Predicting the Success of Term Deposit Marketing Activities

Offline Task for Data Scientists - Check24

Andrea Di Simone
University of Freiburg (DE)

Outline

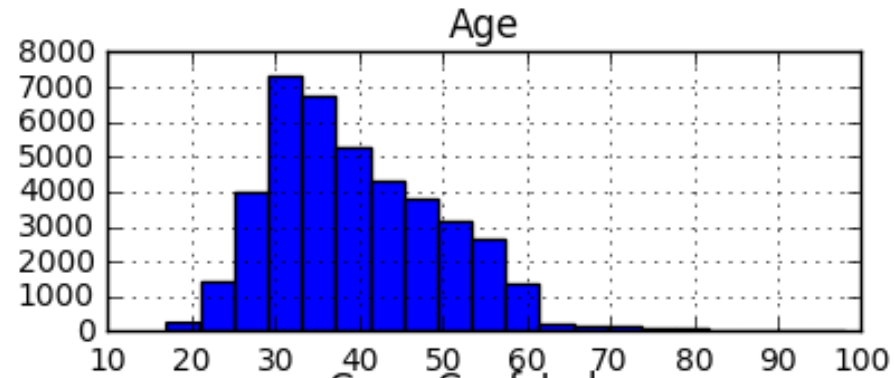
- Inspecting the inputs
  - Campaign success
  - Feature distributions
  - Feature correlations
- Fitting predictive models
  - Data preparation
  - Results
- Dealing with “Duration”

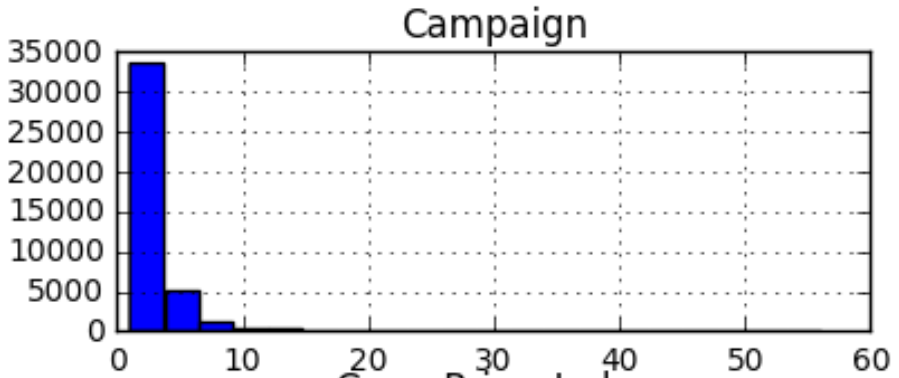
Feature inspection

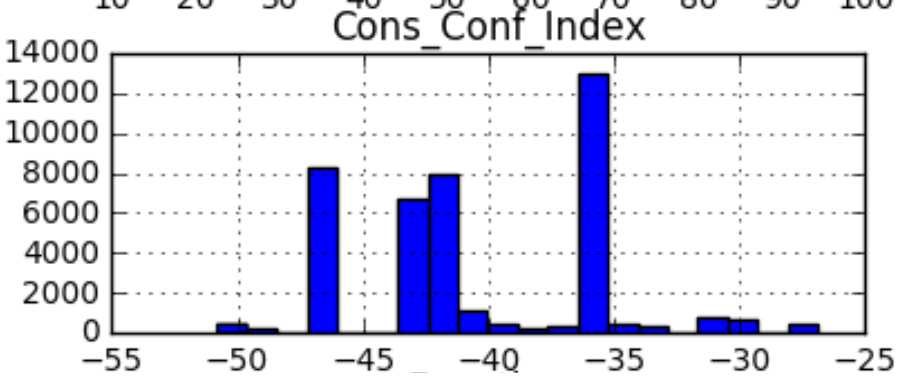
- The dataset appears to be extremely clean
  - No NaN values
  - Many categorical variables are represented by strings
    - These are, however, all regularly formatted, so that no further pre-processing of the strings is needed
- The fraction of success of the campaign is 4640/36548 = **12.7%**

