Data Glacier

Your Deep Learning Partner

Bank Marketing

virtual internship

Dec-15-2022

Agenda

Background

EDA

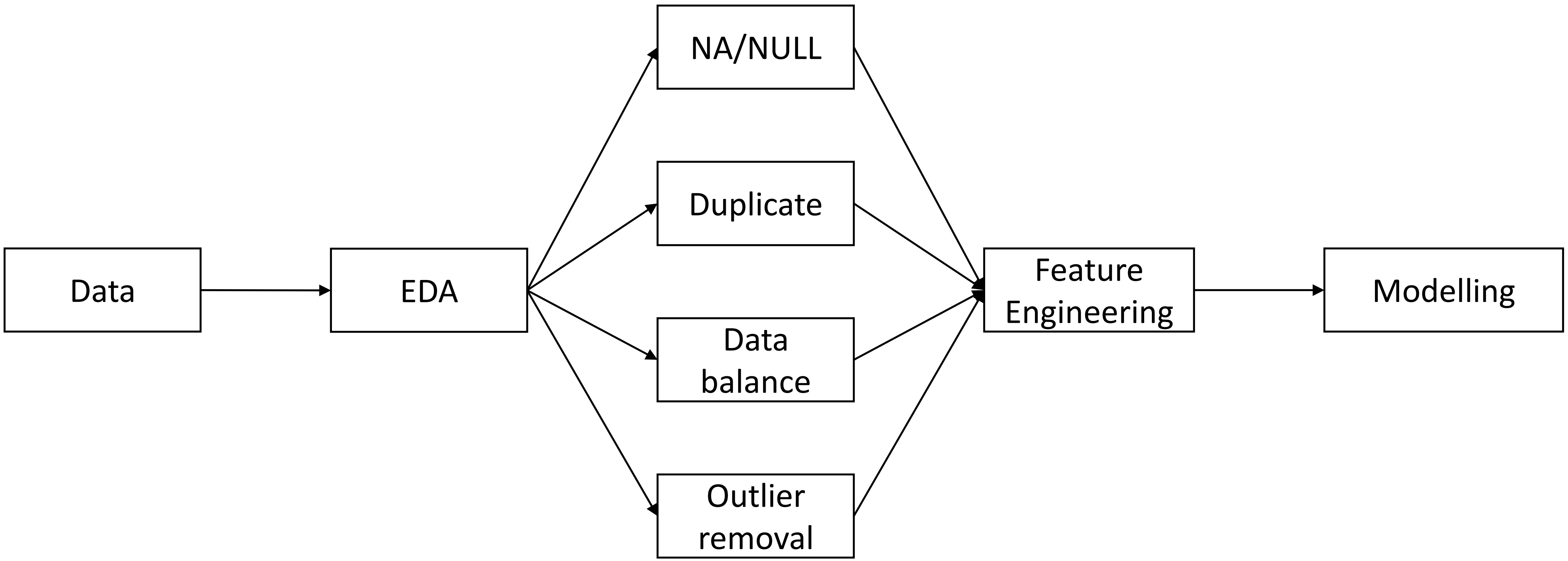
Modelling Recommendations

Background

- ABC Bank wants to sell its term deposit product to customers and before launching the product they want to develop a model which can help them in understanding whether a particular customer will buy their product or not (based on customer's past interaction with the bank or other Financial Institutions).

- Goals : ABC Bank wants to use ML models to shortlist customers whose chances of buying the product is more so that their marketing channel (tele marketing, SMS/email marketing etc) can focus only on those customers.

Roadmap


Data
EDA
NA/NULL
Duplicate
Data balance
Outlier removal
Feature Engineering
Modelling

Data info

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 41188 entries, 0 to 41187
Data columns (total 21 columns):
 #   Column          Non-Null Count  Dtype  
---  ------          --------------  -----  
 0   age             41188 non-null  int64  
 1   job             41188 non-null  object 
 2   marital         41188 non-null  object 
 3   education       41188 non-null  object 
 4   default         41188 non-null  object 
 5   housing         41188 non-null  object 
 6   loan            41188 non-null  object 
 7   contact         41188 non-null  object 
 8   month           41188 non-null  object 
 9   day_of_week     41188 non-null  object 
 10  duration        41188 non-null  int64  
 11  campaign        41188 non-null  int64  
 12  pdays           41188 non-null  int64  
 13  previous        41188 non-null  int64  
 14  poutcome        41188 non-null  object 
 15  emp.var.rate    41188 non-null  float64
 16  cons.price.idx  41188 non-null  float64
 17  cons.conf.idx   41188 non-null  float64
 18  euribor3m       41188 non-null  float64
 19  nr.employed     41188 non-null  float64
 20  y               41188 non-null  object 
dtypes: float64(5), int64(5), object(11)
memory usage: 6.6+ MB
```

