Bank Subscription

By: Tanav Dandekar and Carter Delargy

Problem of the Dataset

- Individual's relationship with the bank (contact, time since last contact, number of contacts before this campaign, etc.)
- Relationship with whether or not the individual subscribed for a term deposit.

	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

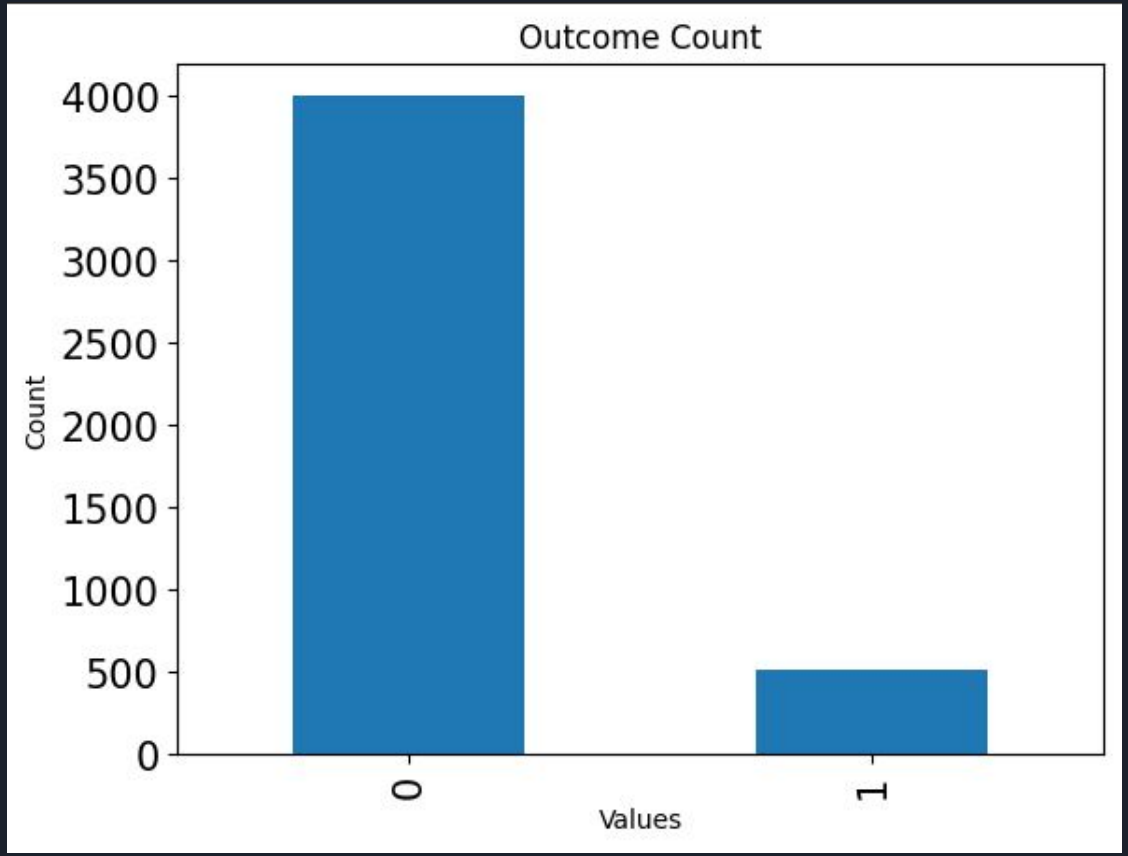
4521 rows × 17 columns

Preprocessing

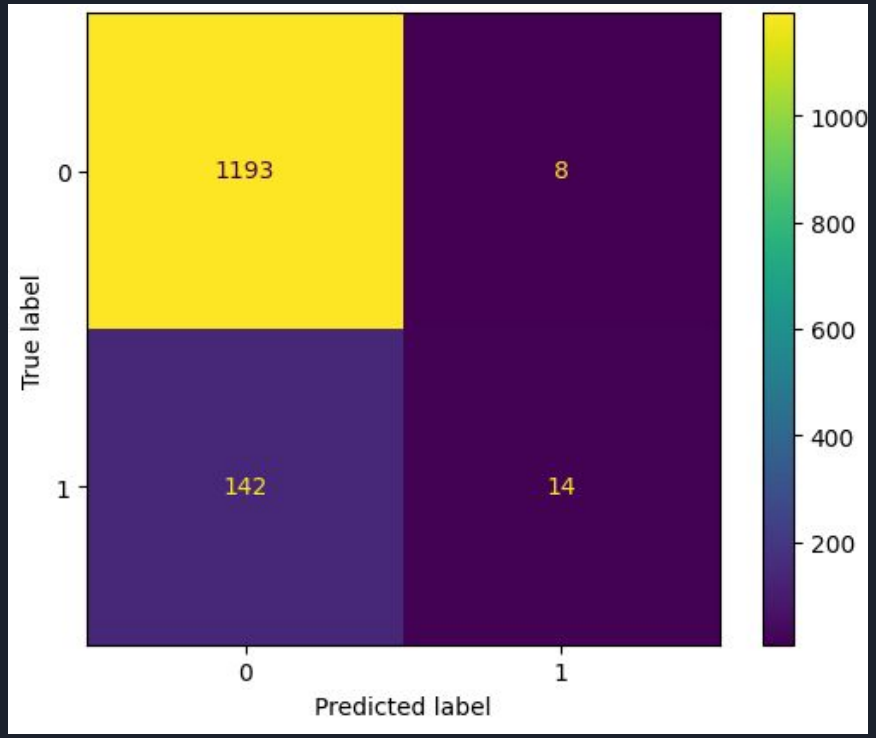
- One hot encoding
- Some mapping

```
def full_preprocessing():
    bank_df = pd.read_csv('bank.csv', delimiter=';')
    jobs = pd.get_dummies(bank_df['job'])
    bank_df = pd.concat([bank_df, jobs], axis=1)
    bank_df = bank_df.drop(['job'], axis=1)

    marital_status = pd.get_dummies(bank_df['marital'])
    bank_df = pd.concat([bank_df, marital_status], axis=1)
    bank_df = bank_df.drop(['marital'], axis=1)

    education = pd.get_dummies(bank_df['education'])
    bank_df = pd.concat([bank_df, education], axis=1)
    bank_df = bank_df.drop(['education'], axis=1)

    bank_df['default'] = bank_df['default'].map({'no':0, 'yes':1})
    bank_df['housing'] = bank_df['housing'].map({'no':0, 'yes':1})
    bank_df['loan'] = bank_df['loan'].map({'no':0, 'yes':1})

    contact = pd.get_dummies(bank_df['contact'])
    bank_df = pd.concat([bank_df, contact], axis=1)
    bank_df = bank_df.drop(['contact'], axis=1)

    bank_df['month'] = bank_df['month'].map({'jan':1, 'feb':2,
                                             'mar':3, 'apr':4,
                                             'may':5,'jun':6,
                                             'jul':7, 'aug':8,
                                             'sep':9,'oct':10,
                                             'nov':11,'dec':12})

    poutcome = pd.get_dummies(bank_df['poutcome'])
    bank_df = pd.concat([bank_df, poutcome], axis=1)
    bank_df = bank_df.drop(['poutcome'], axis=1)

    bank_df['y'] = bank_df['y'].map({'no':0, 'yes':1})

    return bank_df
```

