Recruitment Case Study

FABIO PULVIRENTI

Customers Data

Several intertwined data collections

- Customers information (Age, Gender, County, Income)
- Customers Bank Information
  - Held Loan Previously
  - Products
  - Transactions Statistics
  - **Holding a Loan**
- Geographic Information

Some Data Cleaning required

The datasets were not ready to be used

- Extra Columns / Rows
- Missing Values (some of them consistent)
  - Most frequent values
  - Average (numeric features)
- Typos
  - Wrong Labels
  - Misspelled Counties
- Inconsistency:
  - Female, Male (0,1) vs (1,2)
  - Negative numbers of products
- One of the datasets has been changed with the dual one used for the submission
- Repeated Client IDs
  - The same in all the datasets, those users have been kept

Some Data Cleaning required

The datasets were not ready to be used

- Extra Columns / Rows
- Missing Values (some of them consistent)
  - Most frequent values
  - Average (numeric features)
- Typos
  - Wrong Labels
  - Misspelled Counties
- Inconsistency:
  - Female, Male (0,1) vs (1,2)
  - Negative numbers of products
- One of the datasets has been changed with the dual one used for the submission
- Repeated Client IDs
  - The same in all the datasets, those users have been kept

More details in the attached coding notebook

Predictive Modelling

Target:

- Identify the customers more likely to take on a loan

Steps:

- Feature Transformation
- Feature Identification
- Model evaluation
- Model Tuning
- Likelihood groups identification

Feature Transformation (Examples)

Feature: Income Group

- 0 – 10000, 10001 – 40000, …, 100000+
- This is a categorical feature
  - At the same time, it represents a range binning on a numerical features
- Transformed into a value from 1 to 5

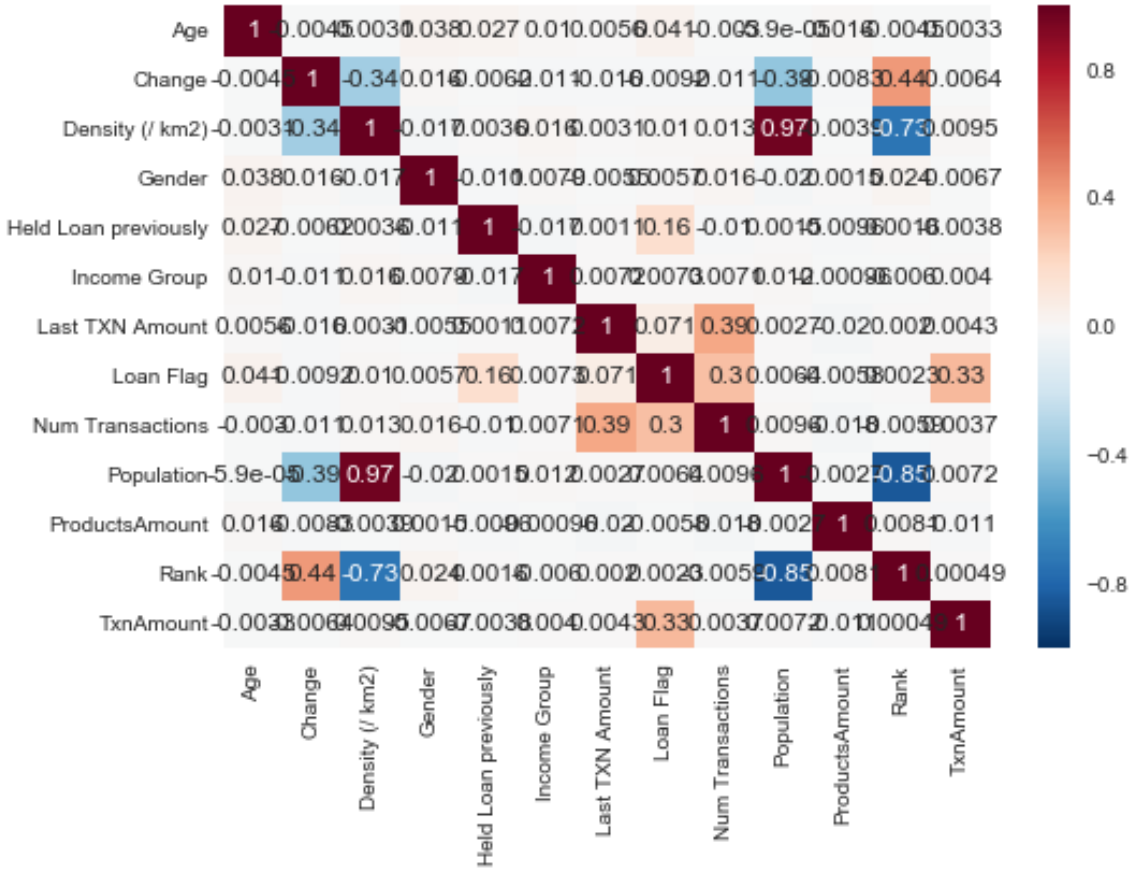
The same cannot be done for the City attribute