```
categorical
```

```
['job',
 'marital',
 'education',
 'default',
 'housing',
 'loan',
 'contact',
 'month',
 'day_of_week',
 'poutcome',
 'y']
```

21 features in total

11 categorical features

Missing values

```
data.isnull().sum()
```

```
age               0
job               0
marital           0
education         0
default           0
housing           0
loan              0
contact           0
month             0
day_of_week       0
duration          0
campaign          0
pdays             0
previous          0
poutcome          0
emp.var.rate      0
cons.price.idx    0
cons.conf.idx     0
euribor3m         0
nr.employed       0
y                 0
dtype: int64
```

```
data.isna().sum()
```

```
age               0
job               0
marital           0
education         0
default           0
housing           0
loan              0
contact           0
month             0
day_of_week       0
duration          0
campaign          0
pdays             0
previous          0
poutcome          0
emp.var.rate      0
cons.price.idx    0
cons.conf.idx     0
euribor3m         0
nr.employed       0
y                 0
dtype: int64
```

0 Null or NA values

Duplicate rows

```
data.duplicated().value_counts()
```

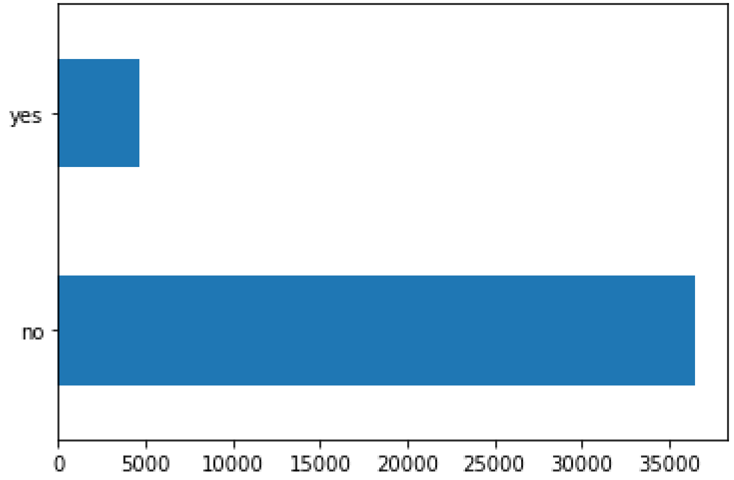
```
False    41176
True        12
dtype: int64
```

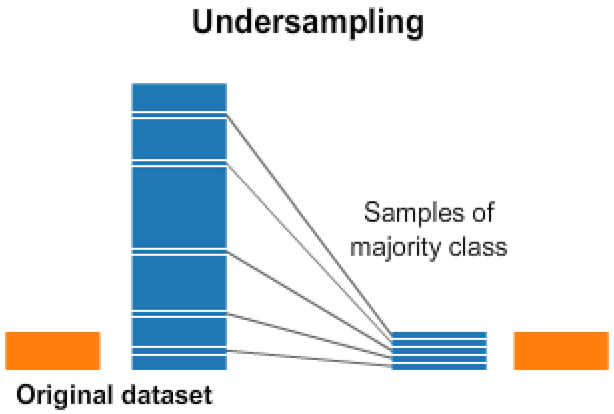
12 duplicate rows

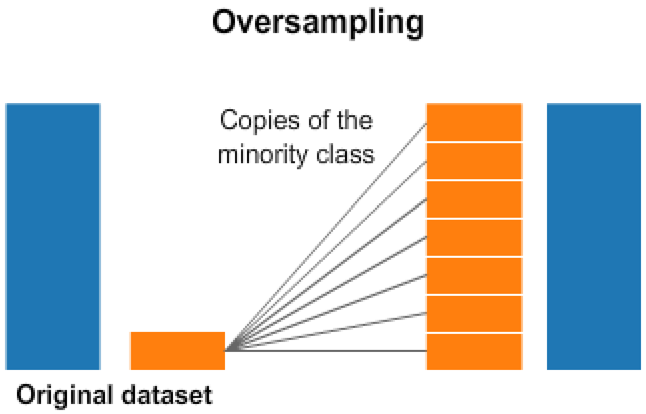
Data balance

```
data['y'].value_counts()
```

```
no     36537
yes     4639
Name: y, dtype: int64
```



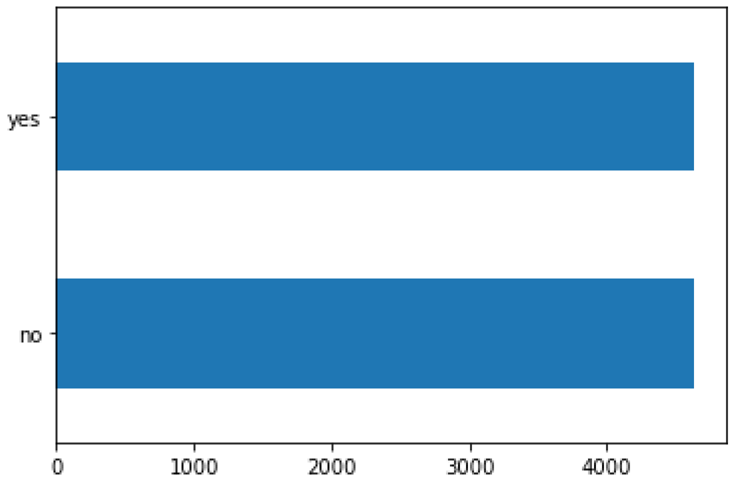




Data balance – undersampling and oversampling

