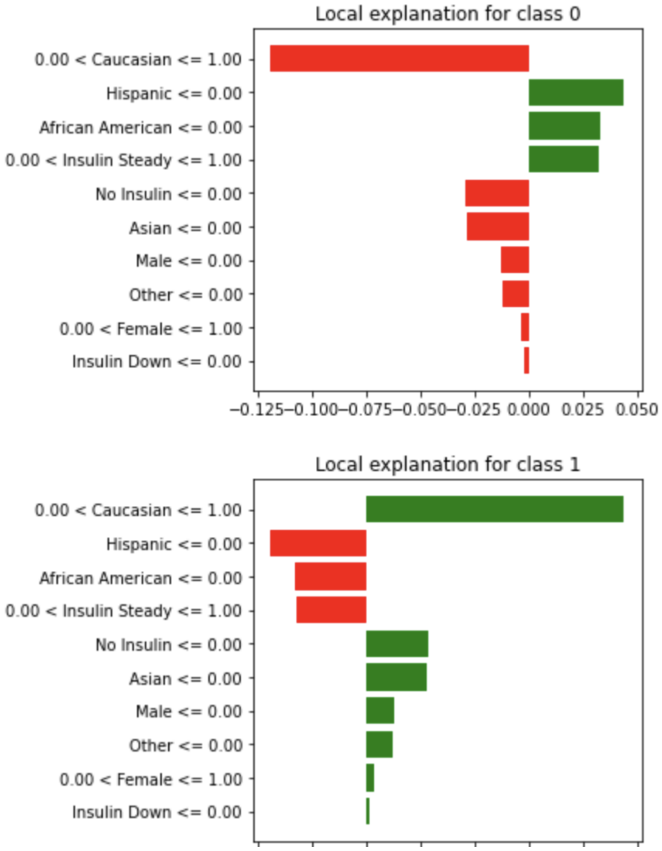
'Room_type: Entire home/apt', 'accommodates', and 'cleaning_fee' are the TOP 3 most important features that can determine the price of an Airbnb home. To evaluate the permutation importances using domain knowledge, we can say that the important features make a lot of sense because people usually pay more for things like an entire home, more accommodates, or a home that charges higher cleaning_fee.

4.1.2 Local Explanations

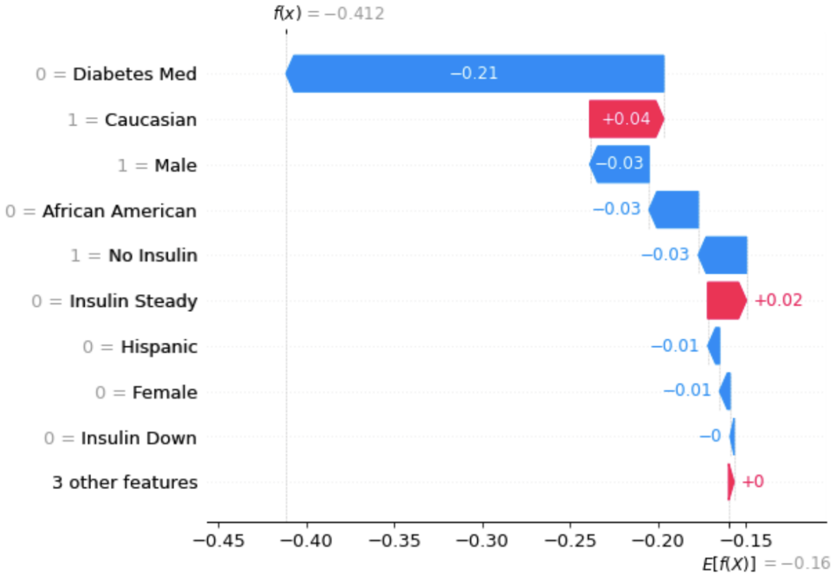
In this section, we will present examples of local explanations for an individual in the healthcare dataset and an individual in the loan dataset that uses the XGBoost model.

Example from Healthcare dataset

The graph below shows the LIME for one individual in the test set of the **healthcare** dataset. This individual is a Caucasian female with steady insulin who is taking diabetes medications. The predicted outcome for this individual is 0, which means that this individual is predicted to not be readmitted to the hospital. According to the graph, we can tell that not being a Hispanic contributes the most to being predicted to class 0, then is not being an African American.



The graph below shows the Shapley values for the same individual in the healthcare dataset.