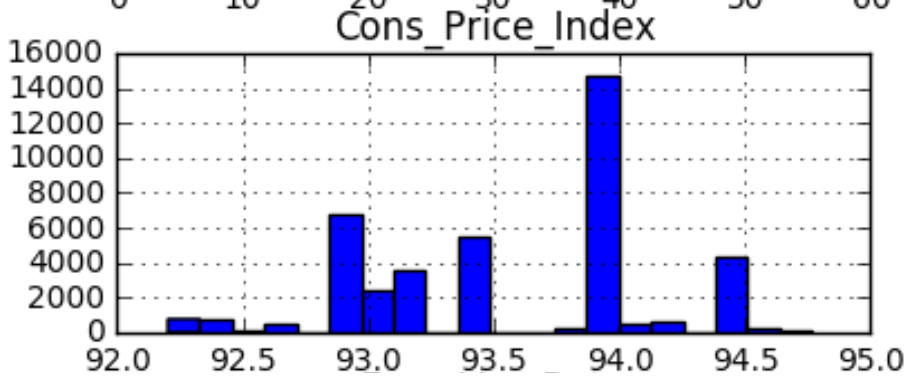
```
RangeIndex: 41188 entries, 0 to 41187
Data columns (total 21 columns):
Age                41188 non-null int64
Job                41188 non-null object
Marital            41188 non-null object
Education          41188 non-null object
Default            41188 non-null object
Housing            41188 non-null object
Loan               41188 non-null object
Contact            41188 non-null object
Month              41188 non-null object
Day_Of_Week        41188 non-null object
Duration           41188 non-null int64
Campaign           41188 non-null int64
Passed_Days        41188 non-null int64
Previous           41188 non-null int64
Previous_Outcome   41188 non-null object
Emp_Var_Rate       41188 non-null float64
Cons_Price_Index   41188 non-null float64
Cons_Conf_Index    41188 non-null float64
Euribor3m          41188 non-null float64
Nr_Employed        41188 non-null float64
Subscription       41188 non-null object
dtypes: float64(5), int64(5), object(11)
```

Numerical Features (1)

