ECON 1612

Big Data, Machine Learning & Society

Semester A

SGS Campus – Group 01

Dr. Chung Phan

Tran Huong Ly

S4000175

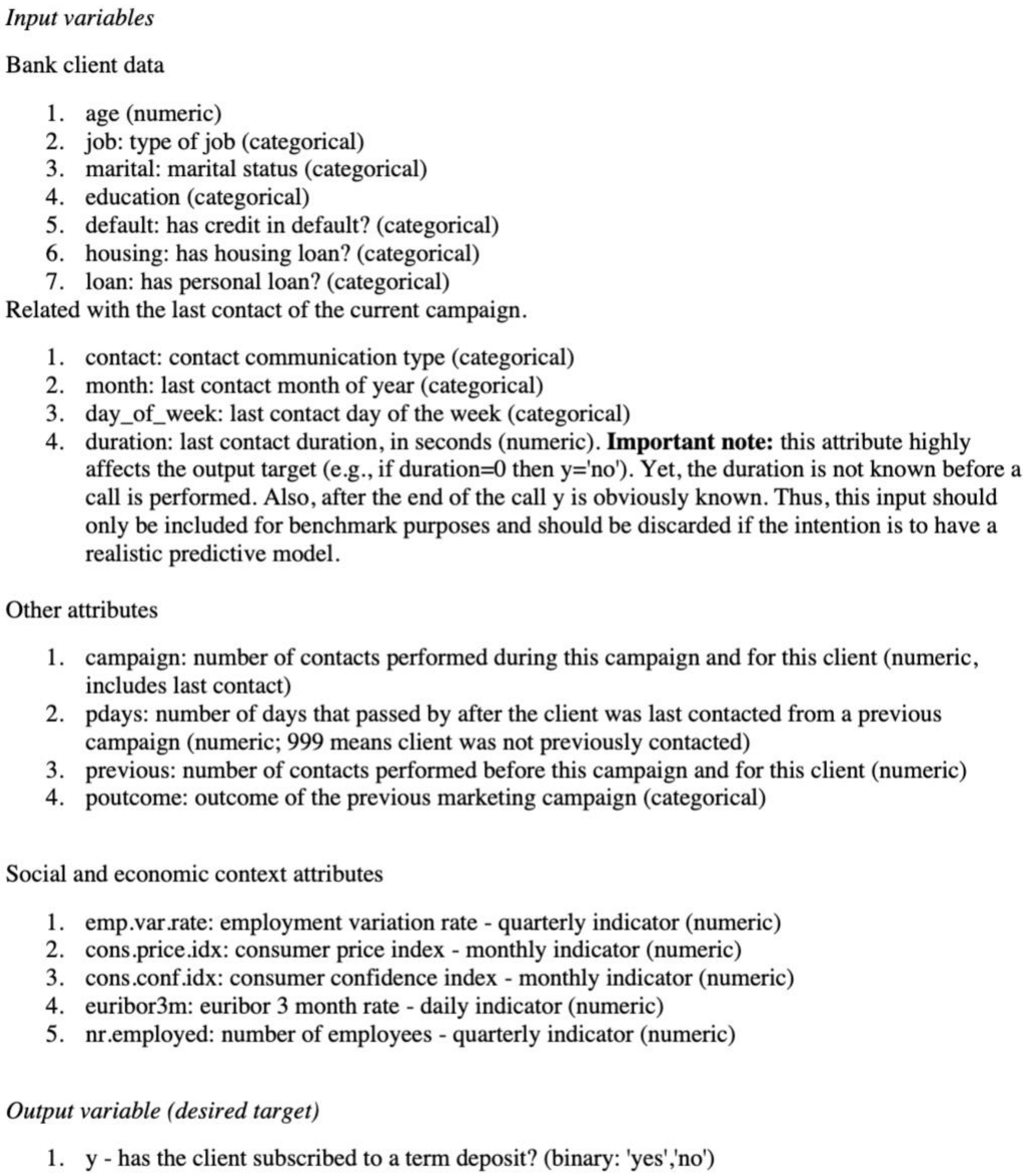
Assignment 3

Word count: 1563 words

(excluding table of contents & references)

Table of Contents

Variable Name Interpretation



Input variables

Bank client data

1. age (numeric)
2. job: type of job (categorical)
3. marital: marital status (categorical)
4. education (categorical)
5. default: has credit in default? (categorical)
6. housing: has housing loan? (categorical)
7. loan: has personal loan? (categorical)

Related with the last contact of the current campaign.

1. contact: contact communication type (categorical)
2. month: last contact month of year (categorical)
3. day_of_week: last contact day of the week (categorical)
4. duration: last contact duration, in seconds (numeric). **Important note:** this attribute highly affects the output target (e.g., if duration=0 then y='no'). Yet, the duration is not known before a call is performed. Also, after the end of the call y is obviously known. Thus, this input should only be included for benchmark purposes and should be discarded if the intention is to have a realistic predictive model.