```
data_under['y'].value_counts().plot(kind='barh')
```

<AxesSubplot:>



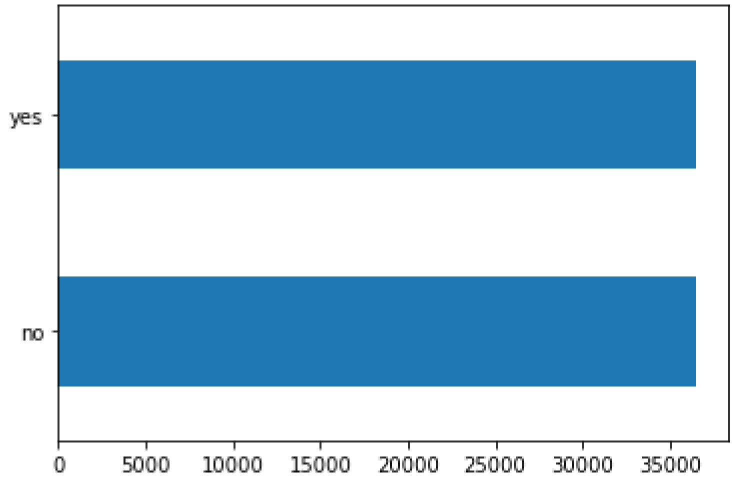