- If we transform it into integers, we do not want that our algorithm thinks that they could be ordered
- Hot Encoding: create a new (boolean) feature for each possible value

Feature Selection

Measure the correlation and discard the most correlated features.

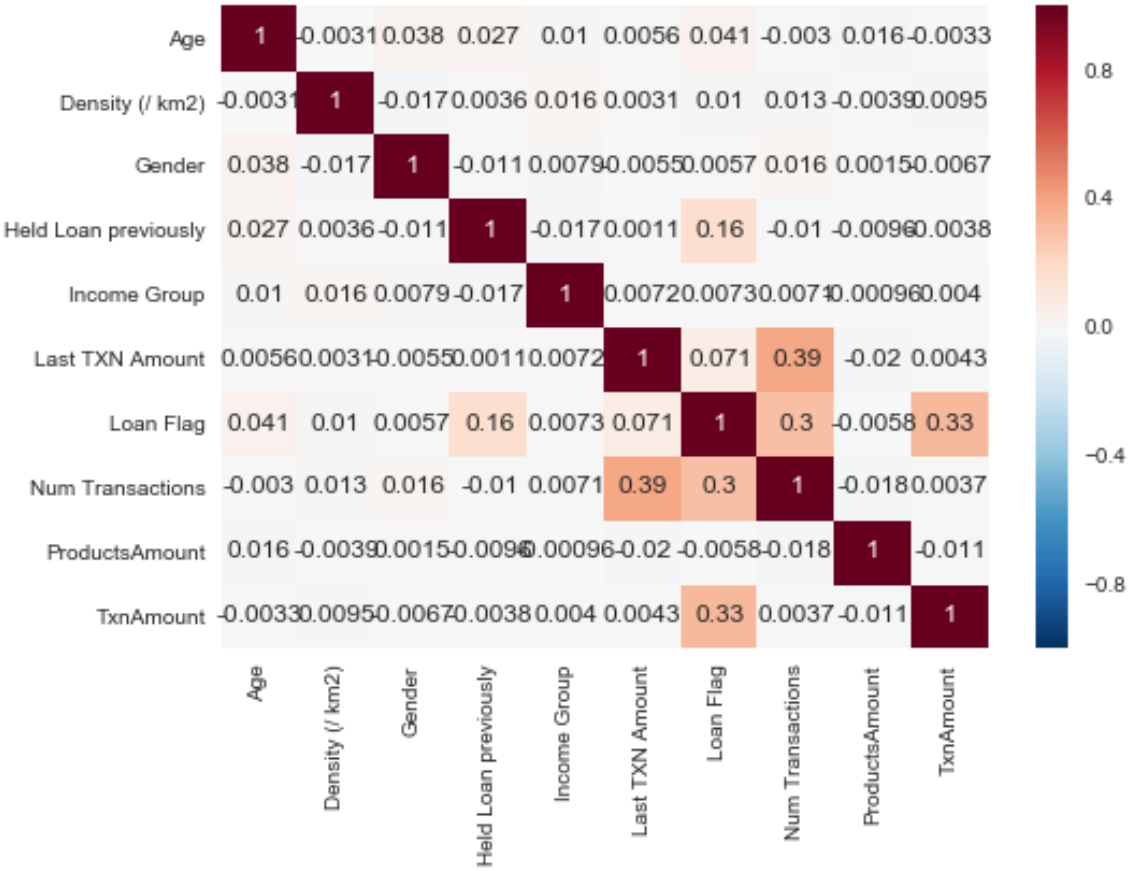
Some of them are very correlated: the ones related to the counties. Delete everything and keep just the most important for our target feature



Feature Selection

Measure the correlation and discard the most correlated features.

