FINANCIAL TELEMARKETING

By: James Mulyono, Shreyas Sharma & Sultan Almuhairy

IDENTIFYING THE ISSUE

Telemarketing calls, when unsuccessfully not only waste time of the banking institutions but also of the call recipients whilst also diminishing the chance of the customer returning to the bank for its services. It increases operating costs with limited success. With the use of a predictive model, selective targeting can be done, saving both time, and money.

THE RESOLUTION TO THE ISSUE

- Creating a special Random MLP model to help predict the likelihood of a successful outcome from a telemarketing call for banking customers in terms of updating their current plan with the bank.

COMPANIES THAT WOULD BE INTERSTED IN THIS MODEL

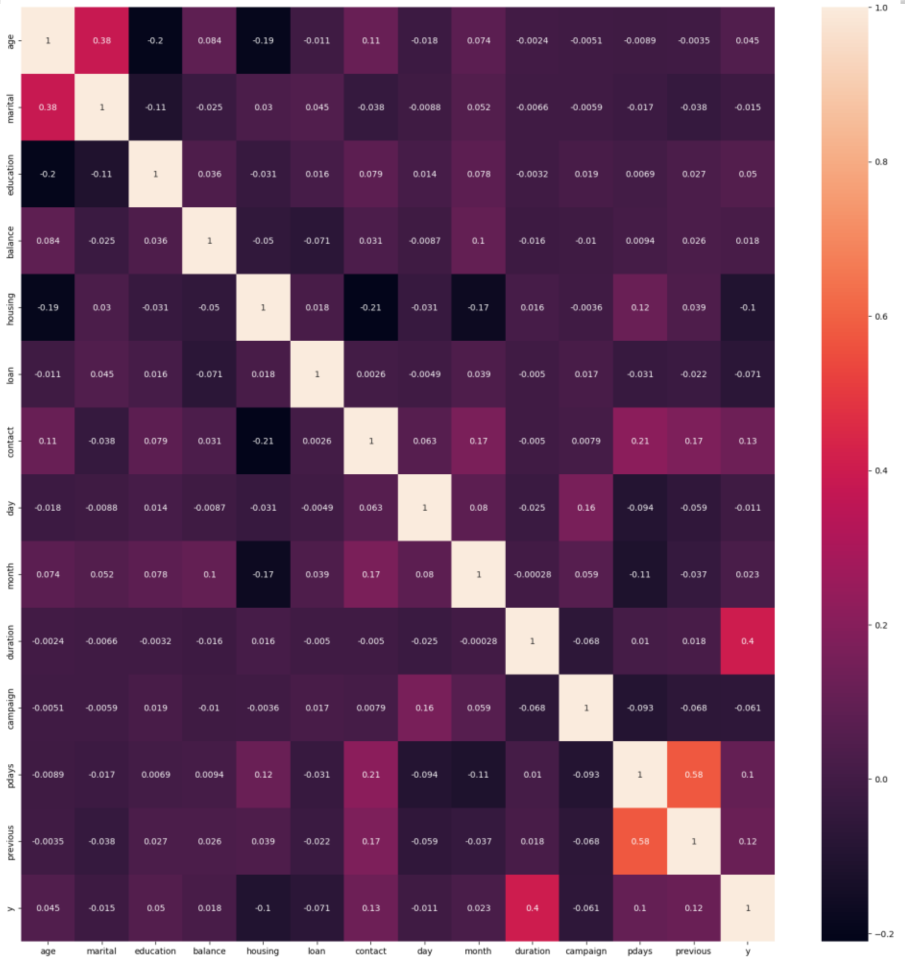
	age	job	marital	education	default	balance	housing	loan	contact	day	month	duration	campaign	pdays	previous	poutcome	y
0	30	unemployed	married	primary	no	1787	no	no	cellular	19	oct	79	1	-1	0	unknown	no
1	33	services	married	secondary	no	4789	yes	yes	cellular	11	may	220	1	339	4	failure	no
2	35	management	single	tertiary	no	1350	yes	no	cellular	16	apr	185	1	330	1	failure	no
3	30	management	married	tertiary	no	1476	yes	yes	unknown	3	jun	199	4	-1	0	unknown	no
4	59	blue-collar	married	secondary	no	0	yes	no	unknown	5	may	226	1	-1	0	unknown	no
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
4516	33	services	married	secondary	no	-333	yes	no	cellular	30	jul	329	5	-1	0	unknown	no
4517	57	self-employed	married	tertiary	yes	-3313	yes	yes	unknown	9	may	153	1	-1	0	unknown	no
4518	57	technician	married	secondary	no	295	no	no	cellular	19	aug	151	11	-1	0	unknown	no
4519	28	blue-collar	married	secondary	no	1137	no	no	cellular	6	feb	129	4	211	3	other	no
4520	44	entrepreneur	single	tertiary	no	1136	yes	yes	cellular	3	apr	345	2	249	7	other	no