```
data_under.duplicated().value_counts()
```

```
False    9278
dtype: int64
```

0 duplicate row for undersampling

```
data_over['y'].value_counts().plot(kind='barh')
```

<AxesSubplot:>




```
data_over.duplicated().value_counts()
```

```
False    41175
True     31899
dtype: int64
```

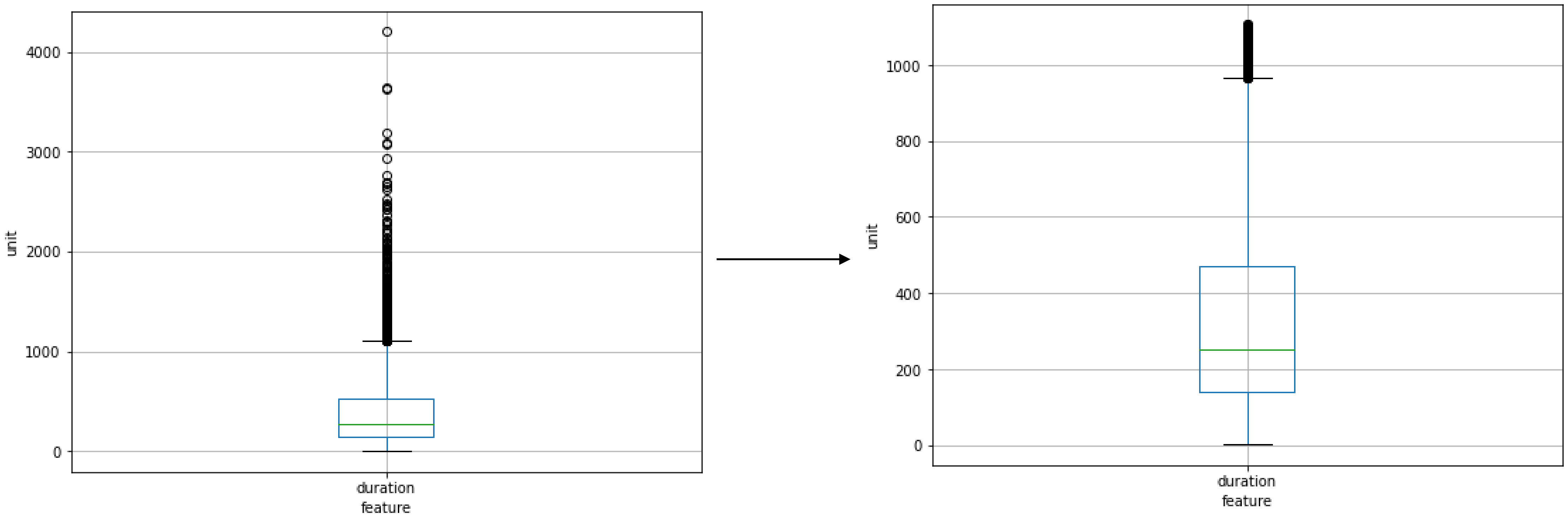
Many duplicate rows for oversampling

Outlier removal