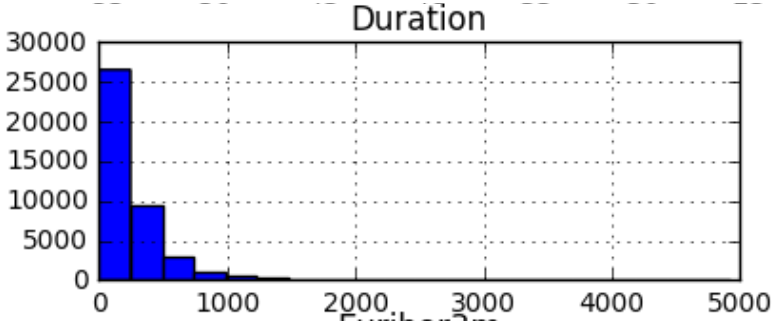


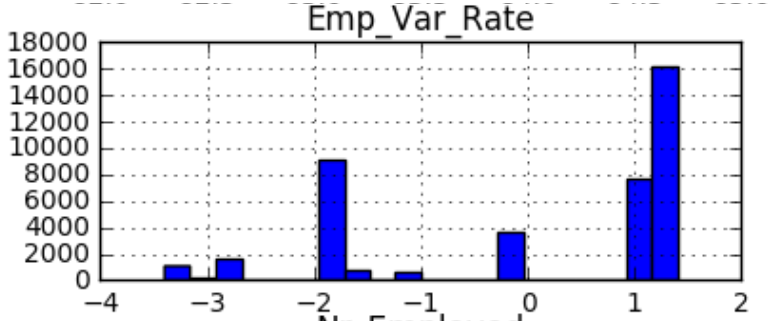


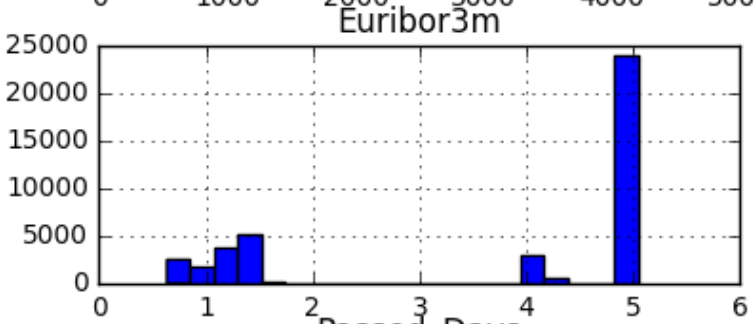


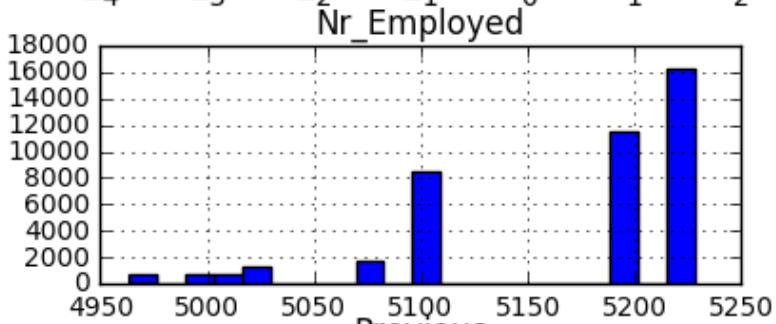


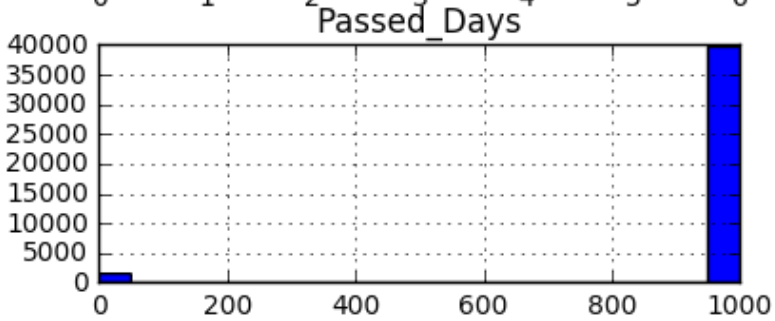
Numerical Features (2)