Some of them are very correlated: the ones related to the counties. Delete everything and keep just the most important for our target feature



Model Selection

Through Cross-validation, we have a look to the behavior of some of the classic approaches

- I usually give a try to Logistic Regression and Decision Trees or Random Forest.
- In this case, the records are not so many and the class to predict is a binary label
  - Quite high expectations for Linear SVM

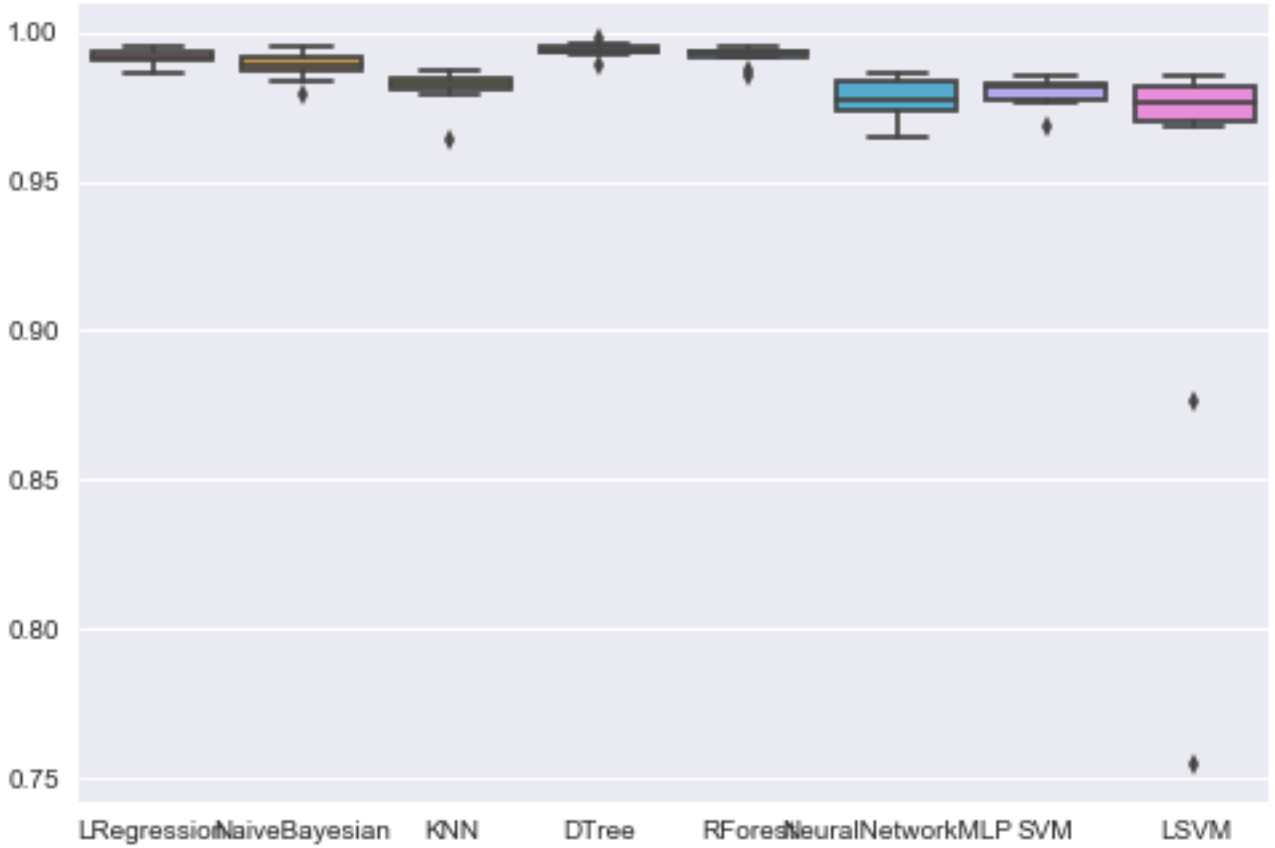
Define a Traning and Test set (80% - 20% of the initial dataset)

Model Selection

Measure the accuracy

They are all very accurate!