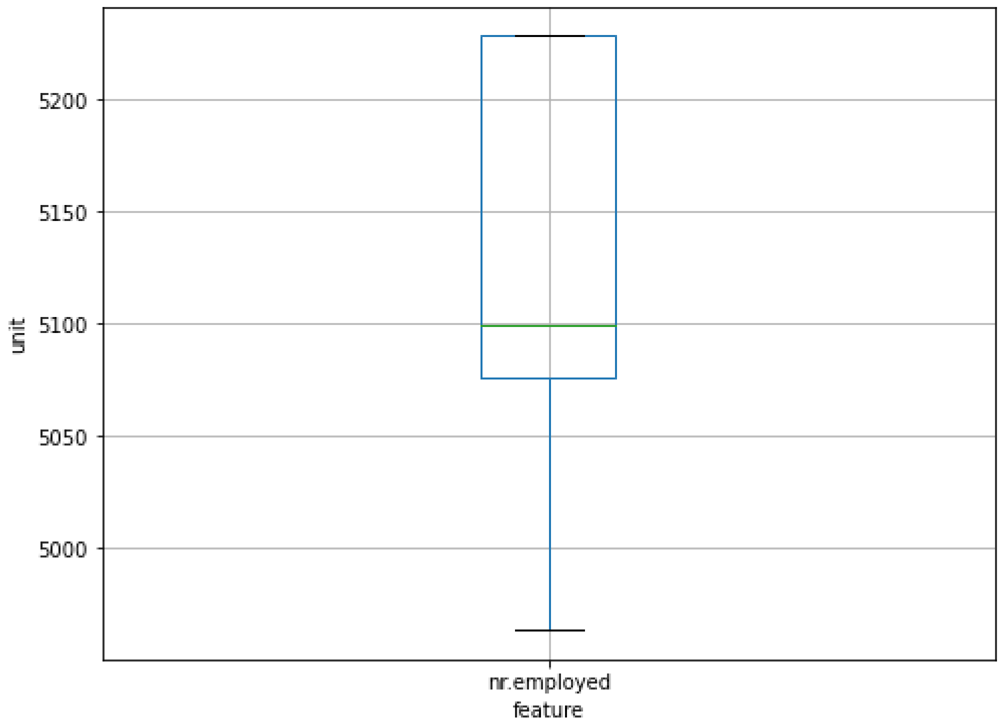
```
data_under.describe()
```

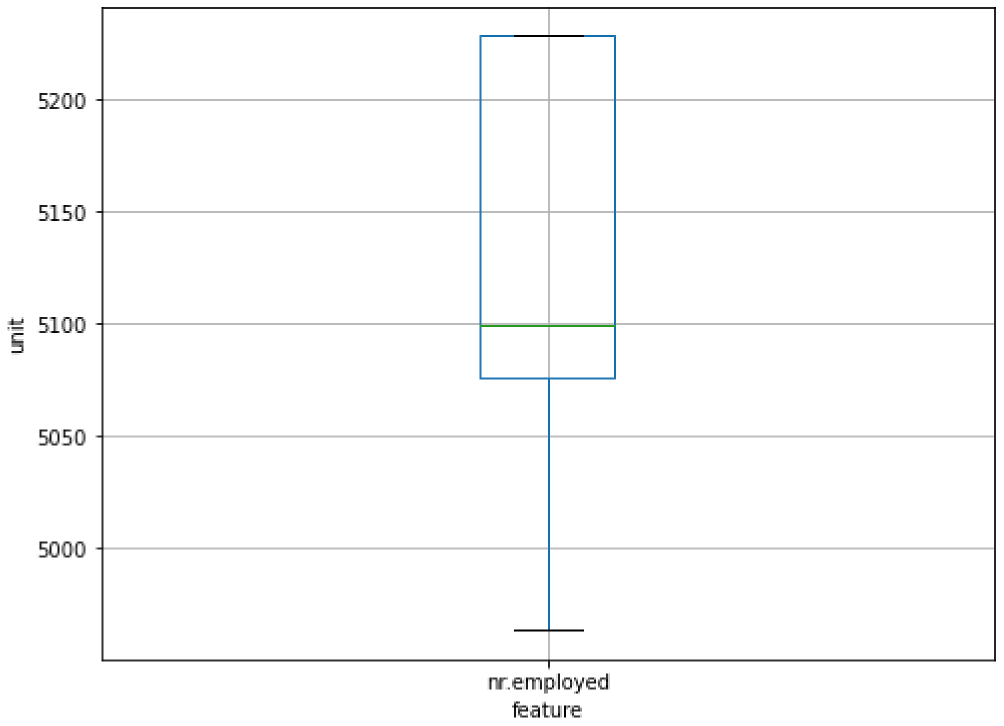
	age	duration	campaign	pdays	previous	emp.var.rate	cons.price.idx	cons.conf.idx	euribor3m	nr.employed
count	9278.000000	9278.000000	9278.000000	9278.000000	9278.000000	9278.000000	9278.000000	9278.000000	9278.000000	9278.000000