Other attributes

1. campaign: number of contacts performed during this campaign and for this client (numeric, includes last contact)
2. pdays: number of days that passed by after the client was last contacted from a previous campaign (numeric; 999 means client was not previously contacted)
3. previous: number of contacts performed before this campaign and for this client (numeric)
4. poutcome: outcome of the previous marketing campaign (categorical)

Social and economic context attributes

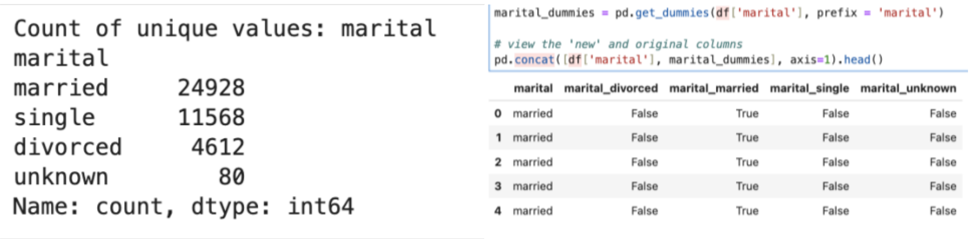
1. emp.var.rate: employment variation rate - quarterly indicator (numeric)
2. cons.price.idx: consumer price index - monthly indicator (numeric)
3. cons.conf.idx: consumer confidence index - monthly indicator (numeric)
4. euribor3m: euribor 3 month rate - daily indicator (numeric)
5. nr.employed: number of employees - quarterly indicator (numeric)

Output variable (desired target)

1. y - has the client subscribed to a term deposit? (binary: 'yes','no')

Introduction

The primary objective of this report is to create and evaluate a predictive machine learning model that accurately forecasts the likelihood of clients subscribing to term deposits following a marketing campaign for a Portuguese bank. To achieve this, data cleaning and transformation were performed using the sklearn library. Specifically, categorical features such as ‘marital’, ‘education’, ‘job’, and ‘poutcome’ were converted into boolean form (true or false) and binary variables with yes/no answers (‘is_housing’, ‘is_loan_’, ‘is_default’, ‘treated’) were converted to binary form (1 or 0). The variable ‘treated’ indicates whether a person was contacted during the campaign (converted from ‘p-days’). Additionally, unnecessary variables ('contact', 'month', 'day_of_week', 'duration', 'campaign') were removed for simplification purposes. The resulting dataset was then split for training purposes to eliminate known outcomes in preparation for training. This approach ensures that the prediction model is built on clean, organized data, enhancing its ability to predict client responses effectively (Phan 2024a).



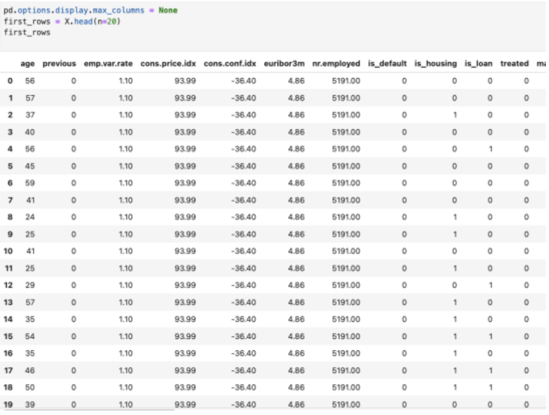
```
Count of unique values: marital
marital
married      24928
single       11568
divorced      4612
unknown         80
Name: count, dtype: int64
```

```
marital_dummies = pd.get_dummies(df['marital'], prefix = 'marital')

# view the 'new' and original columns
pd.concat([df['marital'], marital_dummies], axis=1).head()
```

	marital	marital_divorced	marital_married	marital_single	marital_unknown
0	married	False	True	False	False
1	married	False	True	False	False
2	married	False	True	False	False
3	married	False	True	False	False
4	married	False	True	False	False

Figure 1. Characteristics, execution & output of transforming variable ‘marital’ as one of the categorical variables

Data Summarisation & Description



```
pd.options.display.max_columns = None
first_rows = X.head(n=20)
first_rows
```

	age	previous	emp.var.rate	cons.price.idx	cons.conf.idx	euribor3m	nr.employed	is_default	is_housing	is_loan	treated	ma
0	56	0	1.10	93.99	-36.40	4.86	5191.00	0	0	0	0	
1	57	0	1.10	93.99	-36.40	4.86	5191.00	0	0	0	0	
2	37	0	1.10	93.99	-36.40	4.86	5191.00	0	1	0	0	
3	40	0	1.10	93.99	-36.40	4.86	5191.00	0	0	0	0	
4	56	0	1.10	93.99	-36.40	4.86	5191.00	0	0	1	0	
5	45	0	1.10	93.99	-36.40	4.86	5191.00	0	0	0	0	
6	59	0	1.10	93.99	-36.40	4.86	5191.00	0	0	0	0	
7	41	0	1.10	93.99	-36.40	4.86	5191.00	0	0	0	0	
8	24	0	1.10	93.99	-36.40	4.86	5191.00	0	1	0	0	
9	25	0	1.10	93.99	-36.40	4.86	5191.00	0	1	0	0	
10	41	0	1.10	93.99	-36.40	4.86	5191.00	0	0	0	0	
11	25	0	1.10	93.99	-36.40	4.86	5191.00	0	1	0	0	
12	29	0	1.10	93.99	-36.40	4.86	5191.00	0	0	1	0	
13	57	0	1.10	93.99	-36.40	4.86	5191.00	0	1	0	0	
14	35	0	1.10	93.99	-36.40	4.86	5191.00	0	1	0	0	
15	54	0	1.10	93.99	-36.40	4.86	5191.00	0	1	1	0	
16	35	0	1.10	93.99	-36.40	4.86	5191.00	0	1	0	0	
17	46	0	1.10	93.99	-36.40	4.86	5191.00	0	1	1	0	
18	50	0	1.10	93.99	-36.40	4.86	5191.00	0	1	1	0	
19	39	0	1.10	93.99	-36.40	4.86	5191.00	0	0	0	0	