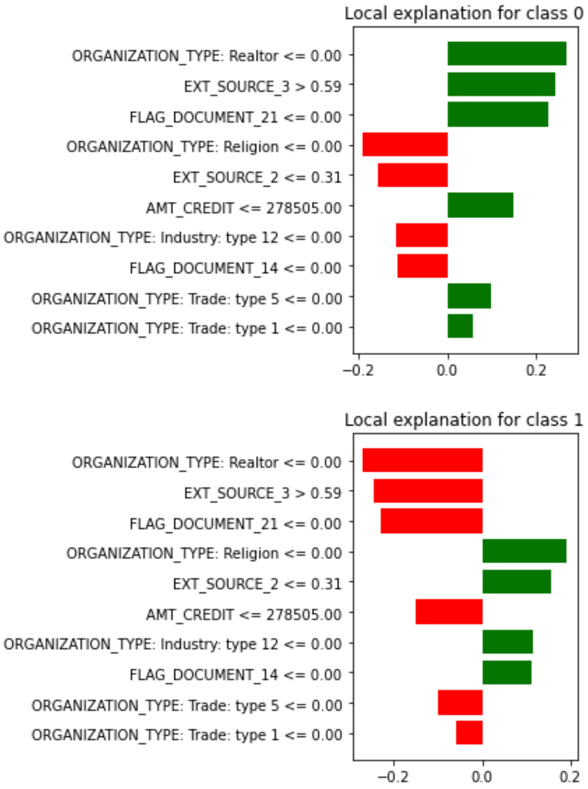


From the graph above, we can see that the average prediction is -0.16, and the actual prediction for this individual is -0.412. And not on Diabetes Med has the most effect of dragging the prediction closer to -0.412. Being a Caucasian has the most effect of dragging the prediction the most closer to -0.16.

Example from loan dataset

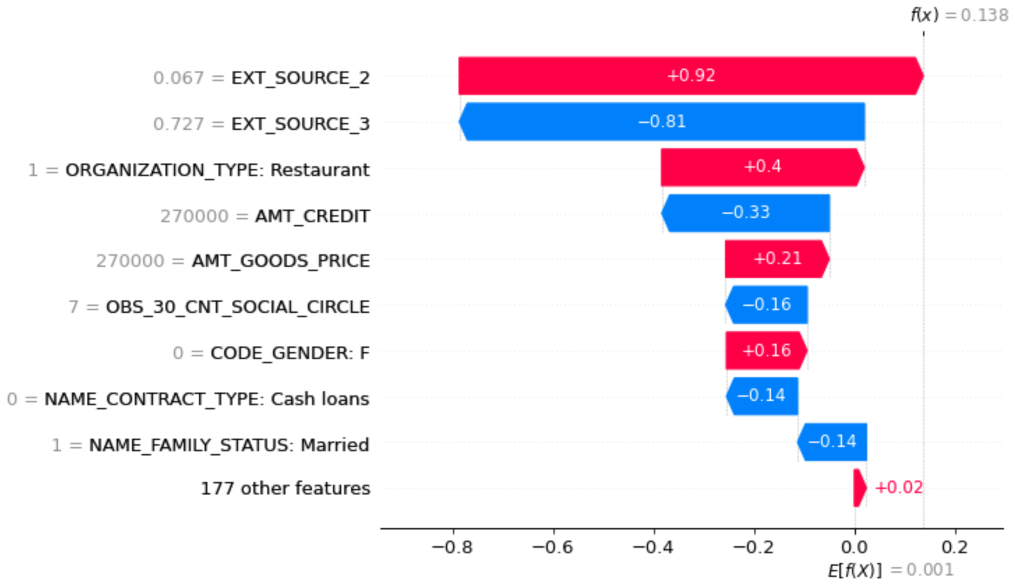
The individual that we choose for the **loan** dataset is predicted to be a defaulter. To see the complete attributes of this individual, please go to *Table 1* in the appendix.

```
Predicted:  1
```



According to the graph above, we can see that not working in a religion-related organization contributes the most to being predicted to a defaulter, and not working in a realtor-related organization has the most negative impact on being predicted to a defaulter.

The graph below shows the Shapley values for the same individual in the loan dataset.



From the graph above, we can see that the average prediction is 0.001, and the actual prediction for this individual is 0.138. Except for the features that are determined by an external source (‘EXT_SOURCE_3’ and ‘EXT_SOURCE_2’), having a 270000 amount of credit has the most effect of dragging the prediction closer to 0.001. And working in a restaurant organization has the most effect of dragging the prediction closer to 0.138.

4.2 Fairness Evaluations

For the result of the fairness evaluation we present one example for each evaluation method to showcase the process and outcome for those methods. In each example, we pick a dataset, a model, and a sensitive variable, which a fair model should be unbiased to. In other words, the model should rely on these variables to make its prediction, introducing discrimination against a protected class. The set of sensitive depends on the task and is often required to be determined with domain knowledge. For example, gender might be a sensitive variable in many cases, but in health care, gender might be an important predictor of certain diseases. In the following examples, we assume the selected sensitive variables are indeed sensitive for the corresponding task. Another assumption we made is that our dataset represents the population. All the evaluations are made on a separate test dataset that the model hasn’t been exposed to.