Yes vs. No

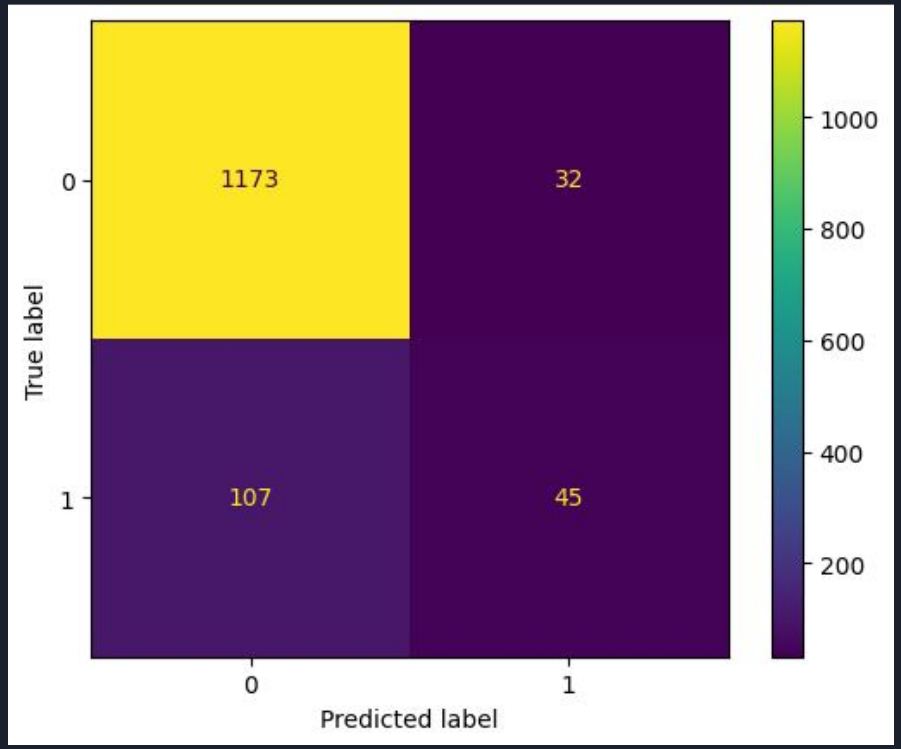

Outcome Count
4000
3500
3000
2500
2000
1500
1000
500
0
Count
0
1
Values

KNN



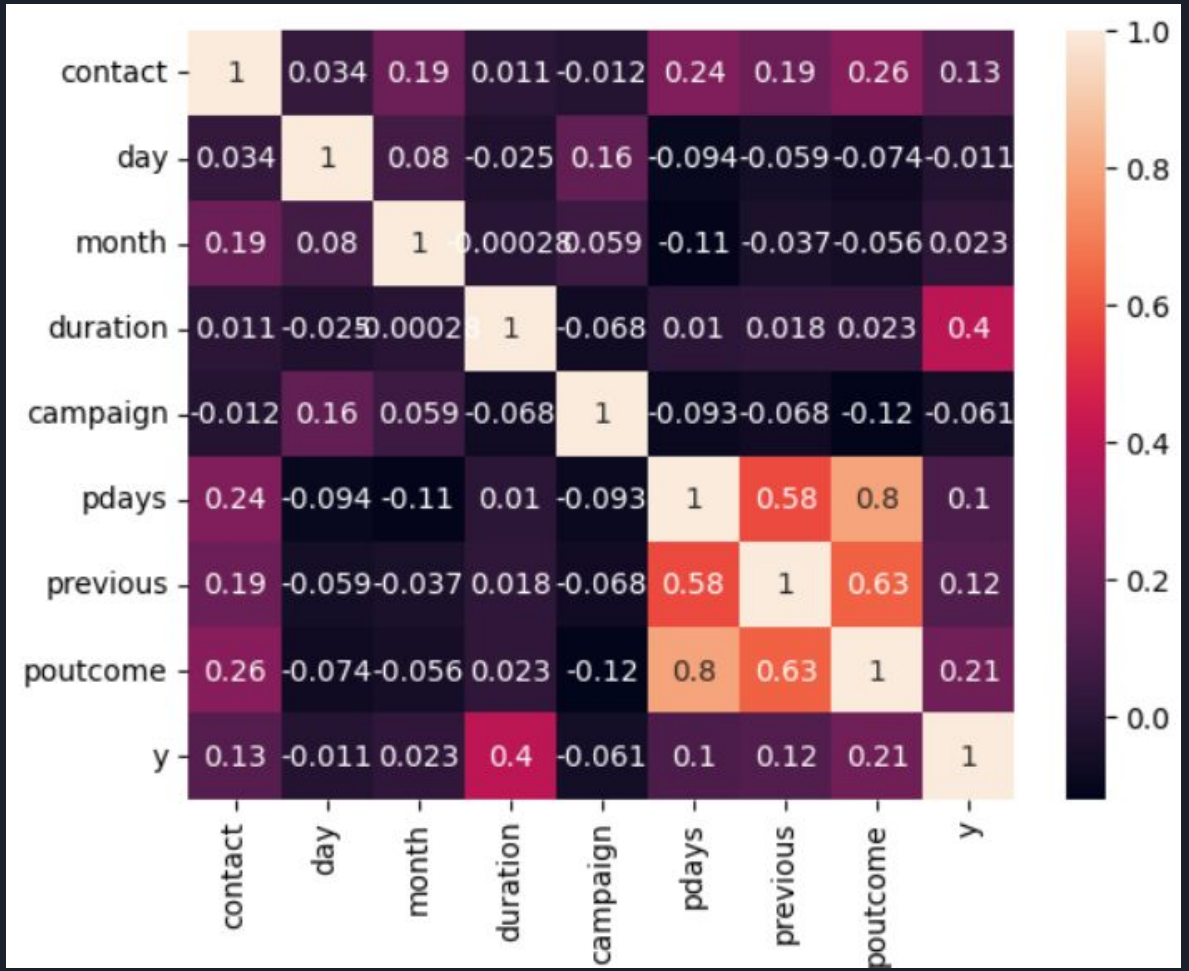
	precision	recall	f1-score	support
0	0.89	0.99	0.94	1201
1	0.64	0.09	0.16	156
accuracy			0.89	1357
macro avg	0.76	0.54	0.55	1357
weighted avg	0.86	0.89	0.85	1357

