Index

Case Study: Banking Marketing 2

I. Dataset Analysis..... 2

II. Feature Engineering 3

III Data Modeling 4

IV. Model Evaluation 8

Case Study: Banking Marketing

Y.B.

Subject: This project aims to predict if the client will take the term deposit product proposed by telemarketing activity.
The data is about direct marketing campaigns (phone calls) of a Portuguese banking institution coming from [Moro et al., 2014] available in the internet.
Coding Environment I used to work on this project is Jupyter Notebook on local PC

The project is coded in Python 3.6 and covers the parts such as dataset analysis, feature engineering, data modeling, model evaluation.

I. Dataset Analysis

As is shown in the following picture, we can see from Pandas that the dataset contains 45,211 records and are composed of numeric type, datetime type, categorical type and binary type.

```
In [25]: df.describe()
```

Out[25]:

	age	duration	campaign	pdays	previous	emp.var.rate	cons.price.idx	cons.conf.idx	euribor3m	nr.employed
count	41188.00000	41188.000000	41188.000000	41188.000000	41188.000000	41188.000000	41188.000000	41188.000000	41188.000000	41188.000000
mean	40.02406	258.285010	2.567593	962.475454	0.172963	0.081886	93.575664	-40.502600	3.621291	5167.035911
std	10.42125	259.279249	2.770014	186.910907	0.494901	1.570960	0.578840	4.628198	1.734447	72.251528
min	17.00000	0.000000	1.000000	0.000000	0.000000	-3.400000	92.201000	-50.800000	0.634000	4963.600000
25%	32.00000	102.000000	1.000000	999.000000	0.000000	-1.800000	93.075000	-42.700000	1.344000	5099.100000
50%	38.00000	180.000000	2.000000	999.000000	0.000000	1.100000	93.749000	-41.800000	4.857000	5191.000000
75%	47.00000	319.000000	3.000000	999.000000	0.000000	1.400000	93.994000	-36.400000	4.961000	5228.100000
max	98.00000	4918.000000	56.000000	999.000000	7.000000	1.400000	94.767000	-26.900000	5.045000	5228.100000

```
In [26]: df.head(100)
```

Out[26]:

	age	job	marital	education	default	housing	loan	contact	month	day_of_week	duration	campaign	pdays	previous	poutcom
0	56	housemaid	married	basic.4y	no	no	no	telephone	may	mon	261	1	999	0	nonexisten
1	57	services	married	high.school	unknown	no	no	telephone	may	mon	149	1	999	0	nonexisten
2	37	services	married	high.school	no	yes	no	telephone	may	mon	226	1	999	0	nonexisten
3	40	admin.	married	basic.6y	no	no	no	telephone	may	mon	151	1	999	0	nonexisten
4	56	services	married	high.school	no	no	yes	telephone	may	mon	307	1	999	0	nonexisten
5	45	services	married	basic.9y	unknown	no	no	telephone	may	mon	198	1	999	0	nonexisten
6	59	admin.	married	professional.course	no	no	no	telephone	may	mon	139	1	999	0	nonexisten
7	41	blue-collar	married	unknown	unknown	no	no	telephone	may	mon	217	1	999	0	nonexisten
8	24	technician	single	professional.course	no	yes	no	telephone	may	mon	380	1	999	0	nonexisten
9	25	services	single	high.school	no	yes	no	telephone	may	mon	50	1	999	0	nonexisten
10	41	blue-collar	married	unknown	unknown	no	no	telephone	may	mon	55	1	999	0	nonexisten

The data can be classified as the following features:

1. Identity features

Age, job, marital, education, default, housing, loan, contact

2. Transformable features

Month: last contact month of year (categorical: 'jan', 'feb', 'mar', ..., 'nov', 'dec')
day_of_work: last contact day of the week (categorical: 'mon','tue','wed','thu','fri')
Duration: last contact duration, in seconds (numeric)

3. Campaign features:

campaign: number of contacts performed during this campaign and for this client (numeric, includes last contact)

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)
previous: number of contacts performed before this campaign and for this client (numeric)
poutcome: outcome of the previous marketing campaign (categorical: 'failure','nonexistent','success')

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

II. Feature Engineering

Now I'm going to work on important features serving for modeling:

1. Defining the outcome

The column 'y' contains the information about if the client has subscribed a term deposit or not. It is represented by a binary type data: 'yes' or 'no'. This value is to defined as the outcome of modeling, also known as the outcome of the prediction.

```
In [28]: df.y.value_counts()

Out[28]: no     36548
         yes     4640
         Name: y, dtype: int64
```