4.2.1 Group Fairness

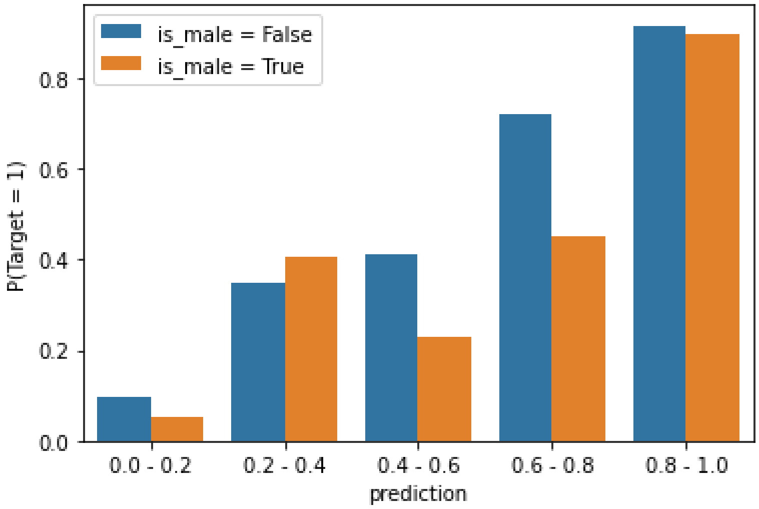
The model we evaluate is the SVM classifier trained on the loan data and the sensitive variable is gender. In this test, we measure the probability of the model giving a positive prediction on male and female individuals. A fair model should rely on the sensitive variable and thus should have the same average prediction on the two groups. The trained SVM classifier has a 0.61 average prediction on the female subjects and 0.7 on the male subjects. The expectation difference is about 0.09. In other words, the model has a bias toward giving male loan borrowers a higher default probability. Therefore, we concluded that the SVM classifier is not fair.

4.2.2 Predictive Parity

The model we evaluate is the XGBoost classifier trained on the loan data. The sensitive variable we picked is marital status, a binary variable indicating if the borrower is married. It is debatable whether marital status is a sensitive variable in loan default prediction. In this experiment, we assume it is and we compute the predictive parity between the married individual and the not married. We compute the true positive rate of the model on the two groups. A fair model should have the same true positive rate on both groups. The not married group has a true positive rate of 0.675 while the married group has a true positive rate of 0.674. The predictivity difference is only 0.001. The model has similar predictive power on the two groups of individuals. Therefore, we conclude that the model is fair in terms of marital status under the predictive parity test.

4.2.3 Conditional Frequencies

The model we evaluate is the XGBoost classifier trained on the Airbnb dataset. We randomly generate the owner’s gender and use it as the sensitive variable. The generated gender is not correlated with the target at all. The correlation coefficient of the two variables is 0.02. The result is shown in the figure below. Each bar of the figure represents the true probability of an instance being positive given that the model prediction is within a certain range. From the graph, we can see that when the model's prediction is within the 0.4~0.6 bin or the 0.6~0.8 bin, male Airbnb host has a significantly lower true probability of being in a positive class. In other words, the model overpredicted on instances with the male owners. Hence, the model is unfair.



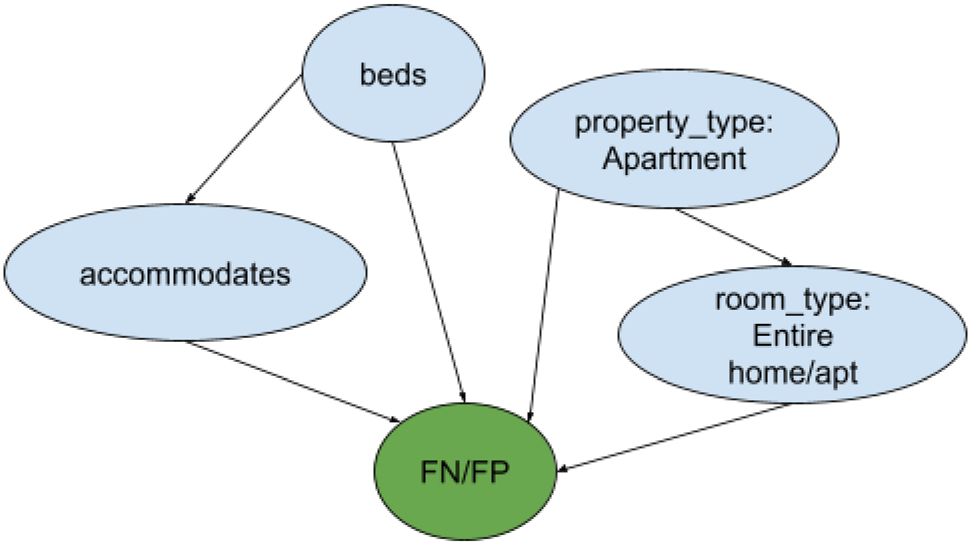
4.2.4 Causal Discrimination

In this experiment, the model we evaluate is the TabNet classifier trained on the Loan dataset and the sensitive variable is gender. We flip the gender of all the instances and measure the average prediction change for each gender. After only flipping the gender feature, 21.0% of the instances have their prediction changed. Changing the gender of the loan borrower from female to male increases the prediction by 11.2% on average, and changing the gender from male to female decreases the prediction by 9.6% on average. The result indicates that the model is unfair and relies on gender to make its prediction. The model is biased toward giving male loan borrowers a higher prediction of default probability. The result confirmed our findings in the Group Fairness experiment. Both models show the same bias, which suggested the dataset is biased. We find that the dataset indeed has a 0.1 correlation between gender and whether the loan defaults. The extent of the bias, 10%, aligned with the result of the Causal Discrimination test and Group Fairness test.