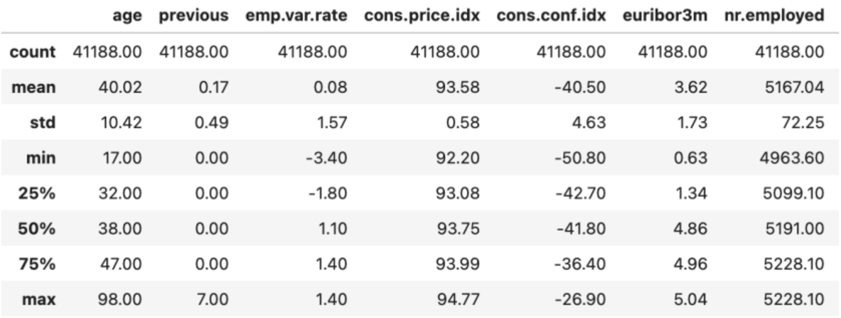
Figure 2. Snippet of the first 20 rows of clean data

After data cleaning and transformation, descriptive statistics of several key variables were examined to obtain early insights from the dataset:

Age: The mean age is 40.02, but the relatively high standard deviation suggests significant variability. This variability could potentially impact the accuracy of any subsequent prediction model. A similar level of unpredictability is observed with the ‘nr.employed’ variable.

Previous contacts: Remarkably, 75% of the dataset had not been contacted before the campaign. Consequently, this variable may not contribute significantly to the prediction model.

Employment variation & 3-month Euribor: Both ‘emp.var.rate’ and ‘euribor3m’ exhibit minimal standard deviation, indicating consistency. These variables could prove valuable for enhancing the dataset’s robustness.



	age	previous	emp.var.rate	cons.price.idx	cons.conf.idx	euribor3m	nr.employed
count	41188.00	41188.00	41188.00	41188.00	41188.00	41188.00	41188.00
mean	40.02	0.17	0.08	93.58	-40.50	3.62	5167.04
std	10.42	0.49	1.57	0.58	4.63	1.73	72.25
min	17.00	0.00	-3.40	92.20	-50.80	0.63	4963.60
25%	32.00	0.00	-1.80	93.08	-42.70	1.34	5099.10
50%	38.00	0.00	1.10	93.75	-41.80	4.86	5191.00
75%	47.00	0.00	1.40	93.99	-36.40	4.96	5228.10
max	98.00	7.00	1.40	94.77	-26.90	5.04	5228.10

Figure 3. Descriptive statistics of some variables

Categorical responses were also visualised and further analysed. Notably, responders were more likely to have graduated from university or to be married, while being less likely to be illiterate or divorced.

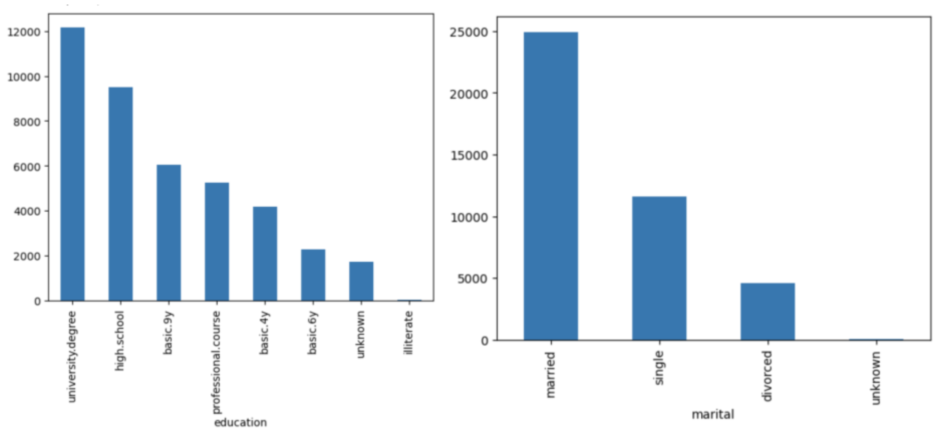


Figure 4. Number of observations in some response categories

Graphing

Understanding economic theory and context is crucial. The scatterplot of two variables “cons.price.idx” and “cons.conf.idx” below reveals a pattern where consumer confidence diminishes as the consumer price index (CPI) ascends, indicating a negative correlation between the two variables. This suggests a hypothesis that the success rate for subscriptions may decrease when CPI increases.

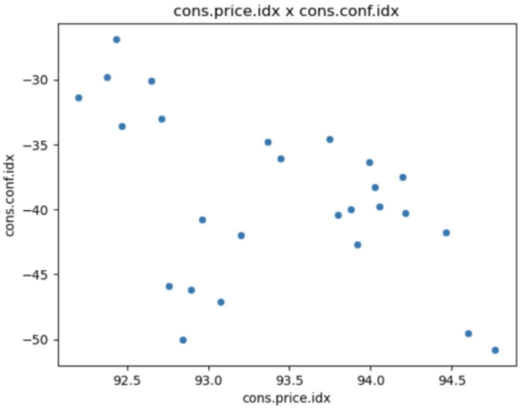


Figure 5. Scatter plot of variables “cons.price.idx” and “cons.conf.idx”

In addition to this, histograms have been instrumental in offering deeper insights during data analysis. A notable observation is the right-skewed distribution of the ‘age’ variable, pointing towards a younger sample demographic. Furthermore, categorical variables like ‘is_loan’ were scrutinized, unveiling that a significant 89% of the individuals in the dataset are encumbered with personal loans.