Logistic Regression

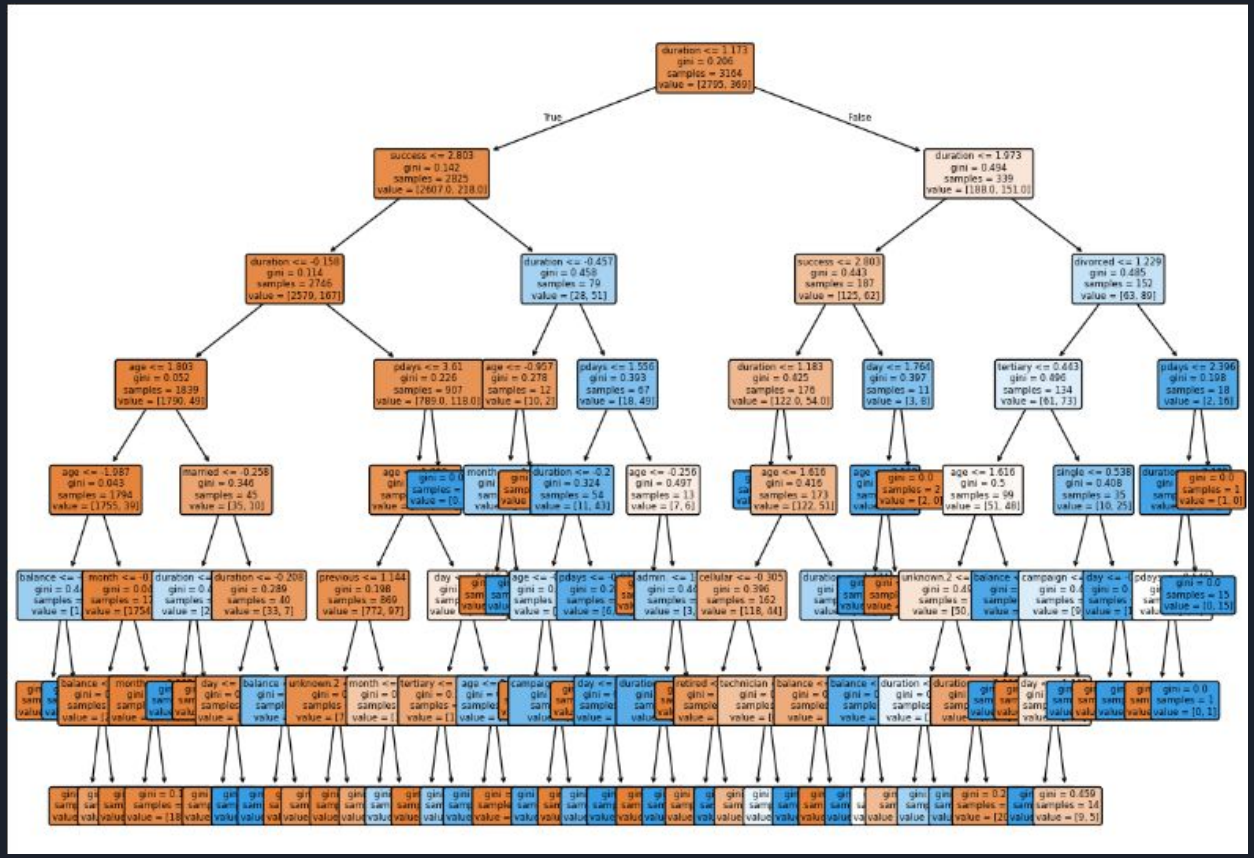


	precision	recall	f1-score	support
0	0.92	0.97	0.94	1205
1	0.58	0.30	0.39	152
accuracy			0.90	1357
macro avg	0.75	0.63	0.67	1357
weighted avg	0.88	0.90	0.88	1357