2. Data preprocessing

1)Check if there's missing data or abnormal data

```
In [38]: #Data preprocessing, checking if there
         df.isna().sum()

Out[38]: 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
```

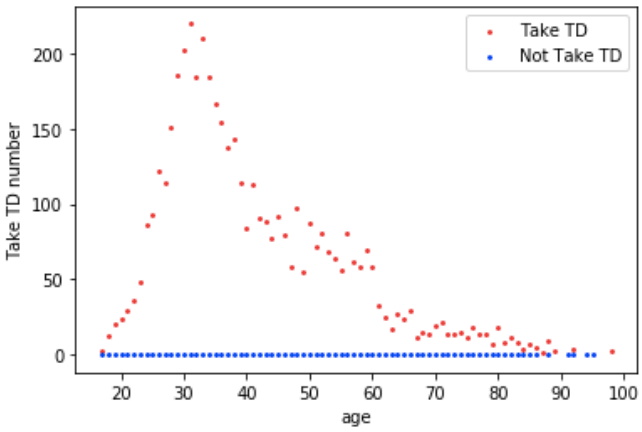
No missing data is found and data looks good.

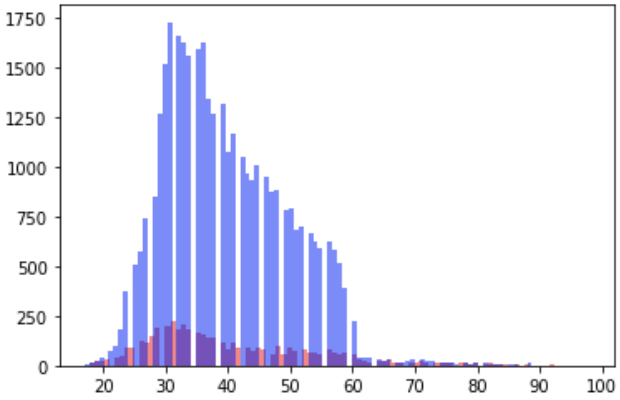
2) Transform the column y from yes or no into numeric values to fit the prediction model
3) Retreat categorical data into numeric to fit the model

3. Visualization features distribution

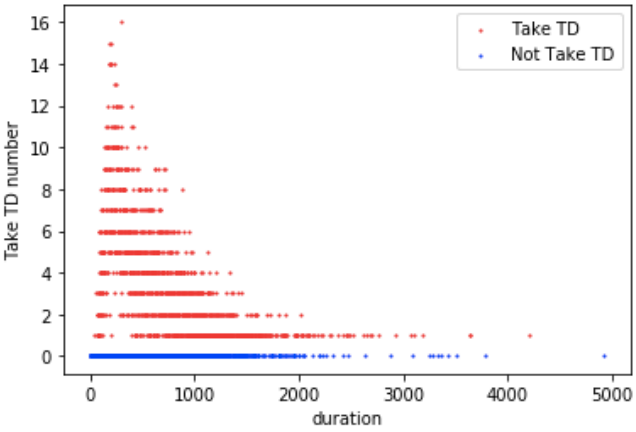
I'll use data visualization to detect the important features. Some might be meaningful for the prediction. The following numeric parameters look interesting

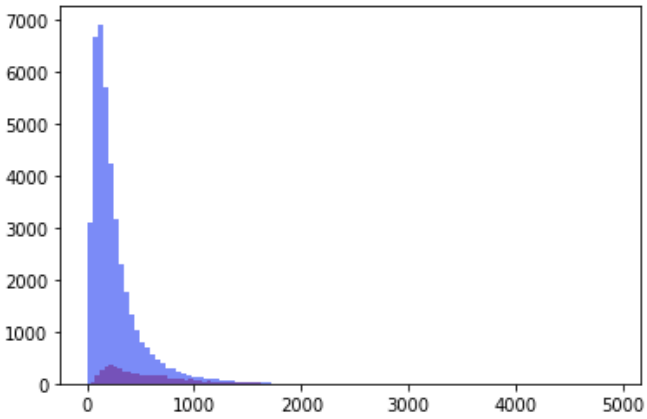
1) Age





2) duration





III Data Modeling

I will now fit the models with different features and predict y

Model 1 LG: using Logistic regression model with different parameters

1) I will firstly work on 3 parameters: duration, euribor3m, housing, and got the evaluation metrics as follow:

```
cf,cf_f1,cf_precision,cf_recall
```

```
(array([[7126,  168],
        [ 686,  258]]),
 array([0.94346617, 0.37664234]),
 array([0.91218638, 0.6056338 ]),
 array([0.97696737, 0.27330508]))
```

```
np.unique(y_test, return_counts=True)
```

```
(array([0, 1]), array([7294,  944]))
```

```
a=metrics.accuracy_score(y_test,y_pred)
```

The confusion matrix shows that the accuracy of the model is 0.8963. The model successfully predicted 7126 Not_take_TD cases out of 7294 in the test set; it predicted 258 cases of take_TD out of 944 in the test set.