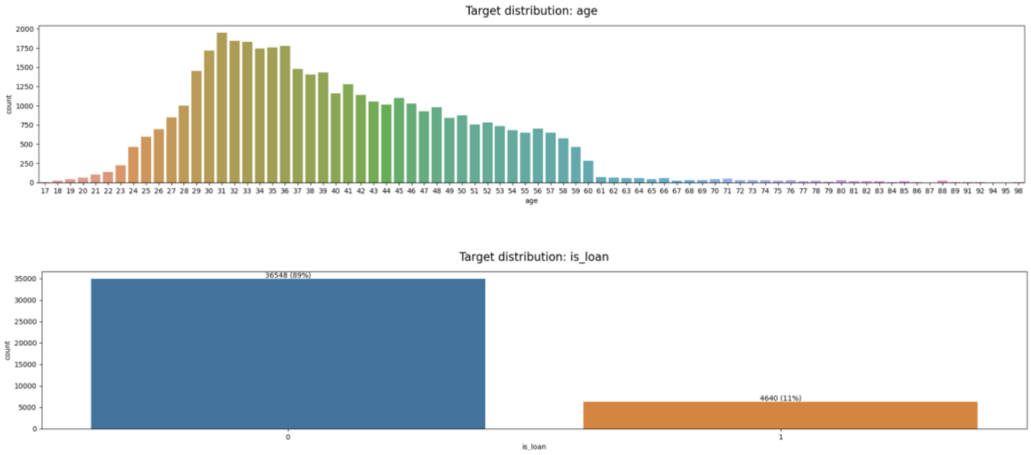
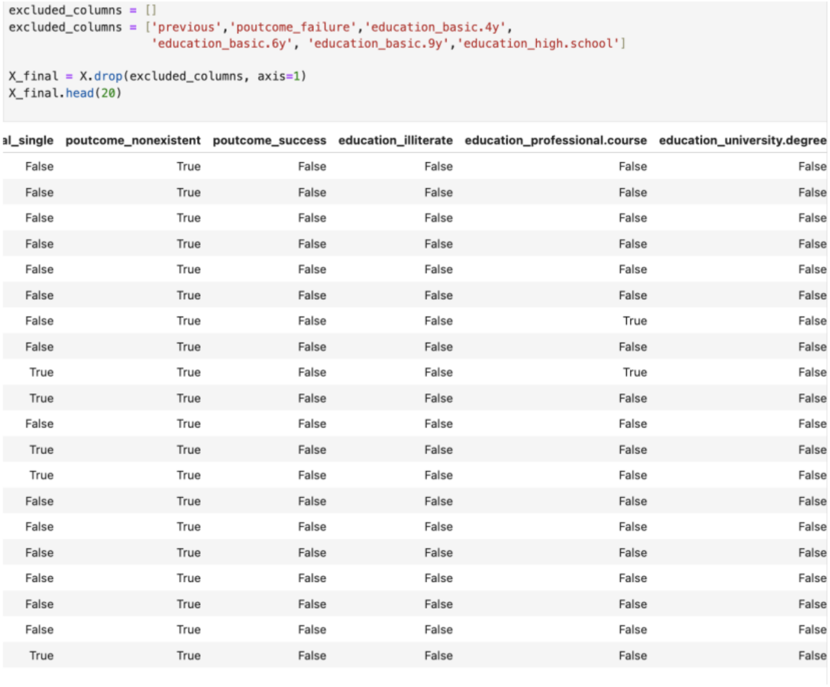


Figure 6. Histograms of variables ‘age’ and ‘is_loan’

Variable Exclusion

In predicting the success of a marketing campaign, several variables were deemed unnecessary, and their removal is essential to not overcomplicate the model (Phan 2024b). Education level significantly impacts outcomes, however basic education (4 years, 6 years, or 9 years) and illiteracy associated with lower chances of success. Additionally, previous campaign results influence current prospects; success rates may be diminished if the previous campaign failed. The number of contacts made prior to the campaign may not always correlate directly with success, especially if for reasons other than the campaign.



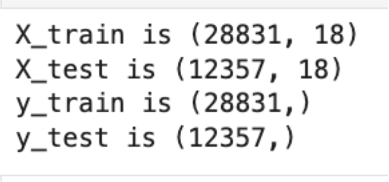
```
excluded_columns = []
excluded_columns = ['previous','poutcome_failure','education_basic.4y',
                    'education_basic.6y', 'education_basic.9y','education_high.school']

X_final = X.drop(excluded_columns, axis=1)
X_final.head(20)
```

al_single	poutcome_nonexistent	poutcome_success	education_illiterate	education_professional.course	education_university.degree
False	True	False	False	False	False
False	True	False	False	False	False
False	True	False	False	False	False
False	True	False	False	False	False
False	True	False	False	False	False
False	True	False	False	False	False
False	True	False	False	True	False
False	True	False	False	False	False
True	True	False	False	True	False
True	True	False	False	False	False
False	True	False	False	False	False
True	True	False	False	False	False
True	True	False	False	False	False
False	True	False	False	False	False
False	True	False	False	False	False
False	True	False	False	False	False
False	True	False	False	False	False
False	True	False	False	False	False
False	True	False	False	False	False
True	True	False	False	False	False

Figure 7. Dataset snippet after variable exclusion

Classification Tree Construction (Unpruned)

After exclusion of more variables, there were 18 variables left in total. The data was split into training and test sets, with X sets not including the known outcomes. The test size is set to 0.3, meaning 30% of the data will be used for testing and 70% for training.



```
X_train is (28831, 18)
X_test is (12357, 18)
y_train is (28831,)
y_test is (12357,)
```

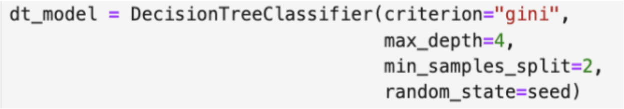
Figure 8. Dimensions of the training & testing sets

A decision tree was then constructed with DecisionTreeClassifier, with the following hyperparameters set:

`criterion="gini"`: This hyperparameter specifies the function to measure the quality of a split. "gini" refers to the Gini impurity, which is a measure of how often a randomly chosen element would be incorrectly classified.