Wow, should we be satisfied?



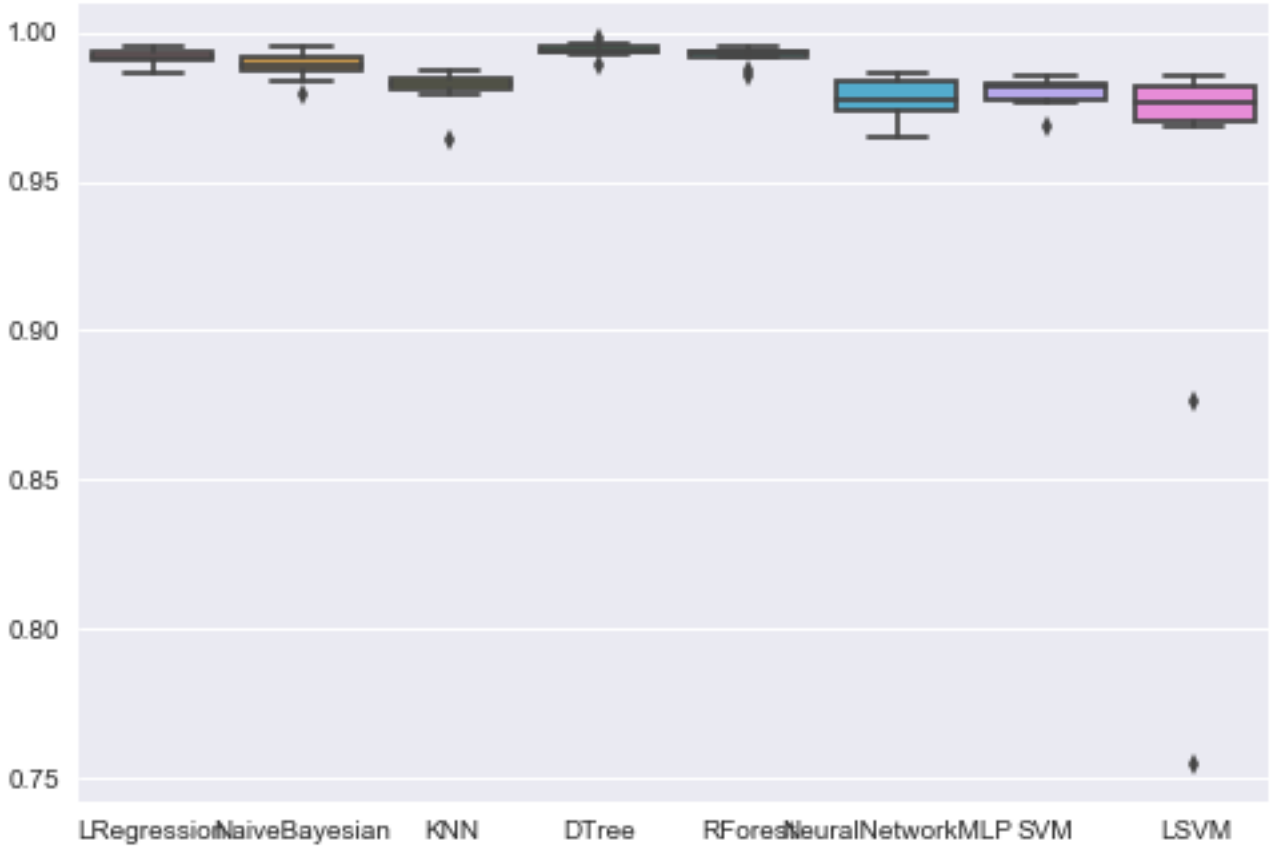
Model Selection

Measure the accuracy

They are all very accurate!

Wow, should we be satisfied?

No!