2) I then work on 4 parameters: ages, duration, euribor3m, housing and got the evaluation metrics as follow:

```
cf,cf_f1,cf_precision,cf_recall
```

```
(array([[7131,  163],
        [ 682,  262]]),
 array([0.94406566, 0.38276114]),
 array([0.91270959, 0.61647059]),
 array([0.97765287, 0.27754237]))
```

```
np.unique(y_test, return_counts=True)
```

```
(array([0, 1]), array([7294,  944]))
```

```
a=metrics.accuracy_score(y_test,y_pred)
```

```
a
```

```
0.8974265598446225
```

The accuracy of the model is 0.8974.
The model successfully predicted 7131 Not_take_TD cases out of 7294 in the test set;
The model successfully predicted 262 Take_TD cases out of 944 cases in the test set
3) We now try 5 parameters: ages, duration, euribor3m, education, housing

```
cf,cf_f1,cf_precision,cf_recall
```

```
(array([[7133,  161],
        [ 677,  267]]),
 array([0.94451801, 0.38921283]),
 array([0.91331626, 0.62383178]),
 array([0.97792706, 0.28283898]))
```

```
np.unique(y_test, return_counts=True)
```

```
(array([0, 1]), array([7294,  944]))
```

```
a=metrics.accuracy_score(y_test,y_pred)
```

```
a
```

```
0.8982762806506434
```

4) We repeat the process and continue with 6 parameters: ages, duration, euribor3m, education, housing, marital

```
cf,cf_f1,cf_precision,cf_recall
```

```
(array([[7126,  168],
        [ 675,  269]]),
 array([0.94415369, 0.38957277]),
 array([0.91347263, 0.61556064]),
 array([0.97696737, 0.28495763]))
```

```
np.unique(y_test, return_counts=True)
```

```
(array([0, 1]), array([7294,  944]))
```

```
a=metrics.accuracy_score(y_test,y_pred)
```

```
a
```

```
0.8976693372177713
```

The accuracy is decreased, but the Take_TD cases prediction has improved from 267 to 269. The adding parameter marital doesn't seem to be helpful.
5) Now I change the parameter "marital" to "poutcome", the result seems much better after adding this parameter

```
cf,cf_f1,cf_precision,cf_recall
```

```
(array([[7096,  198],
        [ 578,  366]]),
 array([0.94815607, 0.48541114]),
 array([0.92468074, 0.64893617]),
 array([0.9728544 , 0.38771186]))
```

```
np.unique(y_test, return_counts=True)
```

```
(array([0, 1]), array([7294,  944]))
```

```
a=metrics.accuracy_score(y_test,y_pred)
```

```
a
```

```
0.9058023792182569
```

The model reached an accuracy of 0.9058 and successfully predicted 366 out of 944 Take_TD cases.
6) I try to reduce the most meaningful parameters which are ages, duration, euribor3m, poutcome and the result is even better. We found increasing parameters are very much time consuming and can't improve the model, so we stay with 4 parameters ages, duration, euribor3m, poutcome:

```
cf,cf_f1,cf_precision,cf_recall
```

```
(array([[7104,  190],
        [ 578,  366]]),
 array([0.94871795, 0.488     ]),
 array([0.92475918, 0.65827338]),
 array([0.97395119, 0.38771186]))
```

```
np.unique(y_test, return_counts=True)
```

```
(array([0, 1]), array([7294,  944]))
```

```
a=metrics.accuracy_score(y_test,y_pred)
```

```
a
```

```
0.9067734887108522
```

We reached an accuracy of 0.9067 and successfully predicted 366 out of 944 taking term deposit cases(38,78%).