`max_depth=5`: This hyperparameter determines the maximum depth of the decision tree. It restricts the depth of the tree to 5 levels, which can help control overfitting and simplify the model.

`min_samples_split=2`: This hyperparameter sets the minimum number of samples required to split an internal node. In this case, a node will only be split if it contains at least 2 samples. This helps prevent the tree from making overly specific decisions based on a small number of samples.



```
dt_model = DecisionTreeClassifier(criterion="gini",
                                  max_depth=4,
                                  min_samples_split=2,
                                  random_state=seed)
```

Figure 9. Tree setup

Unpruned Tree Interpretation

Squared errors represent the discrepancy between predicted outcomes and actual observations. A higher squared error indicates greater inaccuracy, as depicted by the orange/red regions in the accompanying diagram. Conversely, lower squared errors correspond to more accurate predictions, represented by the blue areas. In this model, the scarcity of blue regions suggests significant inaccuracy, with only one outcome achieving exact agreement (squared errors =0.0).

Consequently, this model is unsuitable for accurately predicting the success rate of term deposit subscriptions.

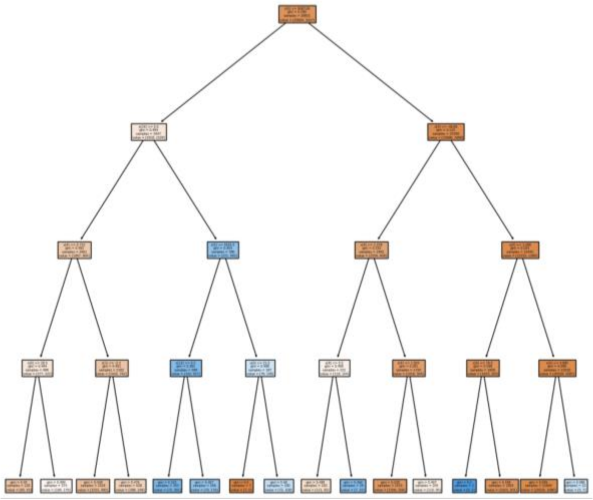
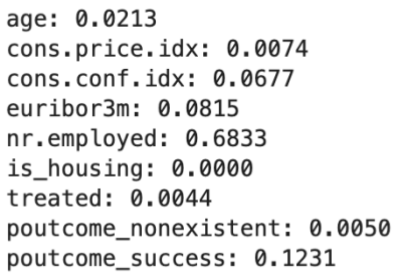


Figure 10. Tree output without pruning

Feature Importance



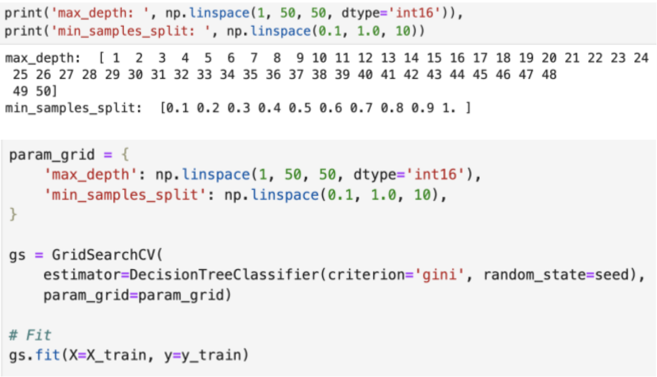
```
age: 0.0213
cons.price.idx: 0.0074
cons.conf.idx: 0.0677
euribor3m: 0.0815
nr.employed: 0.6833
is_housing: 0.0000
treated: 0.0044
poutcome_nonexistent: 0.0050
poutcome_success: 0.1231
```

Figure 11. Feature importance of all important features (importance ≠ 0.0000)

First, the employment rate is deemed of great importance, as it may influence a person’s disposable income and, consequently, their decision to open a term deposit. Second, success rate of previous campaigns also heavily influences future success rate, as there may be a high likelihood someone will re-enter due to this campaign if they have previously entered a term deposit. Notably, amongst the most important features are mostly macroeconomic indicators, including the CPI, CCI, 3-month Euribor rate, and employment rate. However, a potential issue lies in the decision tree’s depth, which may lead to excessive variable importance due to overfitting.

Optimal Tree Using GridsearchCV

GridSearchCV is a method used for hyperparameter tuning that exhaustively searches through a specified parameter grid to determine the optimal combination of hyperparameters for a given machine learning model. Arrays of parameters were defined and fed into GridsearchCV as follows:



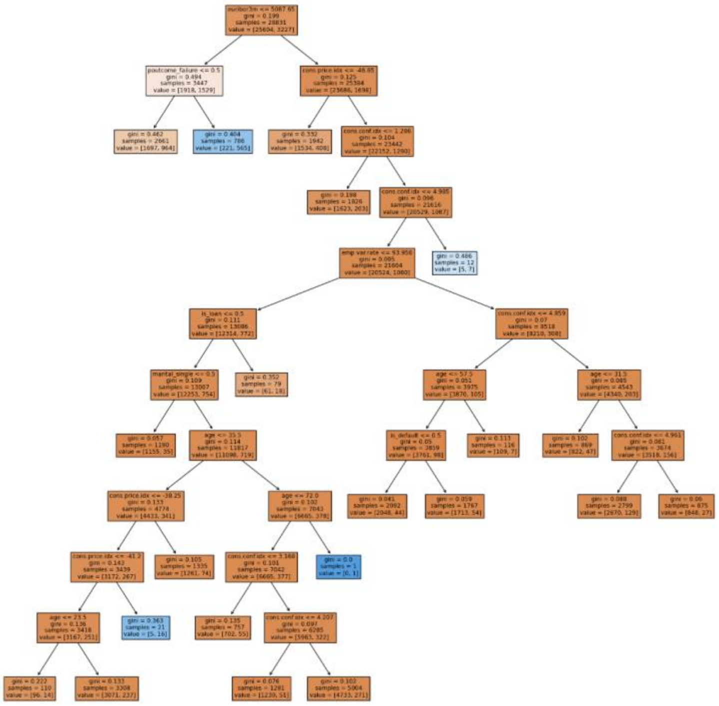
```
print('max_depth: ', np.linspace(1, 50, 50, dtype='int16')),
print('min_samples_split: ', np.linspace(0.1, 1.0, 10))
```