mean	40.432960	389.552598	2.326471	887.687217	0.312891	-0.489211	93.479128	-40.206585	2.971193	5135.853471
std	12.025501	360.742255	2.335458	313.304195	0.697914	1.722400	0.631613	5.344737	1.889231	87.171539
min	17.000000	3.000000	1.000000	0.000000	0.000000	-3.400000	92.201000	-50.800000	0.634000	4963.600000
25%	31.000000	145.000000	1.000000	999.000000	0.000000	-1.800000	92.893000	-42.700000	1.244000	5076.200000
50%	38.000000	266.000000	2.000000	999.000000	0.000000	-0.100000	93.444000	-41.800000	4.021000	5191.000000
75%	48.000000	530.000000	3.000000	999.000000	0.000000	1.400000	93.994000	-36.400000	4.959000	5228.100000
max	98.000000	4199.000000	43.000000	999.000000	6.000000	1.400000	94.767000	-26.900000	5.045000	5228.100000

Outlier removal

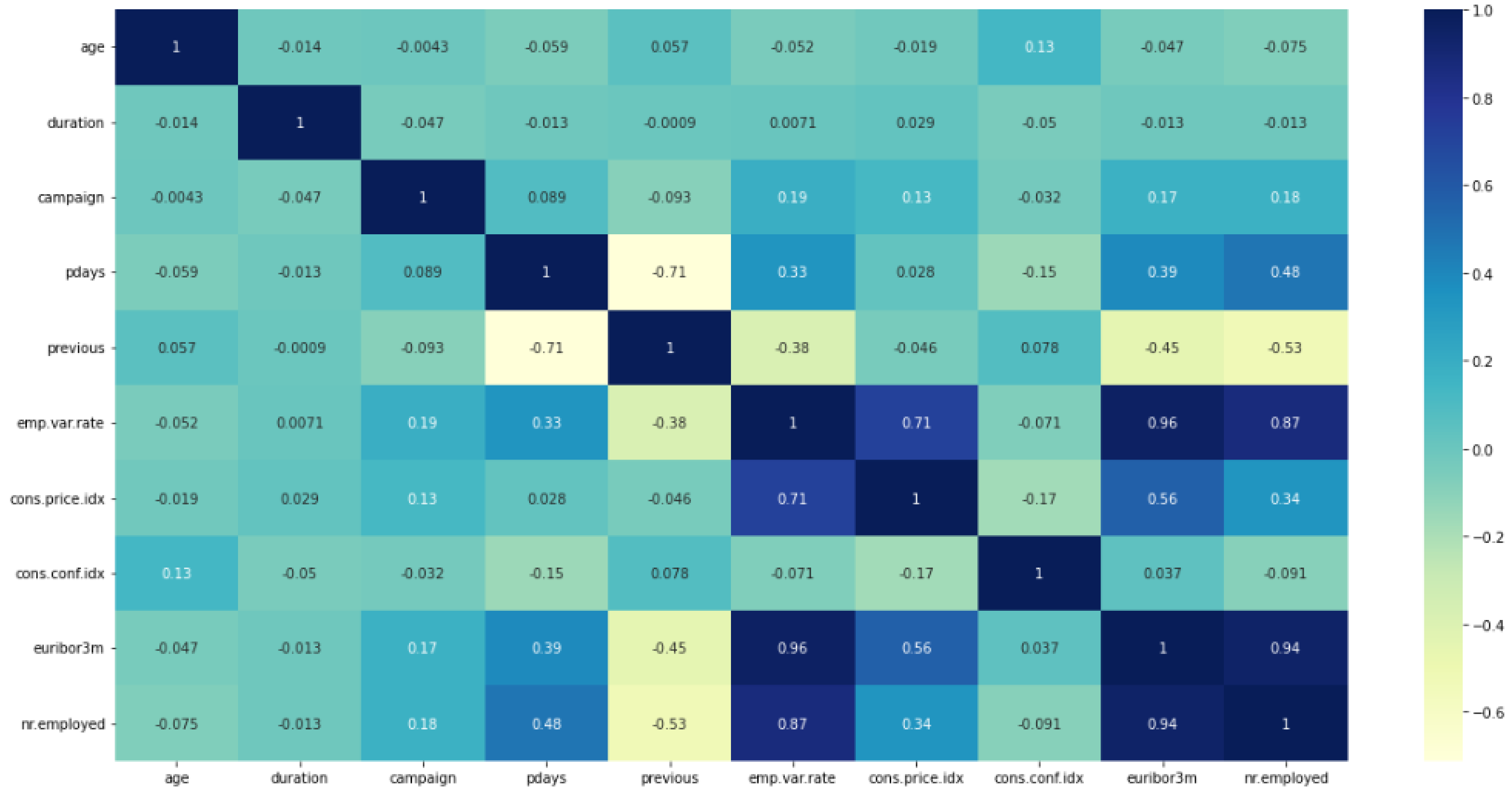


Outlier removal