4521 rows × 17 columns

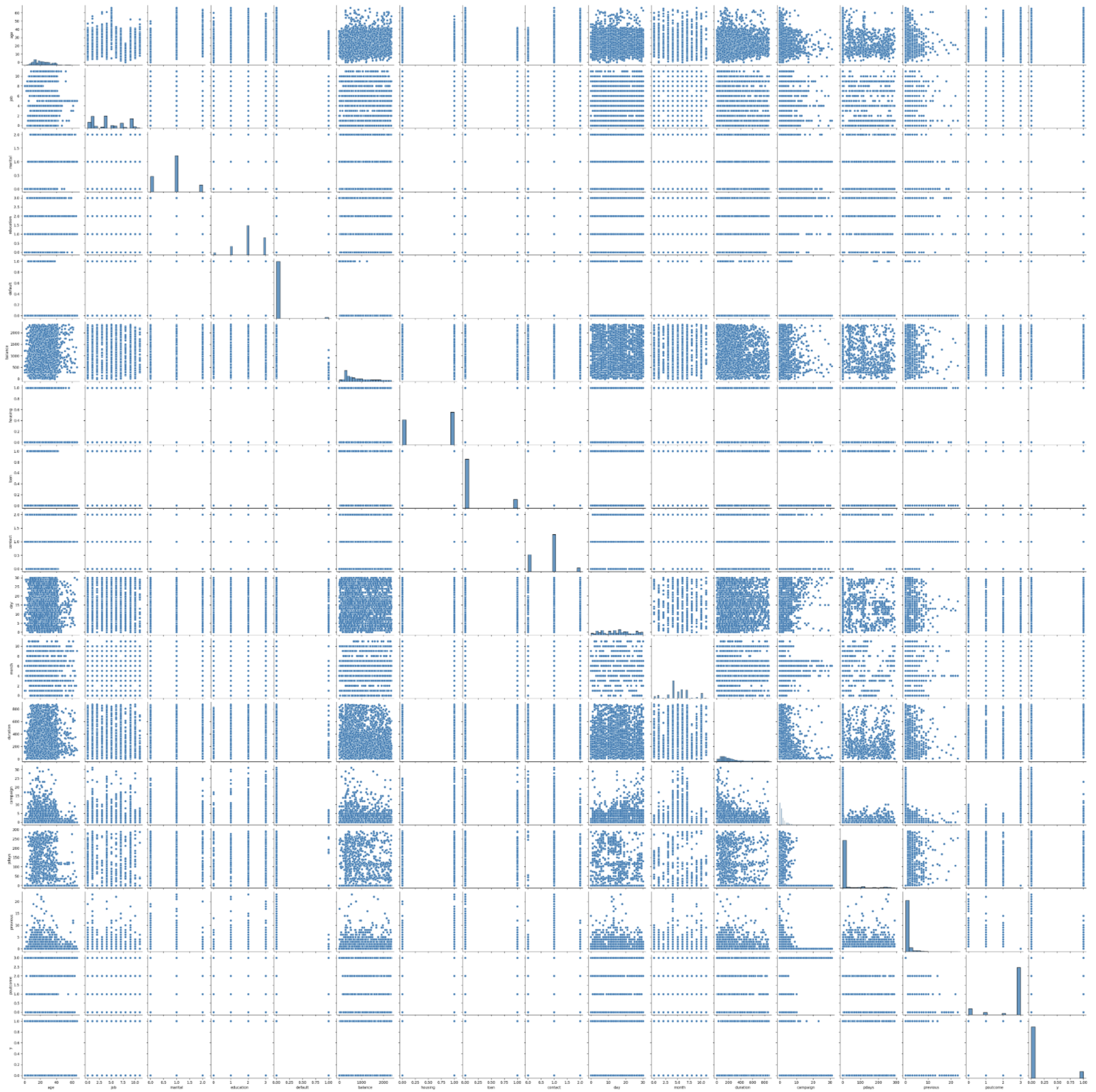
THE DATASET

- The set of data that we have is from a Portuguese banking institution, based on real-world phone calls
- The data set provides us with 17 features and includes over 45000 instances.
- Data collected in 2017 – 2018 in Portugal

HYPOTHESIS



- After taking a look at the data set and creating a correlation heatmap, we found the strongest corelation to be duration, but seeing as duration is uncontrolable, we opted to remove it as a variable, thus leading us to go with the next available option, that being Contact method.
- Therefore, our hypothesis ended up becoming: "What Method of contact is the most effective for the bank telemarketing industry?"


age
job
marital
education
default
balance
housing
loan
contact
day
month
duration
campaign
pdays
previous
poutcome
y

PAIRPLOT

TRAINING MODELS

To solve the problem at hand we have gone through and utilized several training models through analyze each variable mentioned previously to gain the highest accuracy. By training a new model for each separate variable using logistic regression, multilayer perceptron network, and random forest regressor we can accurately compare each variable's affect on the results.