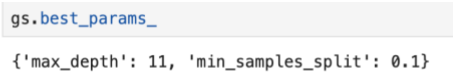
```
max_depth:  [ 1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24
 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48
 49 50]
min_samples_split:  [0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1. ]
```

```
param_grid = {
    'max_depth': np.linspace(1, 50, 50, dtype='int16'),
    'min_samples_split': np.linspace(0.1, 1.0, 10),
}

gs = GridSearchCV(
    estimator=DecisionTreeClassifier(criterion='gini', random_state=seed),
    param_grid=param_grid)

# Fit
gs.fit(X=X_train, y=y_train)
```

Figure 12. Hyperparameter definition and tuning

The assist of GridsearchCV helps in finding the optimal combination of parameter by trial of all combinations and thus constructing the most optimal tree as the following:


euribor3m <= 5087.65 gini = 0.199 samples = 28831 value = [25604, 3227]
poutcome_failure <= 0.5 gini = 0.494 samples = 3447 value = [1918, 1529]
cons.price.idx <= -48.65 gini = 0.125 samples = 25384 value = [23686, 1698]
gini = 0.462 samples = 2661 value = [1697, 964]
gini = 0.404 samples = 786 value = [221, 565]
gini = 0.332 samples = 1942 value = [1534, 408]
cons.conf.idx <= 1.286 gini = 0.104 samples = 23442 value = [22152, 1290]
gini = 0.198 samples = 1826 value = [1623, 203]
cons.conf.idx <= 4.985 gini = 0.096 samples = 21616 value = [20529, 1087]
emp.var.rate <= 93.956 gini = 0.095 samples = 21604 value = [20524, 1080]
gini = 0.486 samples = 12 value = [5, 7]
is_loan <= 0.5 gini = 0.111 samples = 13086 value = [12314, 772]
cons.conf.idx <= 4.859 gini = 0.07 samples = 8518 value = [8210, 308]
marital_single <= 0.5 gini = 0.109 samples = 13007 value = [12253, 754]
gini = 0.352 samples = 79 value = [61, 18]
age <= 57.5 gini = 0.051 samples = 3975 value = [3870, 105]
age <= 31.5 gini = 0.085 samples = 4543 value = [4340, 203]
gini = 0.057 samples = 1190 value = [1155, 35]
age <= 35.5 gini = 0.114 samples = 11817 value = [11098, 719]
is_default <= 0.5 gini = 0.05 samples = 3859 value = [3761, 98]
gini = 0.113 samples = 116 value = [109, 7]
gini = 0.102 samples = 869 value = [822, 47]
cons.conf.idx <= 4.961 gini = 0.081 samples = 3674 value = [3518, 156]
cons.price.idx <= -38.25 gini = 0.133 samples = 4774 value = [4433, 341]
age <= 72.0 gini = 0.102 samples = 7043 value = [6665, 378]
gini = 0.041 samples = 2092 value = [2048, 44]
gini = 0.059 samples = 1767 value = [1713, 54]
gini = 0.088 samples = 2799 value = [2670, 129]
gini = 0.06 samples = 875 value = [848, 27]
cons.price.idx <= -41.2 gini = 0.143 samples = 3439 value = [3172, 267]
gini = 0.105 samples = 1335 value = [1261, 74]
cons.conf.idx <= 3.188 gini = 0.101 samples = 7042 value = [6665, 377]
gini = 0.0 samples = 1 value = [0, 1]
age <= 23.5 gini = 0.136 samples = 3418 value = [3167, 251]
gini = 0.363 samples = 21 value = [5, 16]
gini = 0.135 samples = 757 value = [702, 55]
cons.conf.idx <= 4.207 gini = 0.097 samples = 6285 value = [5963, 322]
gini = 0.222 samples = 110 value = [96, 14]
gini = 0.133 samples = 3308 value = [3071, 237]
gini = 0.076 samples = 1281 value = [1230, 51]
gini = 0.102 samples = 5004 value = [4733, 271]



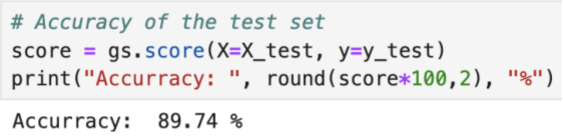
```
gs.best_params_
```

```
{'max_depth': 11, 'min_samples_split': 0.1}
```

Figure 13. The optimal tree and its parameters

Here, the maximum depth of the optimal tree is 11 and the minimum split of samples is defined as 0.1, meaning one node split must divide at least 10% the total amount of samples. This differs from the unpruned tree in which the depth is significantly larger while the split is significantly smaller, resulting in a more accurate prediction by reducing overfitting (Phan 2024c).

Compared to the unpruned tree, this tree has more outcomes that are more accurate compared to true outcomes. There are also less unambiguous outcomes (neither red nor blue), thus also indicating more accuracy. Additionally, the overall accuracy of this model, as calculated below, has surpassed the industry expectation (Nguyen 2024).