4.3 False Positive and False Negative Explanations

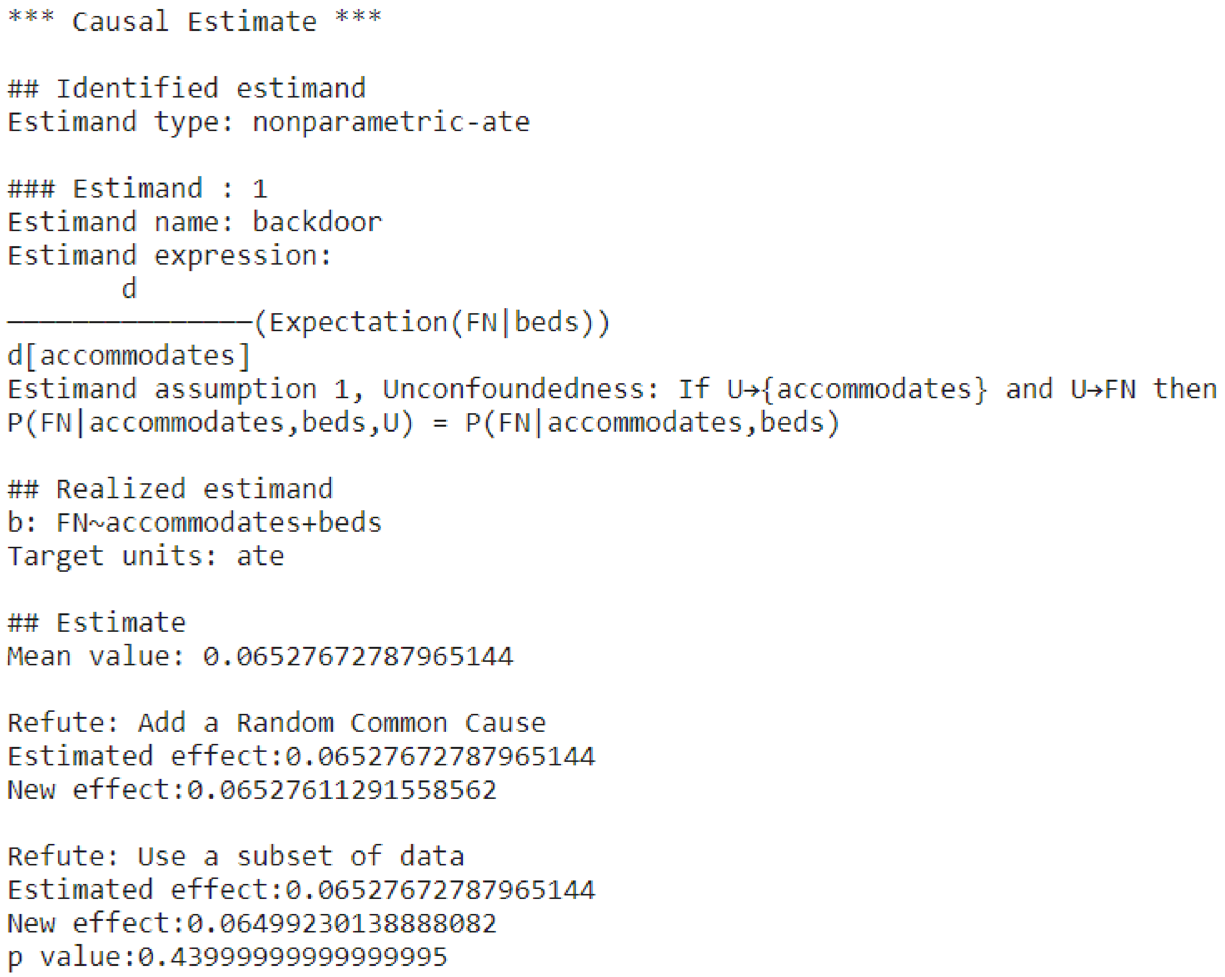
We analyzed both false positives and false negatives predictions of multiple black-box models on various datasets. In this section, we will be using the results found in the Airbnb dataset as an example of our analysis.

We started with explanations of false-negative (FN) predictions. We modified the label and set it to values 0 and 1, where value 1 represents that the prediction of a model is FN and value 0 presents that the prediction of a model is not FN. The same procedure applies to FP explanations as well. Then, using the domain knowledge, we constructed a simple causal graph below with several features of interest.



To clarify, the node in green is the outcome we are interested in, the nodes in blue are the features of interests, and the arrows denote the causal relationship between two nodes. For example, the accommodates, which is the maximum capacity of a listing, might cause a predictor to predict false negatives, and beds, which is the number of beds in the listing, might be a common cause of both accommodates and FN outcomes. To determine if this is the case, we used the DoWhy, an end-to-end library to do causal inference, to identify the causal effect, estimate the identified estimand, and verify the results.

Here is an example of how the feature “accommodates” causes the SVC model to predict false negatives.