The next step is to try other models to see if some can improve the prediction.
Model 2 KNN: using KNN model with 4 parameters: ages, duration, euribor3m, poutcome
This KNN model got an accuracy of 0.9012. It successfully predicts 3491 Not_take_TD

cases out of 3643 in the test set (95%), and predicted 221 cases of Not_Take_TD out of 476 cases in the test set(46.42%)

```
a1
```

```
0.9011896091284293
```

```
cf=metrics.confusion_matrix(y_test2, yk_pred)
cf_f1=metrics.f1_score(y_test2,yk_pred,average=None)
cf_precision=metrics.precision_score(y_test2,yk_pred,average=None)
cf_recall=metrics.recall_score(y_test2,yk_pred,average=None)
```

```
cf,cf_f1,cf_precision,cf_recall
```

```
(array([[3491,  152],
        [ 255,  221]]),
 array([0.94491812, 0.52061249]),
 array([0.93192739, 0.5924933 ]),
 array([0.95827615, 0.46428571]))
```

```
np.unique(y_test2,return_counts=True)
```

```
(array([0, 1]), array([3643,  476]))
```

Let's also see how SVM performs.

Model 3 SVM: using SVM model with 4 parameters: ages, duration, euribor3m, poutcome

```
cf,cf_f1,cf_precision,cf_recall,a_svm
```

```
(array([[7128,  166],
        [ 616,  328]]),
 array([0.9479984 , 0.45618915]),
 array([0.92045455, 0.66396761]),
 array([0.97724157, 0.34745763]),
 0.9050740470988103)
```

```
np.unique(y_test, return_counts=True)
```

```
(array([0, 1]), array([7294,  944]))
```

```
np.unique(y_svm,return_counts=True)
```

```
(array([0, 1]), array([7744,  494]))
```

Here we get an accuracy of 0.90507, It successfully predicts 7128 Not_take_TD cases out of 7294 in the test set, and predicted 328 cases of Take_TD out of 944 cases in the test set(34.75%)

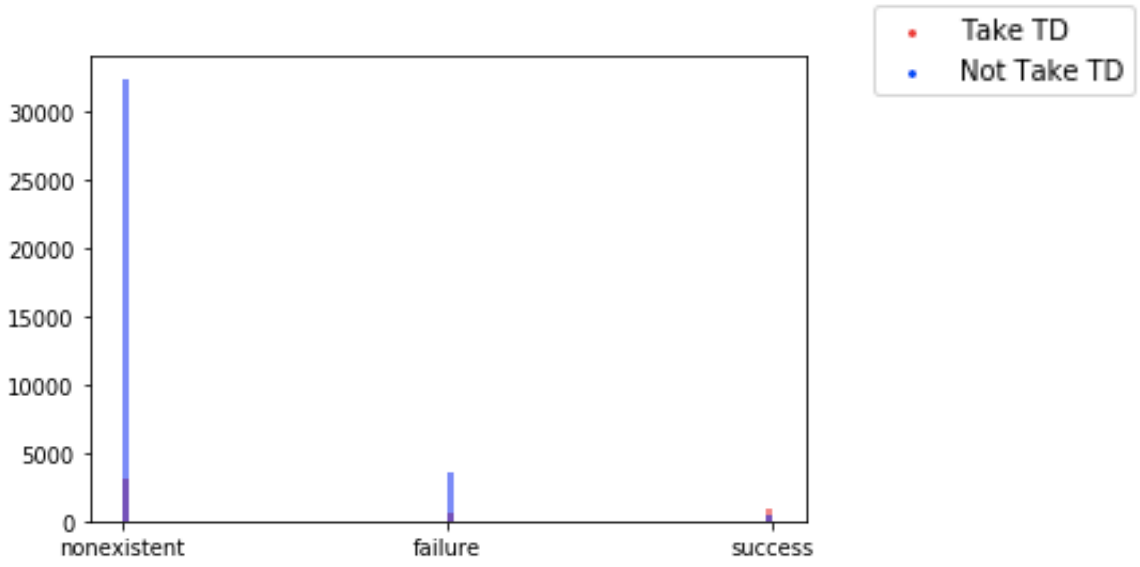
IV. Model Evaluation

Now I'm going to list three models with evaluation metrics so that we can have a comparison of different models.

4 Parameters	Logistic Regression	KNN(K=7)	SVM
Running time	quick	normal	long
Accuracy	0.9067	0.9012	0.881039
F1 score	0.9487	0.9449	0.90507
TP: Not Taking Term Deposit prediction	0.9740	0.9583	0.9772
TN: Taking Term Deposit Prediction	0.3878	0.4642	0.3475

The LG model is the fastest running model. It also has the best accuracy and F1 score. The KNN model predicts best the cases of Take_TD. It also has a high F1 score. It predicts a bit less good the Not_take_TD outcome. The SVM model takes long to run. It predicts best the Not_take_TD outcome. Its score of Accuracy and F1 are lower than the other models.

The model also shows that poutcome is a very meaningful factor for the prediction.



The visualization of the parameter poutcome shows that the clients having experience previous successfully have higher possibility to take the term deposit.