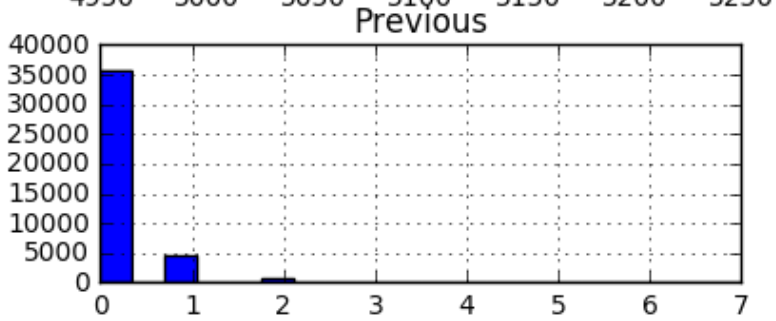




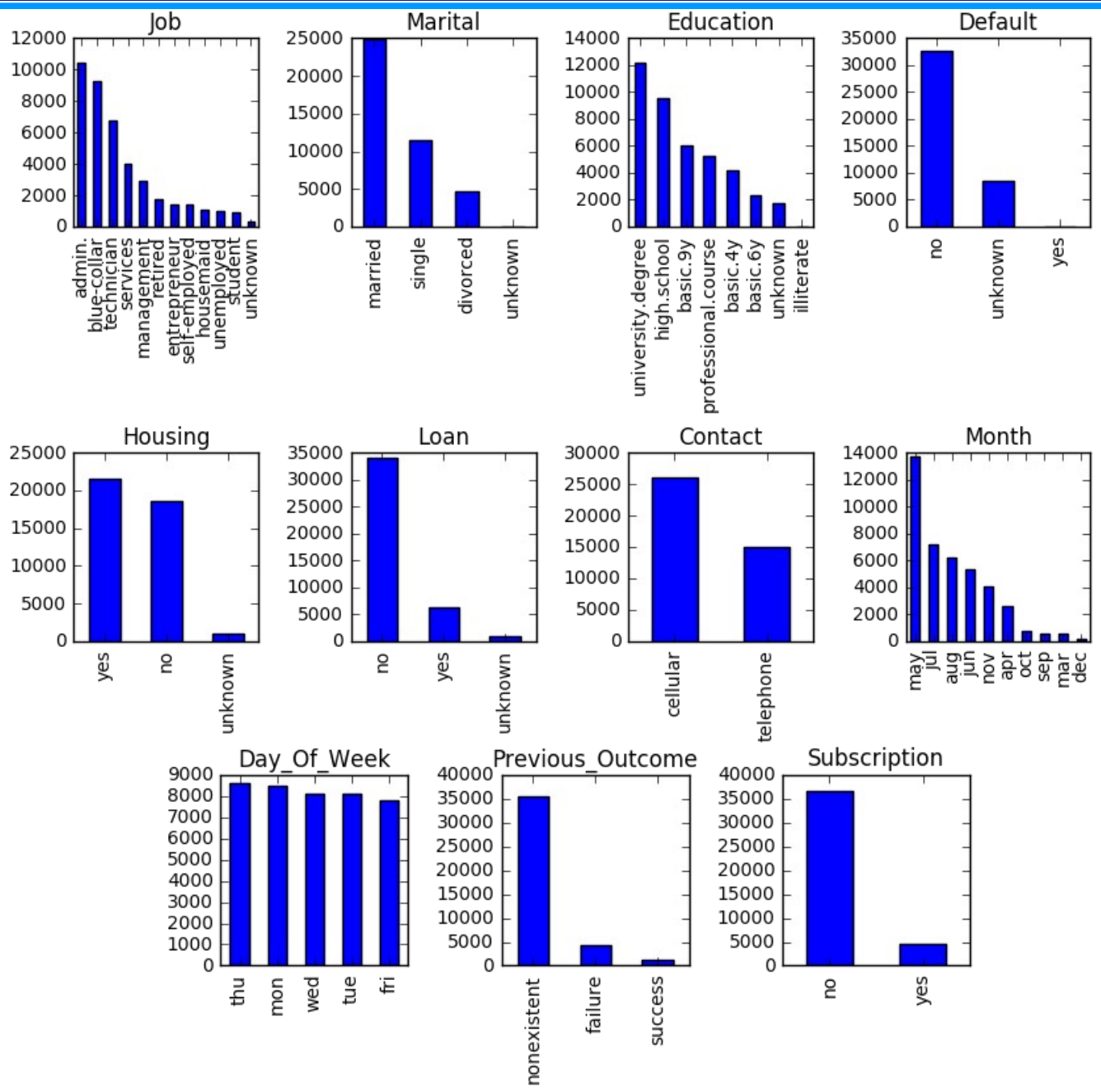




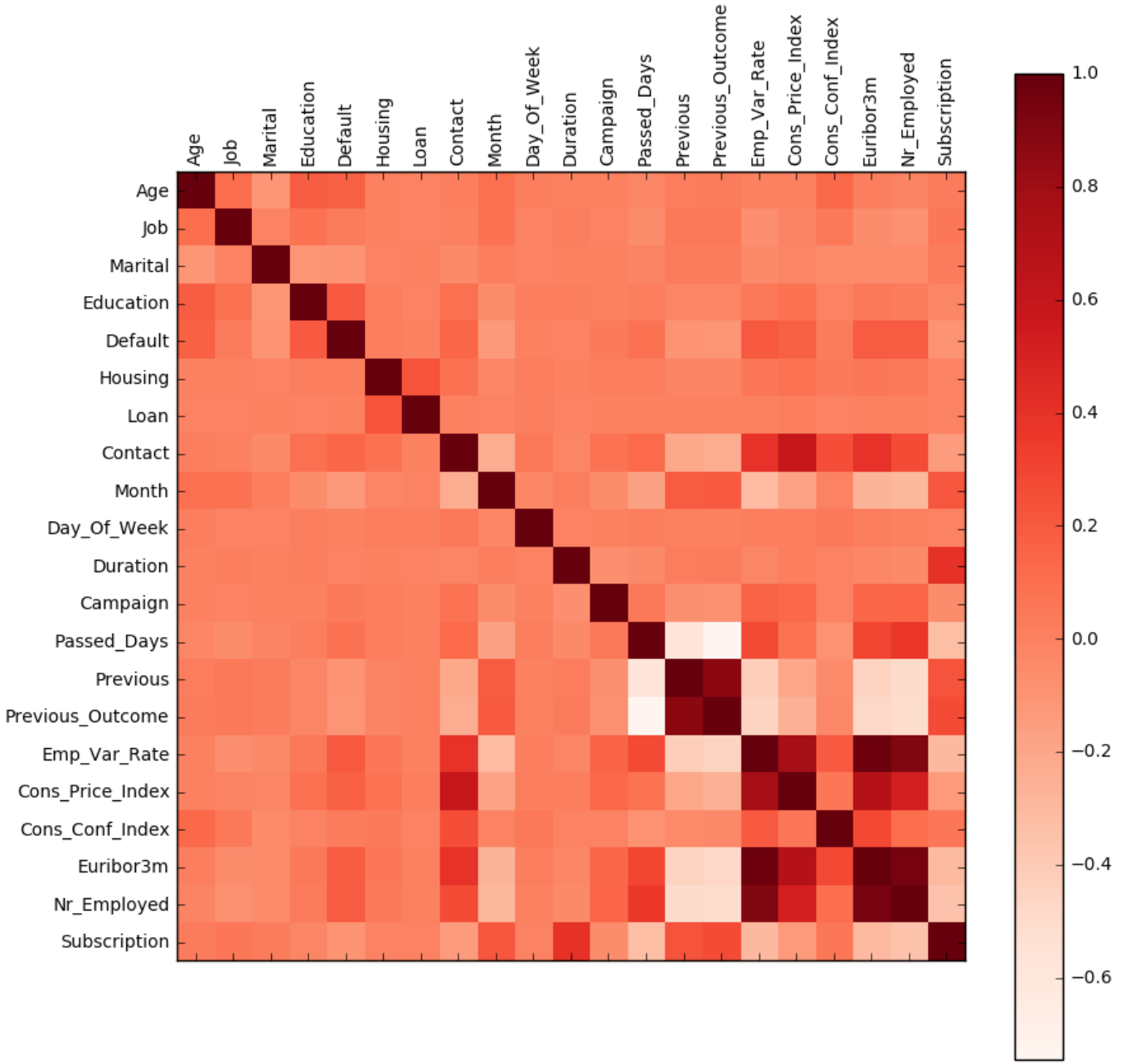




Categorical Features



Feature Correlation (1)



Feature Correlation (2)

- **Subscription** is moderately anti-correlated with **Passed_Days**.
  - Even taking into account that Passed_Days has many "999" values, this seems to suggest that ***subscription is more likely after repeated contacts***
- **Subscription** is, on the other hand, positively correlated with **Duration**, which suggests a ***preference for longer calls*****.**
- **Subscription** is also positively correlated with **Previous** and **Previous_Outcome** (which are, on the other hand, strongly correlated with each other)
  - This suggests that ***previous subscribers are more likely to subscribe again***.
- **Subscription** is anti-correlated with **Cons_Price_Index** and positively correlated with **Cons_Conf_index** which suggests ***more inclination towards subscription in relatively stable contexts***, as far as consumers are concerned.
  - The same argument holds for the anti-correlation with **Emp_Var_Rate**.
- The anti-correlation of **Subscription** and **Euribor3m** probably indicates that consumers are ***more likely to subscribe to long-term investments plans when the yield of short-term deposits is low***.
- I am honestly not sure I understand the meaning and the values of Nr_Employed, and I would therefore refrain from commenting on its link to the subscription probability

Predictive models

- The aim is to predict, given measurements for the features described in the previous slides, whether a phone call will result in a subscription or not
  - Conceptually, this is a (binary) classification problem
- Such models **can be used to predict how likely a set of potential clients would subscribe the plan**, and thus concentrate the campaign on those
- Due do the time constraints of this test, only two models are trained, with arelatively simple pre-processing of the features
  - All numerical features are scaled
  - All categorical features are transformed in One-Hot variables
- The models are fitted using cross validation, and the a minimal grid of possible values for hyper-parameters is tested
  - The optimal parameters are those that yield the best score on the test data
  - In this context, “score” is the accuracy of the model, i.e. the fraction of correct classifications