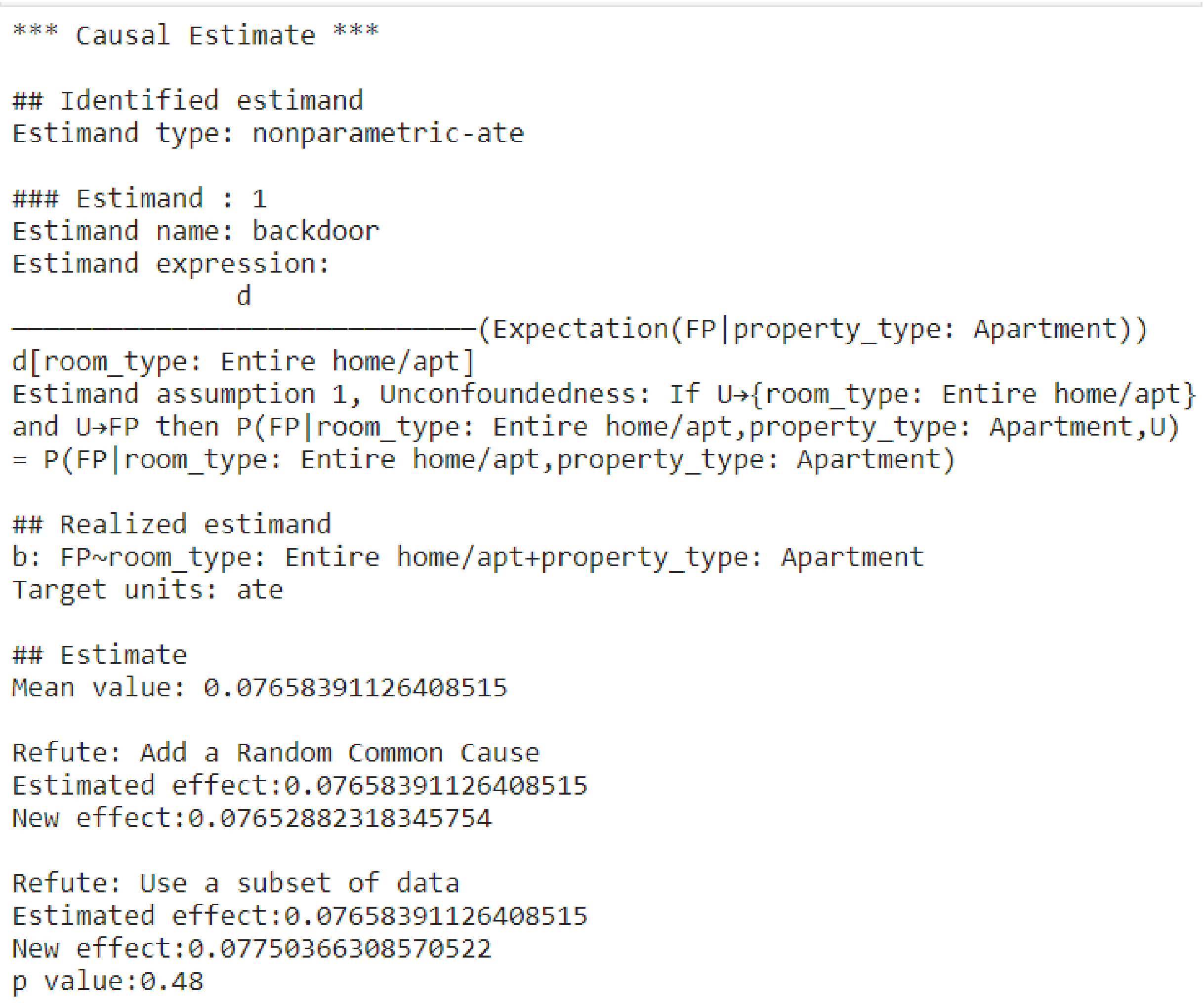


```
*** Causal Estimate ***

## Identified estimand
Estimand type: nonparametric-ate

### Estimand : 1
Estimand name: backdoor
Estimand expression:
       d
───────────────(Expectation(FN|beds))
d[accommodates]
Estimand assumption 1, Unconfoundedness: If U→{accommodates} and U→FN then
P(FN|accommodates,beds,U) = P(FN|accommodates,beds)

## Realized estimand
b: FN~accommodates+beds
Target units: ate

## Estimate
Mean value: 0.06527672787965144

Refute: Add a Random Common Cause
Estimated effect:0.06527672787965144
New effect:0.06527611291558562

Refute: Use a subset of data
Estimated effect:0.06527672787965144
New effect:0.06499230138888082
p value:0.43999999999999995
```

First, the DoWhy causal model sets the feature “accommodates” to be the treatment and identifies the confounder “beds” using the backdoor criterion. Then, the DoWhy causal model simulates the randomized experiment by adjusting the influence of some confounding variables Z, and it is “beds” in this case. The adjustment formula is presented by $p(P|do(T))=\sum_{z} p(Y|T,\ Z)p(Z)$. After identifying the estimand, the DoWhy causal model estimates the estimand using the Average Treatment Effect (ATE), and the result is about 0.065. Thus, having a higher maximum capacity in a listing increases the chance of the SVC predicting FN. To validate the result, we use two refutation tests. The first one is called Random Common Cause which adds randomly drawn covariates to data and re-runs the analysis to see if the causal estimate changes or not. If our assumption was originally correct then the causal estimate shouldn't change by much. In this case, the estimated effect and the new effect are about the same. The second test is called Data Subset Refuter which creates subsets of the data(similar to cross-validation) and checks whether the causal estimates vary across subsets. If our assumptions were correct there shouldn't be many variations. In our example, the estimated effect and the new effect is not significantly different according to the p-value. We can see that our estimate passes all refutation tests. This does not prove its correctness, but it increases confidence in the estimate.