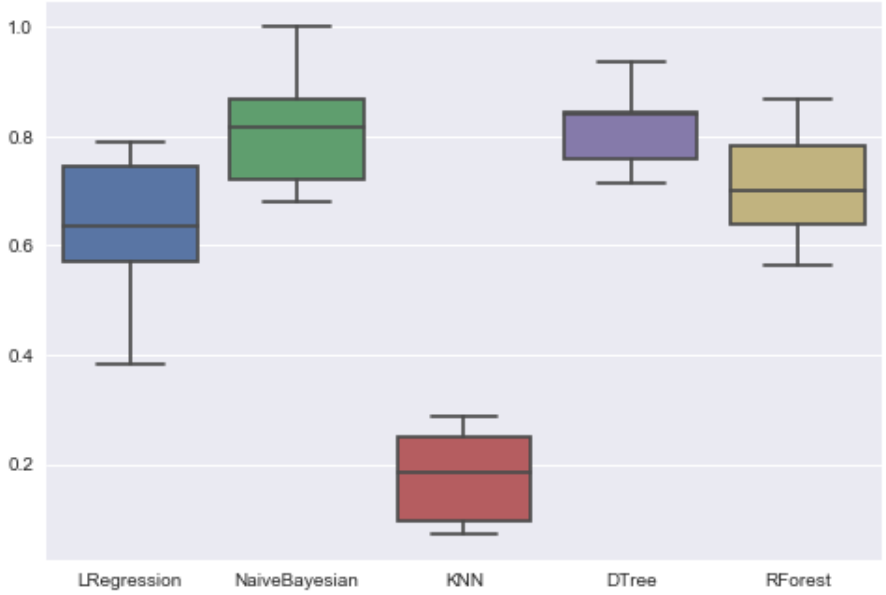


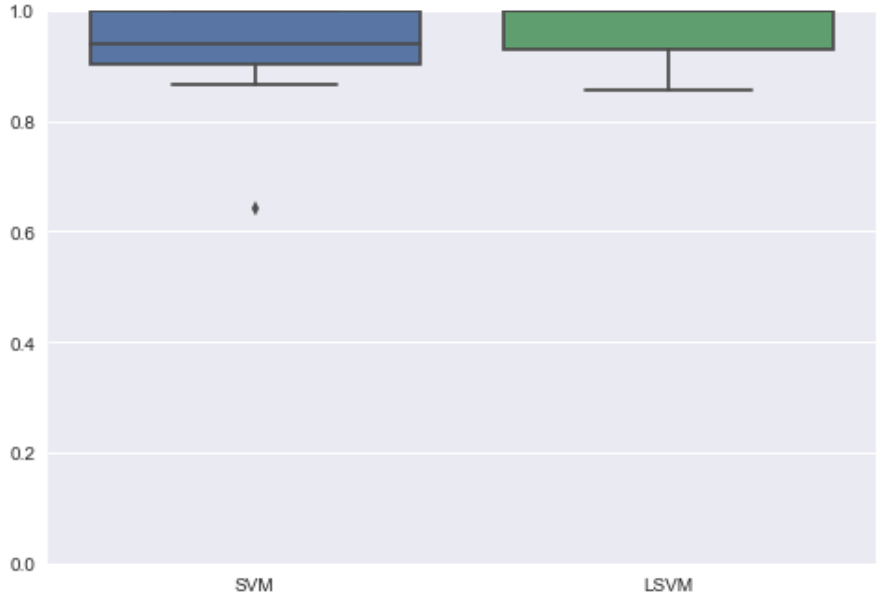
What's going on here?

- Some of the models assume a normal data distribution
- **The very high performances are biased by the dataset class distribution**
  - Only few people have taken up a loan
  - If the model predicts always «NO», it reaches almost 98% of accuracy
  - Accuracy could be easily overwhelmed by the great number of 0s.
    - Recall (True Positives / True Positives + False Negatives)

Model Selection

- Recall: clearer now!
- Very good performance by SVMs





However, our classes are still very unbalanced and no algorithm suits this kind of use cases

Undersampling & Oversampling

- There are just 210 samples with a Loan.
  - 2% of all the collections
- Resampling, we rebalance the datasets
- UnderSampling : we select just 210 samples without a Loan
- OverSampling : we create synthetic records to rebalance the datasets (SMOTE)
- In our decision, we really want to pay more attention to the TRUE POSITIVES
  - It is more important to get all the customers willing to take up a loan
  - We take the risk to assign too many positive labels
    - Even if we double them, they would be still a small amount