MULTILAYER PERCEPTRON NEURAL NETWORK

For some reason throughout our data collection mlp and logistic regression had very similar results. On average when looking at all our different models they produced results of around 89% on our training data accuracy score and 90% for model accuracy score. Specifically for our mlp model we utilized a hidden layer size of 100,100 and max iteration of 500.

```
0.8827433628318584
0.892817679558011l
[[808   0]
 [ 97   0]]
0.8827433628318584
0.892817679558011l
[[808   0]
 [ 97   0]]
```

LOGISTIC REGRESSION

- As mentioned in the previous slide for most of our models both logistic regression and mlp classifier had same results. For the model we put a max iteration of 500.

RANDOM FOREST REGRESSION

- This training model throughout the experiment has proven to be the least accurate with a huge disparity between its rf score and accuracy. As such, data provided by aforementioned training model will be disregarded.

```
0.005183965744335217
0.8795580110497238
[[796    0]
 [109    0]]
```

```
4.587657819854485e-06
0.8928176795580111
[[808    0]
 [ 97    0]]
```

```
0.5288379106188197
0.8784530386740331
[[795    0]
 [110    0]]
```

SMOTE

- By using smote to balance our data we noticed that the results were very overfitted. As such we chose to not use it and use the pure data.

- Overfitted Data:

```
model_rf = RandomForestRegressor()

model_rf.fit(x_train, y_train)

print(model_rf.score(x_train, y_train))
print(metrics.accuracy_score(y_test,y_pred))
print(metrics.confusion_matrix(y_test,y_pred))
```

```
0.9678635582957248
0.815625
[[658 151]
 [144 647]]
```

AGE

:How old the recipient is

```
      age
0      11
1      14
2      16
3      11
4      40
...   ...
4516   14
4517   38
4518   38
4519    9
4520   25

[4521 rows x 1 columns]
0.8855088495575221
0.881767955801105
[[798   0]
 [107   0]]
0.8855088495575221
0.881767955801105
[[798   0]
 [107   0]]
0.05132274521215552
0.881767955801105
[[798   0]
 [107   0]]
```

MARITAL STATUS

:Is the recipient Married, divorced, or single?

```
      marital
0           1
1           1
2           0
3           1
4           1
...       ...
4516        1
4517        1
4518        1
4519        1
4520        0

[4521 rows x 1 columns]
0.8860619469026548
0.8795580110497238
[[796   0]
 [109   0]]
0.8860619469026548
0.8795580110497238
[[796   0]
 [109   0]]
0.005183965744335217
0.8795580110497238
[[796   0]
 [109   0]]
```

EDUCATION

:Highest degree of education of the recipient

```
      education
0             1
1             2
2             3
3             3
4             2
...         ...
4516          2
4517          3
4518          2
4519          2
4520          3

[4521 rows x 1 columns]
0.8877212389380531
0.8729281767955801
[[790   0]
 [115   0]]
0.8877212389380531
0.8729281767955801
[[790   0]
 [115   0]]
0.0015446669610111874
0.8729281767955801
[[790   0]
 [115   0]]
```

JOB

:The recipient's occupation

```
      job
0      10
1       7
2       4
3       4
4       1
...   ...
4516    7
4517    6
4518    9
4519    1
4520    2

[4521 rows x 1 columns]
```

```
lr = LogisticRegression(max_iter=500)

model_lr=lr.fit(x_train,y_train)
y_pred = model_lr.predict(x_test)

print(lr.score(x_train,y_train))
print(metrics.accuracy_score(y_test,y_pred))
print(metrics.confusion_matrix(y_test,y_pred))
```

```
0.8857853982300885
0.8806629834254144
[[797   0]
 [108   0]]
```

```
mlp = MLPClassifier( activation = "logistic", hidden_layer_sizes = (100,100),max_iter=500)
model_mlp = mlp.fit(x_train,y_train)
y_pred = model_mlp.predict(x_test)

print(model_mlp.score(x_train,y_train))
print(metrics.accuracy_score(y_test,y_pred))
print(metrics.confusion_matrix(y_test,y_pred))
```

```
0.8857853982300885
0.8806629834254144
[[797   0]
 [108   0]]
```

```
model_rf = RandomForestRegressor( )

model_rf.fit(x_train, y_train)

print(model_rf.score(x_train, y_train))
print(metrics.accuracy_score(y_test,y_pred))
print(metrics.confusion_matrix(y_test,y_pred))
```

```
0.015575093479746283
0.8806629834254144
[[797   0]
 [108   0]]
```