In conclusion, after the causal inference, we can say that having a higher maximum capacity in a listing increases the chance of the SVC predicting FN. This is intuitive because a listing with a higher maximum capacity is more likely to be expensive in reality, so the model will give FN predictions if the model fails to capture this positive correlation.

Here is another example of how room type causes the LGBM model to predict false positives.



```
*** Causal Estimate ***

## Identified estimand
Estimand type: nonparametric-ate

### Estimand : 1
Estimand name: backdoor
Estimand expression:
              d
─────────────────────────────(Expectation(FP|property_type: Apartment))
d[room_type: Entire home/apt]
Estimand assumption 1, Unconfoundedness: If U→{room_type: Entire home/apt}
and U→FP then P(FP|room_type: Entire home/apt,property_type: Apartment,U)
= P(FP|room_type: Entire home/apt,property_type: Apartment)

## Realized estimand
b: FP~room_type: Entire home/apt+property_type: Apartment
Target units: ate

## Estimate
Mean value: 0.07658391126408515

Refute: Add a Random Common Cause
Estimated effect:0.07658391126408515
New effect:0.07652882318345754

Refute: Use a subset of data
Estimated effect:0.07658391126408515
New effect:0.07750366308570522
p value:0.48
```

The same procedure used to generate causal inference for the previous example applies here as well. The DoWhy model first sets the “room_type” to be the treatment and identifies the cofounder “property_type” and tries to estimate. Then, it estimates the ATE to be around 0.077. So, the predictor tends to give FP predictions if the room type is the entire home/apt. The estimate also passes all refutation tests. Again, this does not prove its correctness, but it increases confidence in the estimate. The result makes sense because if a model puts too much weight on the room type, it will predict the higher prices for listings with an entire place, while this is not always the case in reality. Sometimes a shared place could be more expensive than an entire apartment depending on the location and other factors.

4.4 Counterfactual Recourse

We generated counterfactual recourse explanations for an individual from the loan dataset negatively impacted by the model's prediction. The individual was a male, approximately 32 years old, was unemployed, did not own a car, did not own a home, did not possess higher education, and was classified by the model as a loan defaulter. There were several actionable attributes that the individual could have taken action on to avoid being classified as a loan defaulter.

Actionable Attribute	**Current Value**	**Required Value**
'NAME_INCOME_TYPE: Unemployed'	1	0
'FLAG_OWN_CAR: N'	1	0
'NAME_HOUSING_TYPE: Rented apartment'	1	0
'NAME_EDUCATION_TYPE: Higher Education'	0	1

To significantly decrease the likelihood of being a loan defaulter, the individual should be employed. Employment status is a significant predictor in the model and negatively affects unemployed individuals. The individual should also prove homeownership and car ownership when applying for a loan to decrease their likelihood of being classified as a loan defaulter. In addition, the individual should apply for a loan when they have obtained the highest educational level possibly, ideally higher education. The educational level of an applicant is an attribute that affects their income status and homeownership status, and is, therefore, a major predictor in the model.

5. References

Fairness Definitions Explained. Verma Sahil, Rubin Julia. Retrieved February 3, 2022, from https://fairware.cs.umass.edu/papers/Verma.pdf

Loan Defaulter. Gaurav Dutta. Retrieved December 3, 2021, from https://www.kaggle.com/gauravduttakiit/loan-defaulter

Molnar, C. (2021, November 11). Interpretable machine learning. 9.5 Shapley Values. Retrieved December 2, 2021, from https://christophm.github.io/interpretable-ml-book/shapley.html.

Prediction on Hospital Readmission. Abhishek Sharma. Retrieved February 4, 2022, from https://www.kaggle.com/iabhishekofficial/prediction-on-hospital-readmission

Sainyam Galhotra, Romila Pradhan, & Babak Salimi. (2021). Explaining Black-Box Algorithms Using Probabilistic Contrastive Counterfactuals.

Seattle Airbnb Open Data. Airbnb. Retrieved December 1, 2021, from https://www.kaggle.com/airbnb/seattle?select=listings.csv

Sharma A. & Kiciman E., Foundations of Causal Inference and its impacts on Machine Learning, Webinar.