Model Selection

- In the end, we selected Logistic Regression, Random Forest, Decision Tree, Naïve Bayesian and Linear SVM to be tested against the testing set
- For these, we have tried the train with the full training set and the resampled ones (Under and Over sampled)
- In the end we selected Random Forest
  - Logistic Regression was close
  - Linear SVM gave too many False Negatives with the testing set

Random Forest - Tuning

- Random Forest seems the most promising overall
- As already mentioned, we could give the highest priority to the recall score of the 1 class
- Tuning process
  - Number of estimators (150)
  - Min Sample leaf (1)
  - Max Features (7)

	Precision	Recall
Taken up a loan	0.33	0.96

Alternatively, we could just maximize the hits but I really think that being precise about who is going to take up a loan is the priority

- Evaluating a model just for the number of hits is very naïve and not related to the use case

What we could have done with more time?

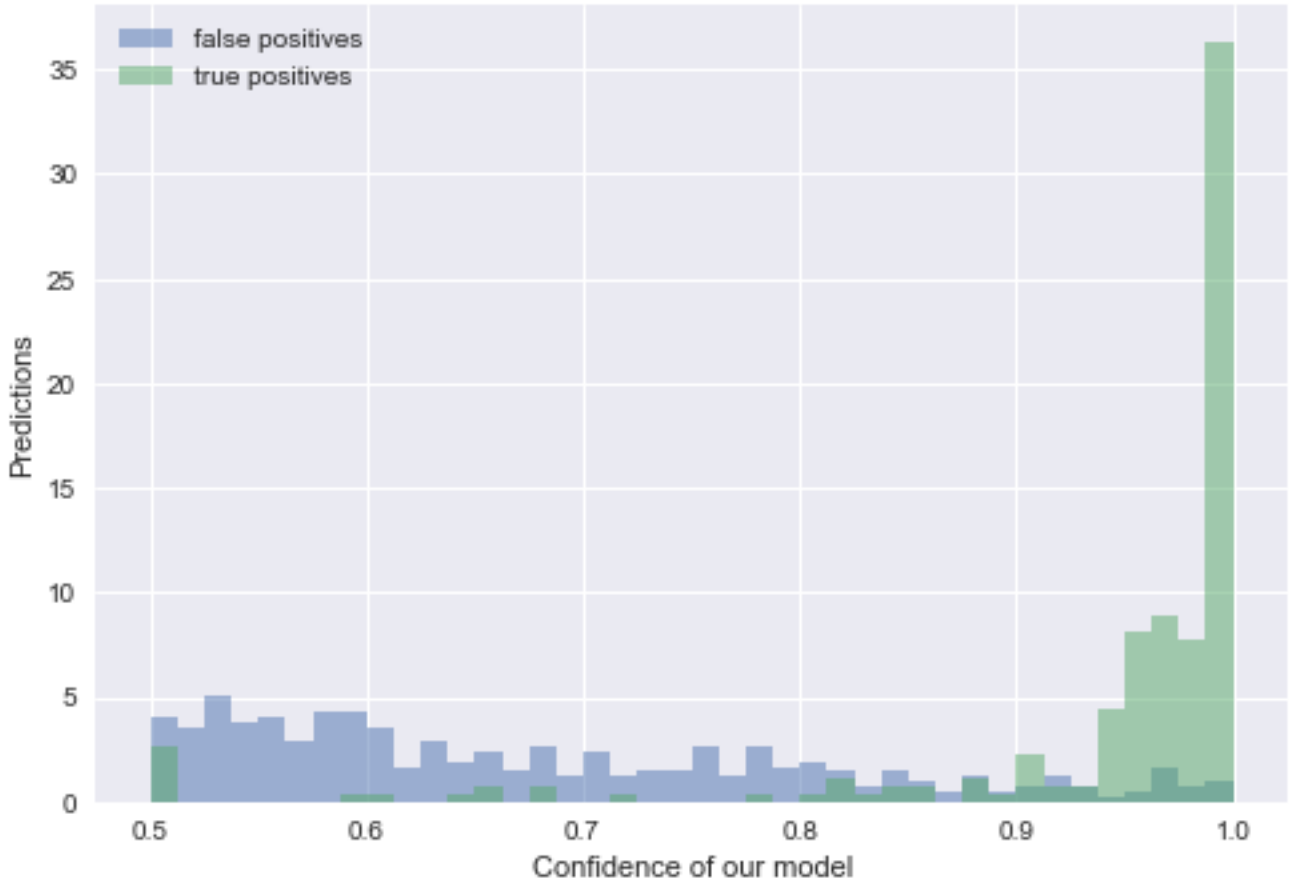
- Explore other categorical features like the Province
  - We have only used the City information and we decided to remove it
- Try more possibile parameter configurations for the other (not so promising) techniques
  - Still think that SVM could really fit this use case
- Explore different confidence thresholds of our final classifier to reduce the number of False positives