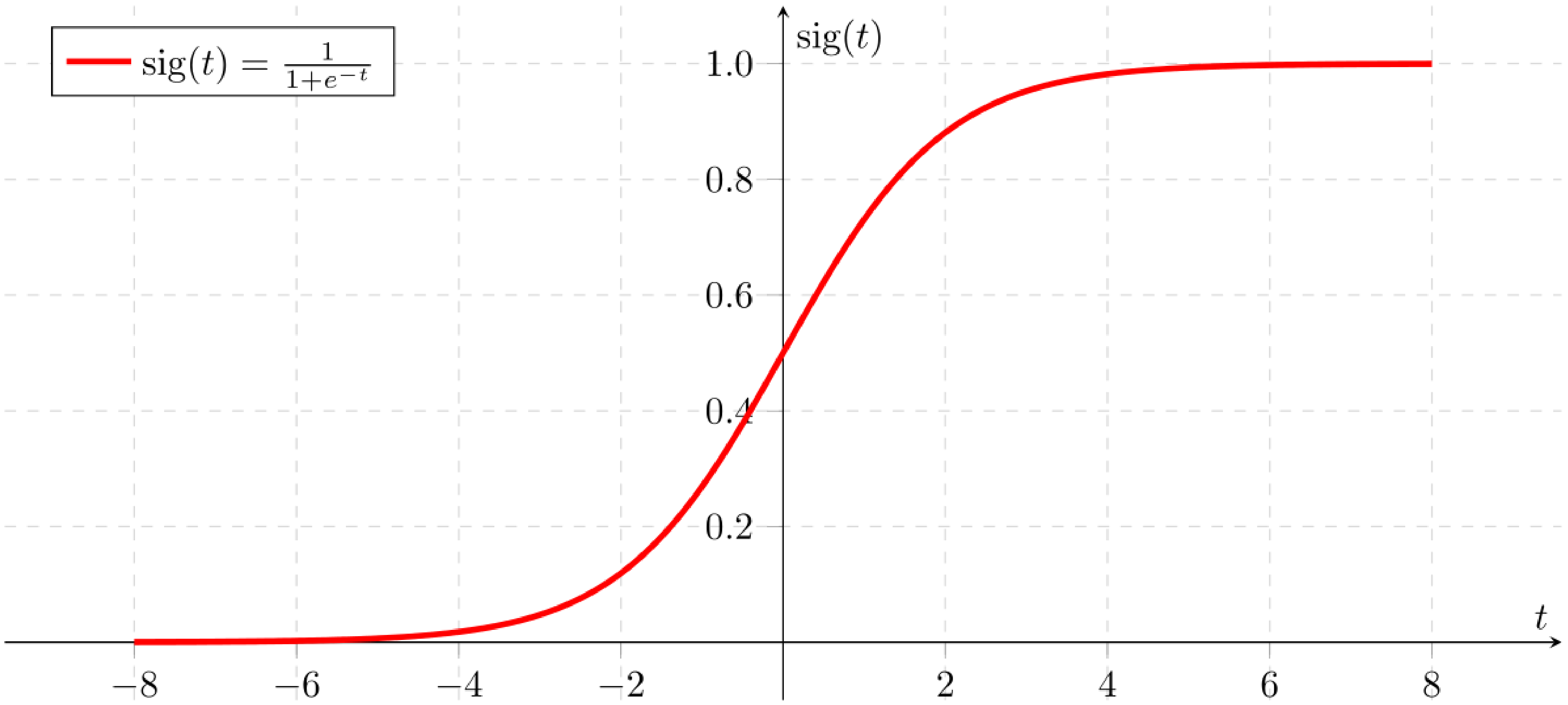


Feature Engineering



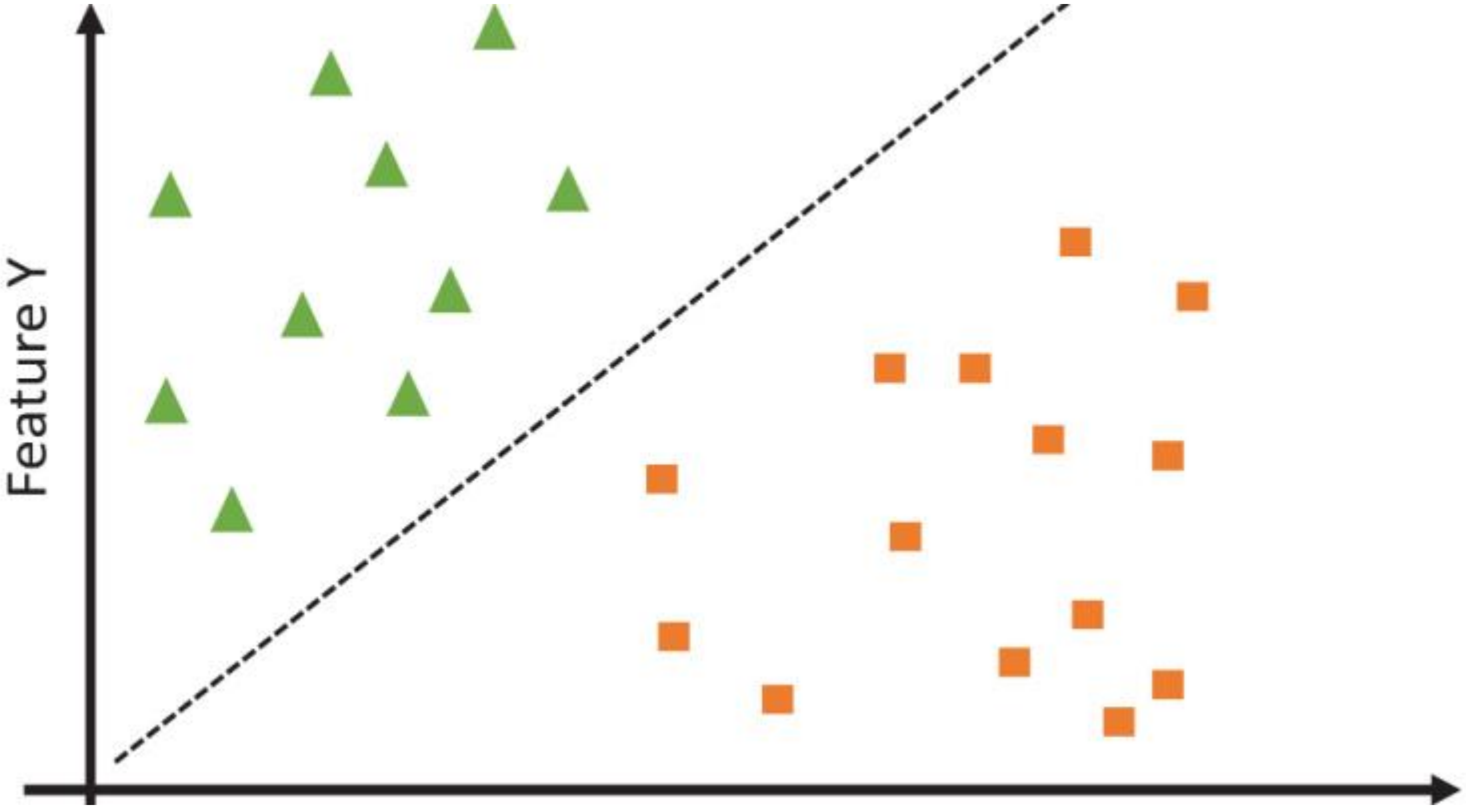
Modelling Recommendations

Logistic Regression



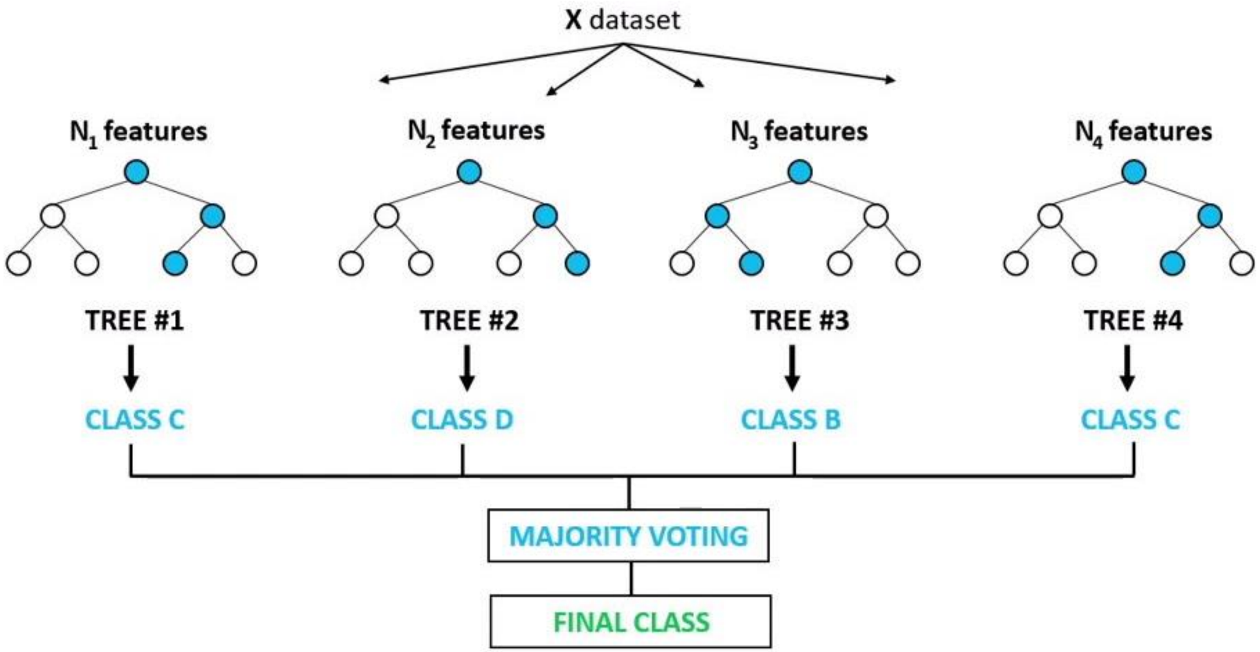
Modelling Recommendations

SVM



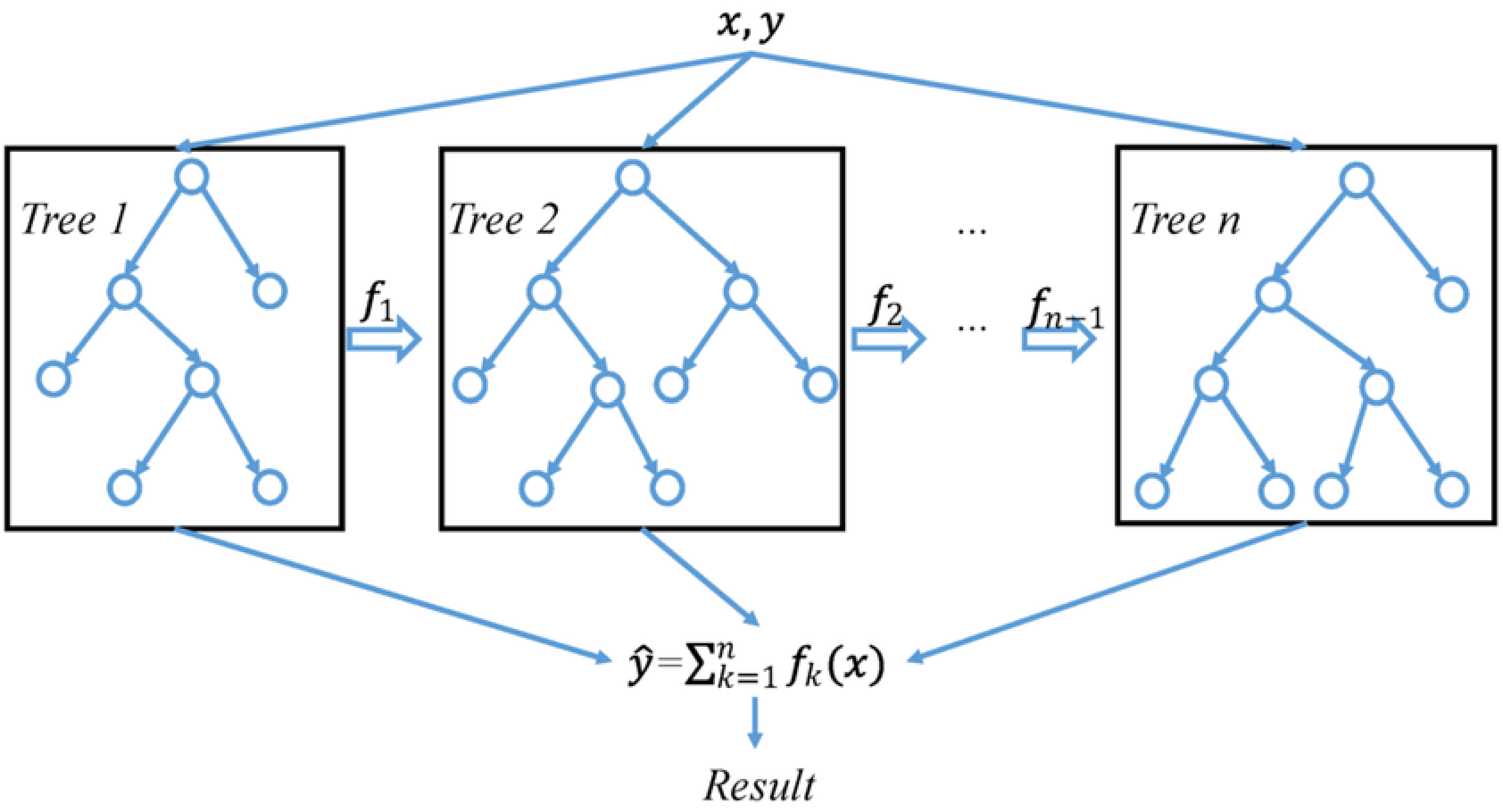
Modelling Recommendations

Random Forest



Modelling Recommendations

Xgboost



Thank You

Data Glacier
Your Deep Learning Partner