6. Appendix

Link 1: Link to the Project Proposal that we wrote last quarter for reference purposes:

https://docs.google.com/document/d/1ySit8KEDxtwS3NWKTko_ZTBpjzFVZXwxeMHgBX_li2M/edit?usp=sharing

Table 1: Attributes for the individual in loan dataset in the local explanation example

Attribute	**Value**
NAME_CONTRACT_TYPE: Cash loans	0.0

NAME_CONTRACT_TYPE: Revolving loans	1.0
CODE_GENDER: F	0.0
CODE_GENDER: M	1.0
CODE_GENDER: XNA	0.0
FLAG_OWN_CAR: N	1.0
FLAG_OWN_CAR: Y	0.0
FLAG_OWN_REALTY: N	0.0
FLAG_OWN_REALTY: Y	1.0
NAME_TYPE_SUITE: Children	0.0
NAME_TYPE_SUITE: Family	0.0
NAME_TYPE_SUITE: Group of people	0.0
NAME_TYPE_SUITE: Other_A	0.0
NAME_TYPE_SUITE: Other_B	0.0
NAME_TYPE_SUITE: Spouse, partner	0.0
NAME_TYPE_SUITE: Unaccompanied	1.0
NAME_TYPE_SUITE: nan	0.0
NAME_INCOME_TYPE: Businessman	0.0
NAME_INCOME_TYPE: Commercial associate	0.0
NAME_INCOME_TYPE: Maternity leave	0.0
NAME_INCOME_TYPE: Pensioner	0.0
NAME_INCOME_TYPE: State servant	0.0
NAME_INCOME_TYPE: Student	0.0
NAME_INCOME_TYPE: Unemployed	0.0
NAME_INCOME_TYPE: Working	1.0
NAME_EDUCATION_TYPE: Academic degree	0.0
NAME_EDUCATION_TYPE: Higher education	1.0

NAME_EDUCATION_TYPE: Incomplete higher	0.0
NAME_EDUCATION_TYPE: Lower secondary	0.0
NAME_EDUCATION_TYPE: Secondary / secondary special	0.0
NAME_FAMILY_STATUS: Civil marriage	0.0
NAME_FAMILY_STATUS: Married	1.0
NAME_FAMILY_STATUS: Separated	0.0
NAME_FAMILY_STATUS: Single / not married	0.0
NAME_FAMILY_STATUS: Unknown	0.0
NAME_FAMILY_STATUS: Widow	0.0
NAME_HOUSING_TYPE: Co-op apartment	0.0
NAME_HOUSING_TYPE: House / apartment	1.0
NAME_HOUSING_TYPE: Municipal apartment	0.0
NAME_HOUSING_TYPE: Office apartment	0.0
NAME_HOUSING_TYPE: Rented apartment	0.0
NAME_HOUSING_TYPE: With parents	0.0
WEEKDAY_APPR_PROCESS_START: FRIDAY	0.0
WEEKDAY_APPR_PROCESS_START: MONDAY	0.0
WEEKDAY_APPR_PROCESS_START: SATURDAY	0.0
WEEKDAY_APPR_PROCESS_START: SUNDAY	0.0
WEEKDAY_APPR_PROCESS_START: THURSDAY	0.0
WEEKDAY_APPR_PROCESS_START: TUESDAY	0.0
WEEKDAY_APPR_PROCESS_START: WEDNESDAY	1.0
OCCUPATION_TYPE: Accountants	0.0
OCCUPATION_TYPE: Cleaning staff	0.0
OCCUPATION_TYPE: Cooking staff	0.0
OCCUPATION_TYPE: Core staff	0.0

OCCUPATION_TYPE: Drivers	0.0
OCCUPATION_TYPE: HR staff	0.0
OCCUPATION_TYPE: High skill tech staff	0.0
OCCUPATION_TYPE: IT staff	0.0
OCCUPATION_TYPE: Laborers	0.0
OCCUPATION_TYPE: Low-skill Laborers	0.0
OCCUPATION_TYPE: Managers	0.0
OCCUPATION_TYPE: Medicine staff	0.0
OCCUPATION_TYPE: Private service staff	0.0
OCCUPATION_TYPE: Realty agents	0.0
OCCUPATION_TYPE: Sales staff	0.0
OCCUPATION_TYPE: Secretaries	0.0
OCCUPATION_TYPE: Security staff	1.0
OCCUPATION_TYPE: Waiters/barmen staff	0.0
OCCUPATION_TYPE: nan	0.0
ORGANIZATION_TYPE: Advertising	0.0
ORGANIZATION_TYPE: Agriculture	0.0
ORGANIZATION_TYPE: Bank	0.0
ORGANIZATION_TYPE: Business Entity Type 1	0.0
ORGANIZATION_TYPE: Business Entity Type 2	0.0
ORGANIZATION_TYPE: Business Entity Type 3	0.0
ORGANIZATION_TYPE: Cleaning	0.0
ORGANIZATION_TYPE: Construction	0.0
ORGANIZATION_TYPE: Culture	0.0
ORGANIZATION_TYPE: Electricity	0.0
ORGANIZATION_TYPE: Emergency	0.0

ORGANIZATION_TYPE: Government	0.0
ORGANIZATION_TYPE: Hotel	0.0
ORGANIZATION_TYPE: Housing	0.0
ORGANIZATION_TYPE: Industry: type 1	0.0
ORGANIZATION_TYPE: Industry: type 10	0.0
ORGANIZATION_TYPE: Industry: type 11	0.0
ORGANIZATION_TYPE: Industry: type 12	0.0
ORGANIZATION_TYPE: Industry: type 13	0.0
ORGANIZATION_TYPE: Industry: type 2	0.0
ORGANIZATION_TYPE: Industry: type 3	0.0
ORGANIZATION_TYPE: Industry: type 4	0.0
ORGANIZATION_TYPE: Industry: type 5	0.0
ORGANIZATION_TYPE: Industry: type 6	0.0
ORGANIZATION_TYPE: Industry: type 7	0.0
ORGANIZATION_TYPE: Industry: type 8	0.0
ORGANIZATION_TYPE: Industry: type 9	0.0
ORGANIZATION_TYPE: Insurance	0.0
ORGANIZATION_TYPE: Kindergarten	0.0
ORGANIZATION_TYPE: Legal Services	0.0
ORGANIZATION_TYPE: Medicine	0.0
ORGANIZATION_TYPE: Military	0.0
ORGANIZATION_TYPE: Mobile	0.0
ORGANIZATION_TYPE: Other	0.0
ORGANIZATION_TYPE: Police	0.0
ORGANIZATION_TYPE: Postal	0.0
ORGANIZATION_TYPE: Realtor	0.0