Likelihood Groups

- We should also divide the individuals of the test set in 5 categories, depending on the loan uptake rate
  - Very High Likelihood
  - High Likelihood
  - Medium Likelihood
  - Low Likelihood
  - Very Low Likelihood
- We used the confidence of our model
  - If > 50 %, the classifier predicts 1

Model Confidence

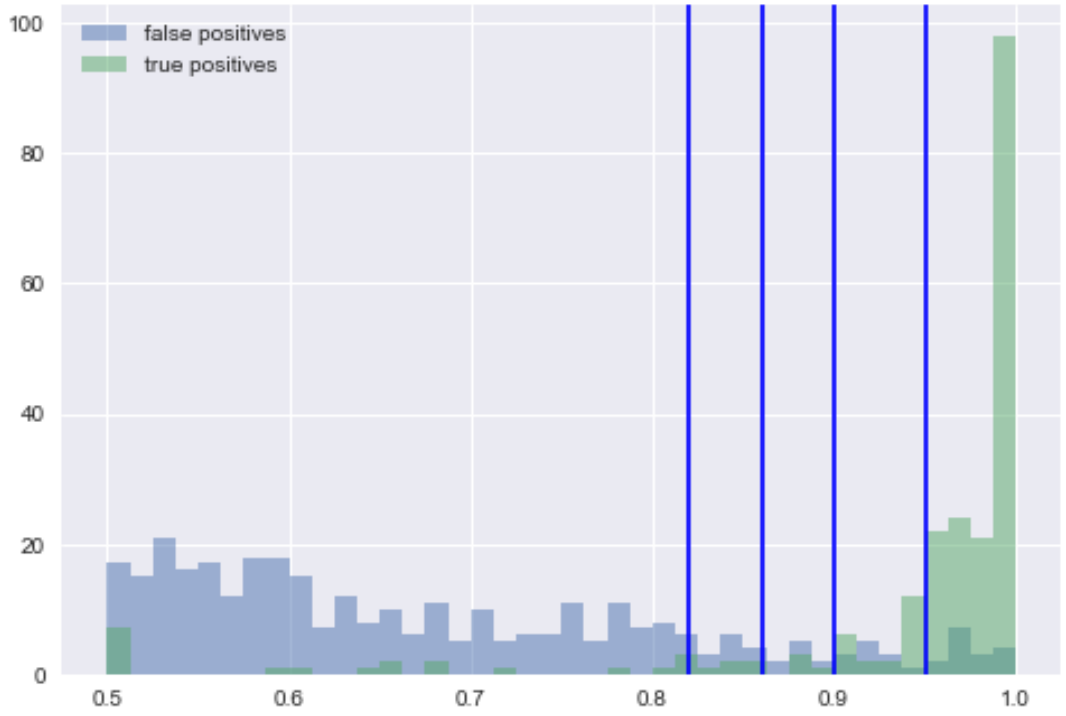
True Positives and False Positives distribution with respect to the confidence of the prediction



Likelihood Groups

We tried to select confidence thresholds to divide users in groups characterized by a certain likelihood to get a loan

- Very High Likelihood – 90%
- High Likelihood – 70%
- Medium Likelihood – 50%
- Low Likelihood – 30%
- Very Low Likelihood <10%



Likelihood Groups

- Groups size: relying on the confidence, our group size is very related to the data.
- However, the groups with a lower uptake rate are surely more numerous than the users more likely to take a loan
- Business relevance:
  - It really depends on what we want to do with these insights
  - For instance, Airlines companies understand who is going to book a ticket
    - When they are sure, they rise the prices
  - Probably, you cannot have this behavior with everybody but only with the (few) people of the top likelihood classes
    - Avoid customers class actions etc. ?

Thank you