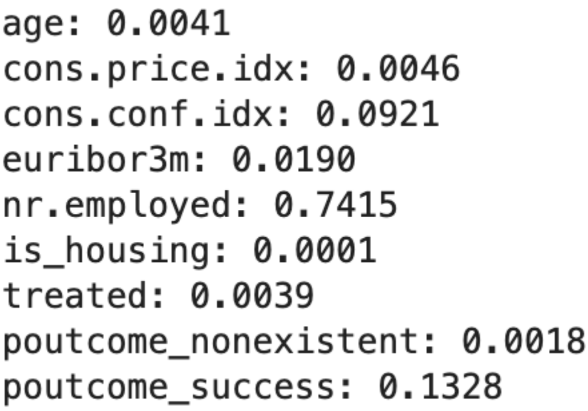


```
# Accuracy of the test set
score = gs.score(X=X_test, y=y_test)
print("Accurracy: ", round(score*100,2), "%")
```

```
Accurracy:  89.74 %
```

Figure 14. Accuracy test

Feature Importance After GridsearchCV



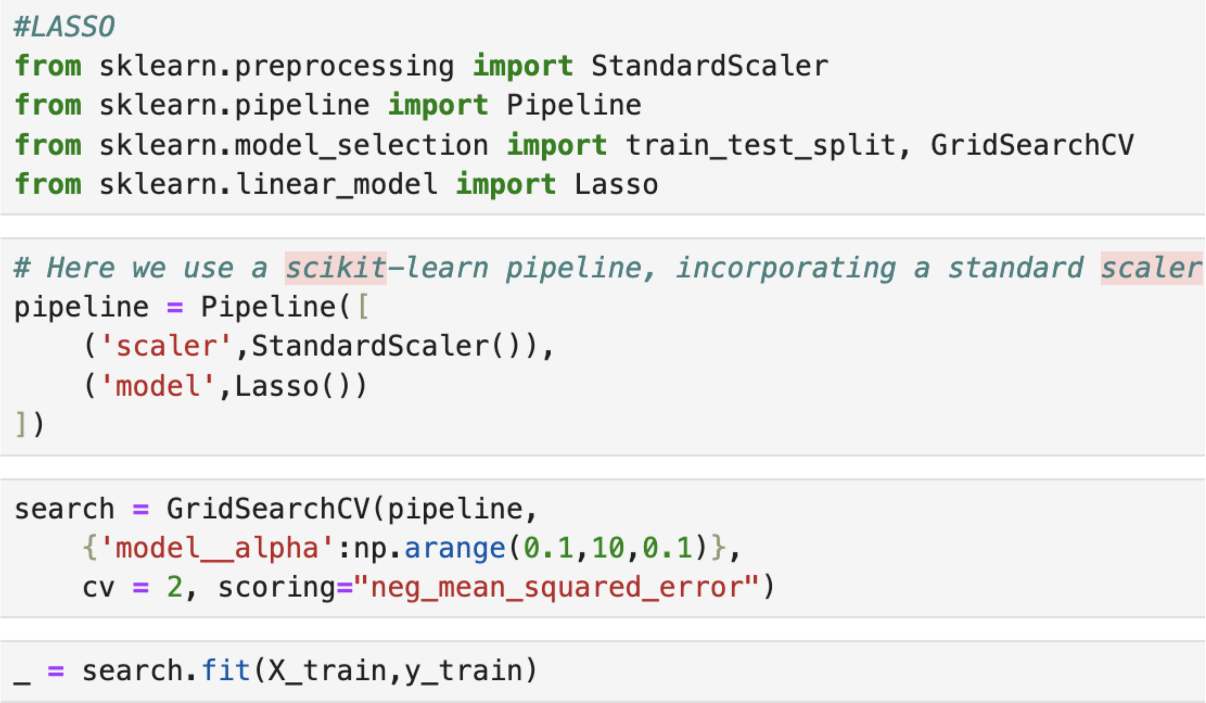
```
age: 0.0041
cons.price.idx: 0.0046
cons.conf.idx: 0.0921
euribor3m: 0.0190
nr.employed: 0.7415
is_housing: 0.0001
treated: 0.0039
poutcome_nonexistent: 0.0018
poutcome_success: 0.1328
```

Figure 15. Feature importance after GridsearchCV

The order of importance does not change, however the importance of every variable following GridsearchCV has decreased significantly. This solves the dilemma of overestimated significance due to overfitting and multicollinearity. Most influential features remain to be macroeconomic indicators, for same potential reasons discussed above.

LASSO Model

The LASSO model is a linear regression model which adds a penalty term to the loss function based on the absolute values of the coefficients (Phan 2024d). Lasso regularization helps to regularise data by encouraging sparsity in the coefficient values, effectively performing feature selection, and simplifying the model by reducing overfitting. The data was fed again into GridsearchCV, with the integration of the LASSO model as follows:



```
#LASSO
from sklearn.preprocessing import StandardScaler
from sklearn.pipeline import Pipeline
from sklearn.model_selection import train_test_split, GridSearchCV
from sklearn.linear_model import Lasso
```

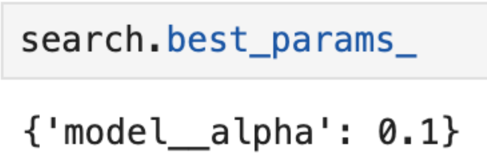
```
# Here we use a scikit-learn pipeline, incorporating a standard scaler
pipeline = Pipeline([
    ('scaler',StandardScaler()),
    ('model',Lasso())
])
```

```
search = GridSearchCV(pipeline,
    {'model__alpha':np.arange(0.1,10,0.1)},
    cv = 2, scoring="neg_mean_squared_error")
```

```
_ = search.fit(X_train,y_train)
```

Figure 16. Re-running GridseachCV with LASSO model

The hyperparameters of the optimal model specify an alpha value (penalization term) equal to 0.1. This choice indicates a relatively small penalization term. The rationale behind this decision lies in the sample data's size, which comprises fewer than 30 features. However, this modest penalization may expose the model to the risk of underfitting. Notably, this scenario highlights a potential limitation when applying a LASSO model (Phan 2024d).