BALANCE

:How much Money the account has

```
      balance
0        1475
1        2030
2        1303
3        1352
4         274
...       ...
4516      119
4517        0
4518      558
4519     1187
4520     1186

[4521 rows x 1 columns]
0.8863384955752213
0.8784530386740331
[[795    0]
 [110    0]]
0.8863384955752213
0.8784530386740331
[[795    0]
 [110    0]]
0.5288379106188197
0.8784530386740331
[[795    0]
 [110    0]]
```

DEFAULT

:Are any outstanding balances due on credit / loans?

```
      default
0           0
1           0
2           0
3           0
4           0
...       ...
4516        0
4517        1
4518        0
4519        0
4520        0

[4521 rows x 1 columns]
0.8827433628318584
0.8928176795580111
[[808    0]
 [ 97    0]]
0.8827433628318584
0.8928176795580111
[[808    0]
 [ 97    0]]
4.587657819854485e-06
0.8928176795580111
[[808    0]
 [ 97    0]]
```

HOUSING

:Does the recipient have any housing loans due?

```
      housing
0           0
1           1
2           1
3           1
4           1
...       ...
4516        1
4517        1
4518        0
4519        0
4520        1

[4521 rows x 1 columns]
0.8813606194690266
0.8983425414364641
[[813   0]
 [ 92   0]]
0.8813606194690266
0.8983425414364641
[[813   0]
 [ 92   0]]
0.010128776412640206
0.8983425414364641
[[813   0]
 [ 92   0]]
```

LOAN

:Amount in loans the customeer has due to the bank

```
      loan
0        0
1        1
2        0
3        1
4        0
...    ...
4516     0
4517     1
4518     0
4519     0
4520     1

[4521 rows x 1 columns]
0.8832964601769911
0.8906077348066298
[[806   0]
 [ 99   0]]
0.8832964601769911
0.8906077348066298
[[806   0]
 [ 99   0]]
0.005026846137220375
0.8906077348066298
[[806   0]
 [ 99   0]]
```

CONTACT

:How the recipient was contacted (cellular, telephone, unknown, etc)

```
      contact
0           1
1           1
2           1
3           0
4           0
...       ...
4516        1
4517        0
4518        1
4519        1
4520        1

[4521 rows x 1 columns]
0.884679203539823
0.8850828729281768
[[801    0]
 [104    0]]
0.884679203539823
0.8850828729281768
[[801    0]
 [104    0]]
0.019117359333625794
0.8850828729281768
[[801    0]
 [104    0]]
```

P-DAYS

:number of days that passed by after the client was last contacted from a previous campaign

```
      pdays
0         0
1       228
2       219
3         0
4         0
...     ...
4516      0
4517      0
4518      0
4519    140
4520    161

[4521 rows x 1 columns
0.8855088495575221
0.881767955801105
[[798    0]
 [107    0]]
0.8855088495575221
0.881767955801105
[[798    0]
 [107    0]]
0.1838886837309962
0.881767955801105
[[798    0]
 [107    0]]
```

P-OUTCOME

:outcome of the previous marketing campaign

```
      poutcome
0            3
1            0
2            0
3            3
4            3
...        ...
4516         3
4517         3
4518         3
4519         1
4520         1

[4521 rows x 1 columns]
0.8841261061946902
0.887292817679558
[[803    0]
 [102    0]]
0.8841261061946902
0.887292817679558
[[803    0]
 [102    0]]
0.08334790271951575
0.887292817679558
[[803    0]
 [102    0]]
```

RANKING

```
from sklearn.feature_selection import RFE

rfe=RFE(estimator = LogisticRegression(), n_features_to_select=1)
rfe_fit = rfe.fit(x_train,y_train)

print(rfe_fit.ranking_)
```

```
[ 7 14  4 10 11  9  2  8 15 13  1  3  6 12  5]
```

```
print(x.columns)
```

```
Index(['age', 'job', 'marital', 'education', 'default', 'balance', 'loan',
       'contact', 'day', 'month', 'duration', 'campaign', 'pdays', 'previous',
       'poutcome'],
      dtype='object')
```

- 1. Duration
- 2. Loan
- 3. Campaign
- 4. Marital Status
- 5. P-Outcome
- 6. pDays
- 7. Age
- 8. Contact
- 9. Balance
- 10. Education
- 11. Default
- 12. Previous
- 13. Month
- 14. Job
- 15. Day

CONCLUSION

With the help of the model algorithm, the predictor has almost a 90% accuracy. This can be applied directly into financial institutions telemarketing, saving both time and money for not only the institution but also the customers, as it provides a direct correlation between the success rate of the marketing call, and the background factors to check before calling.