ORGANIZATION_TYPE: Religion	0.0
ORGANIZATION_TYPE: Restaurant	1.0
ORGANIZATION_TYPE: School	0.0
ORGANIZATION_TYPE: Security	0.0
ORGANIZATION_TYPE: Security Ministries	0.0
ORGANIZATION_TYPE: Self-employed	0.0
ORGANIZATION_TYPE: Services	0.0
ORGANIZATION_TYPE: Telecom	0.0
ORGANIZATION_TYPE: Trade: type 1	0.0
ORGANIZATION_TYPE: Trade: type 2	0.0
ORGANIZATION_TYPE: Trade: type 3	0.0
ORGANIZATION_TYPE: Trade: type 4	0.0
ORGANIZATION_TYPE: Trade: type 5	0.0
ORGANIZATION_TYPE: Trade: type 6	0.0
ORGANIZATION_TYPE: Trade: type 7	0.0
ORGANIZATION_TYPE: Transport: type 1	0.0
ORGANIZATION_TYPE: Transport: type 2	0.0
ORGANIZATION_TYPE: Transport: type 3	0.0
ORGANIZATION_TYPE: Transport: type 4	0.0
ORGANIZATION_TYPE: University	0.0
ORGANIZATION_TYPE: XNA	0.0
SK_ID_CURR	183124.0
CNT_CHILDREN	0.0
AMT_INCOME_TOTAL	90000.0
AMT_CREDIT	270000.0
AMT_ANNUITY	13500.0

AMT_GOODS_PRICE	270000.0
REGION_POPULATION_RELATIVE	0.018209
DAYS_BIRTH	-21252.0
DAYS_EMPLOYED	-1899.0
DAYS_REGISTRATION	-12311.0
DAYS_ID_PUBLISH	-4425.0
FLAG_MOBIL	1.0
FLAG_EMP_PHONE	1.0
FLAG_WORK_PHONE	0.0
FLAG_CONT_MOBILE	1.0
FLAG_PHONE	1.0
FLAG_EMAIL	0.0
CNT_FAM_MEMBERS	2.0
REGION_RATING_CLIENT	3.0
REGION_RATING_CLIENT_W_CITY	3.0
HOUR_APPR_PROCESS_START	13.0
REG_REGION_NOT_LIVE_REGION	0.0
REG_REGION_NOT_WORK_REGION	0.0
LIVE_REGION_NOT_WORK_REGION	0.0
REG_CITY_NOT_LIVE_CITY	0.0
REG_CITY_NOT_WORK_CITY	0.0
LIVE_CITY_NOT_WORK_CITY	0.0
EXT_SOURCE_2	0.0672882842760214
EXT_SOURCE_3	0.7267112092725120
OBS_30_CNT_SOCIAL_CIRCLE	7.0
DEF_30_CNT_SOCIAL_CIRCLE	0.0

OBS_60_CNT_SOCIAL_CIRCLE	7.0
DEF_60_CNT_SOCIAL_CIRCLE	0.0
DAYS_LAST_PHONE_CHANGE	-1042.0
FLAG_DOCUMENT_2	0.0
FLAG_DOCUMENT_3	0.0
FLAG_DOCUMENT_4	0.0
FLAG_DOCUMENT_5	0.0
FLAG_DOCUMENT_6	0.0
FLAG_DOCUMENT_7	0.0
FLAG_DOCUMENT_8	0.0
FLAG_DOCUMENT_9	0.0
FLAG_DOCUMENT_10	0.0
FLAG_DOCUMENT_11	0.0
FLAG_DOCUMENT_12	0.0
FLAG_DOCUMENT_13	0.0
FLAG_DOCUMENT_14	0.0
FLAG_DOCUMENT_15	0.0
FLAG_DOCUMENT_16	0.0
FLAG_DOCUMENT_17	0.0
FLAG_DOCUMENT_18	0.0
FLAG_DOCUMENT_19	0.0
FLAG_DOCUMENT_20	0.0
FLAG_DOCUMENT_21	0.0
AMT_REQ_CREDIT_BUREAU_HOUR	0.0
AMT_REQ_CREDIT_BUREAU_DAY	0.0
AMT_REQ_CREDIT_BUREAU_WEEK	0.0

AMT_REQ_CREDIT_BUREAU_MON	0.0
AMT_REQ_CREDIT_BUREAU_QRT	0.0
AMT_REQ_CREDIT_BUREAU_YEAR	3.0