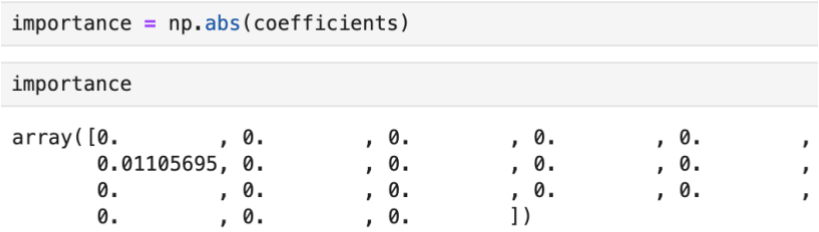


```
search.best_params_
```

```
{'model__alpha': 0.1}
```

Figure 17. Hyperparameters of best model

All feature importance levels can be visualised as below, with only one feature deemed important LASSO regulation:

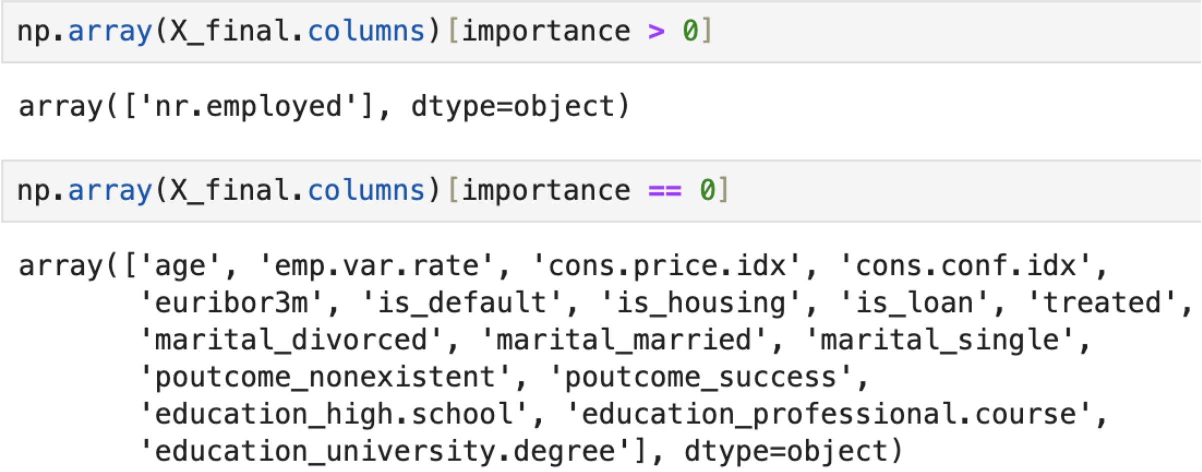


```
importance = np.abs(coefficients)
```

```
importance
```

```
array([0.        , 0.        , 0.        , 0.        , 0.        ,
       0.01105695, 0.        , 0.        , 0.        , 0.        ,
       0.        , 0.        , 0.        , 0.        , 0.        ,
       0.        , 0.        , 0.        ])
```

Figure 18. Array of importance coefficients



```
np.array(X_final.columns)[importance > 0]
```

```
array(['nr.employed'], dtype=object)
```

```
np.array(X_final.columns)[importance == 0]
```

```
array(['age', 'emp.var.rate', 'cons.price.idx', 'cons.conf.idx',
       'euribor3m', 'is_default', 'is_housing', 'is_loan', 'treated',
       'marital_divorced', 'marital_married', 'marital_single',
       'poutcome_nonexistent', 'poutcome_success',
       'education_high.school', 'education_professional.course',
       'education_university.degree'], dtype=object)
```

After aligning feature names and coefficients, it becomes evident that the employment rate stands out as the sole significant feature under the penalisation term of the LASSO model. This contrasts vastly compared to the two other models, which also considered other factors to be important.

Conclusion

After constructing three machine learning models to predict the success rate of subscribing to a term deposit following a marketing campaign, we deduce that the employment rate significantly influences the outcome. Additionally, the success of previous campaigns strongly impacts future success rates. Armed with this insight, the bank can tailor its marketing efforts toward employed individuals who have been exposed to prior campaigns.

However, it’s crucial to note that our model captures correlation rather than causation. The factors mentioned above do not directly cause an increase in term deposits; other variables are also at play. Furthermore, multicollinearity poses a challenge—variables affecting the outcome are correlated with each other. To mitigate this, we recommend removing unnecessary data during the cleaning process (e.g., ‘contact,’ ‘month,’ ‘day_of_week,’ ‘duration,’ and ‘campaign’) and applying regularization techniques such as the LASSO model.

References

Nguyen M (2024) ‘Industry talk: Tyme Group’ [PowerPoint slides, ECON1612], RMIT University, Melbourne.

Phan C (2024a) ‘What Are Big Data?' [PowerPoint slides, ECON1612], RMIT University, Melbourne.

Phan C (2024b) ‘How Machine Learning Can Improve Our Predictions?' [PowerPoint slides, ECON1612], RMIT University, Melbourne.

Phan C (2024c) ‘Insights into the mechanics of Machine Learning?' [PowerPoint slides, ECON1612], RMIT University, Melbourne.

Phan C (2024d) ‘Off-The-Shelf ML Models: Lasso and Ridge' [PowerPoint slides, ECON1612], RMIT University, Melbourne.