Models result

- A simple LogisticRegression model yields a score of 90.9%
  - Only the regularization parameter was optimized
- Using a RandomForestClassifier, only a modest improvement is obtained, with a score of 91.3%
  - Only the number of trees and the minimum number of entries required to split a leaf were optimized

Dealing with "Duration"

- I chose to leave "Duration" as a feature during model fitting and evaluation
  - Clearly this cannot be exploited on new calls, for which the duration is not known a priori
- For new calls, a possible strategy could be to assign, randomly, an *a-priori* duration, based on the distribution observed in the past
  - This works as long as Duration is not strongly correlated with any of the other features, which seems to be the case, based on the correlation plot shown in the previous slides
- A somewhat more sophisticated approach could be to train a model to predict the duration
  - This requires, however, better features (i.e. more discriminant as far as the duration is concerned), since the ones included in this dataset don't appear to be particularly relevant
- The only 100% correct approach is, of course, to leave out Duration from the model fitting
  - This would result, however, in loss of predictive power, since Duration is correlated with the probability of subscription

Possible improvements

- The time constraints imposed on this test don't allow to perform a more detailed analysis
- With more time, there are a few things that I would have tried
  - **Check correlations after fit:** Simpler models (like LogisticRegression) allow easy inspection of the fitted parameters/coefficients. Due to the intermediate One-Hot encoding their interpretation is now non-trivial, but given more time it is always fruitful to check whether the fit discovered anything unexpected
  - **Feature engineering**: it would be interesting to add some interaction terms between the categorical variables, i.e. building new features like Education*Marital. When done after the One-Hot encoding, this could help in exploiting some of the correlations between categories
  - **Better feature selection**: Additional steps can be added to the fitting chain, in order to reduce the number of features being fitted. At the very least, I would have tried a model-based selection and principal-components-analysis