Heatmap


contact 1 0.034 0.19 0.011 -0.012 0.24 0.19 0.26 0.13
day 0.034 1 0.08 -0.025 0.16 -0.094 -0.059 -0.074 -0.011
month 0.19 0.08 1 0.00028 0.059 -0.11 -0.037 -0.056 0.023
duration 0.011 -0.025 0.0002 1 -0.068 0.01 0.018 0.023 0.4
campaign -0.012 0.16 0.059 -0.068 1 -0.093 -0.068 -0.12 -0.061
pdays 0.24 -0.094 -0.11 0.01 -0.093 1 0.58 0.8 0.1
previous 0.19 -0.059 -0.037 0.018 -0.068 0.58 1 0.63 0.12
poutcome 0.26 -0.074 -0.056 0.023 -0.12 0.8 0.63 1 0.21
y 0.13 -0.011 0.023 0.4 -0.061 0.1 0.12 0.21 1
contact
day
month
duration
campaign
pdays
previous
poutcome
y
1.0
0.8
0.6
0.4
0.2
0.0

Tree Model


duration <= 1.173
gini = 0.206
samples = 3164
value = [2705, 369]
True
False
success <= 2.803
gini = 0.142
samples = 2825
value = [2607.0, 218.0]
duration <= 1.973
gini = 0.494
samples = 339
value = [188.0, 151.0]
duration <= -0.158
gini = 0.114
samples = 2746
value = [2579, 167]
duration <= -0.457
gini = 0.458
samples = 79
value = [28, 51]
success <= 2.803
gini = 0.443
samples = 187
value = [125, 62]
divorced <= 1.229
gini = 0.485
samples = 152
value = [63, 89]
age <= 1.803
gini = 0.052
samples = 1839
value = [1790, 49]
pdays <= 3.61
gini = 0.226
samples = 907
value = [789.0, 118.0]
age <= -0.907
gini = 0.278
samples = 12
value = [10, 2]
pdays <= 1.556
gini = 0.393
samples = 67
value = [18, 49]
duration <= 1.183
gini = 0.425
samples = 176
value = [122.0, 54.0]
day <= 1.764
gini = 0.397
samples = 11
value = [3, 8]
tertiary <= 0.443
gini = 0.496
samples = 134
value = [61, 73]
pdays <= 2.396
gini = 0.198
samples = 18
value = [2, 16]
age <= -1.087
gini = 0.043
samples = 1794
value = [1755, 39]
married <= -0.258
gini = 0.346
samples = 45
value = [35, 10]
duration <= -0.2
gini = 0.324
samples = 58
value = [11, 47]
age <= -0.256
gini = 0.497
samples = 13
value = [7, 6]
age <= 1.616
gini = 0.416
samples = 173
value = [122, 51]
age <= 1.616
gini = 0.5
samples = 99
value = [51, 48]
single <= 0.538
gini = 0.408
samples = 35
value = [10, 25]
previous <= 1.144
gini = 0.198
samples = 869
value = [772, 97]
duration <= -0.208
gini = 0.289
samples = 40
value = [33, 7]
cellular <= -0.305
gini = 0.396
samples = 162
value = [118, 44]
unknown.2 <=
gini = 0.46
samples =
value = [50,
campaign <=
gini = 0.4
samples =
value = [9,
gini = 0.0
samples = 15
value = [0, 15]

PostgreSQL Database

```
bank_df_people = bank_df[['age','job','marital','education','default','balance','housing','loan']]
bank_df_relationship = bank_df[['contact','day','month','duration','campaign','pdays','previous','poutcome','y']]
bank_df_y = bank_df[['y','age']]
```

```
from sqlalchemy import create_engine
```

```
engine = create_engine('postgresql://postgres:password@localhost:5432/final_project')
```

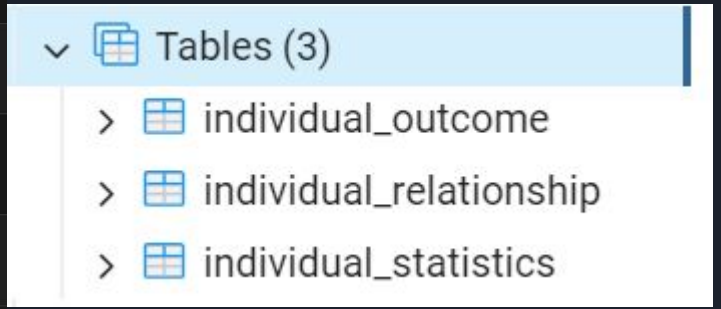
```
bank_df_people.to_sql("individual_statistics", con=engine, if_exists="replace",
                      index=False)
```

```
521
```

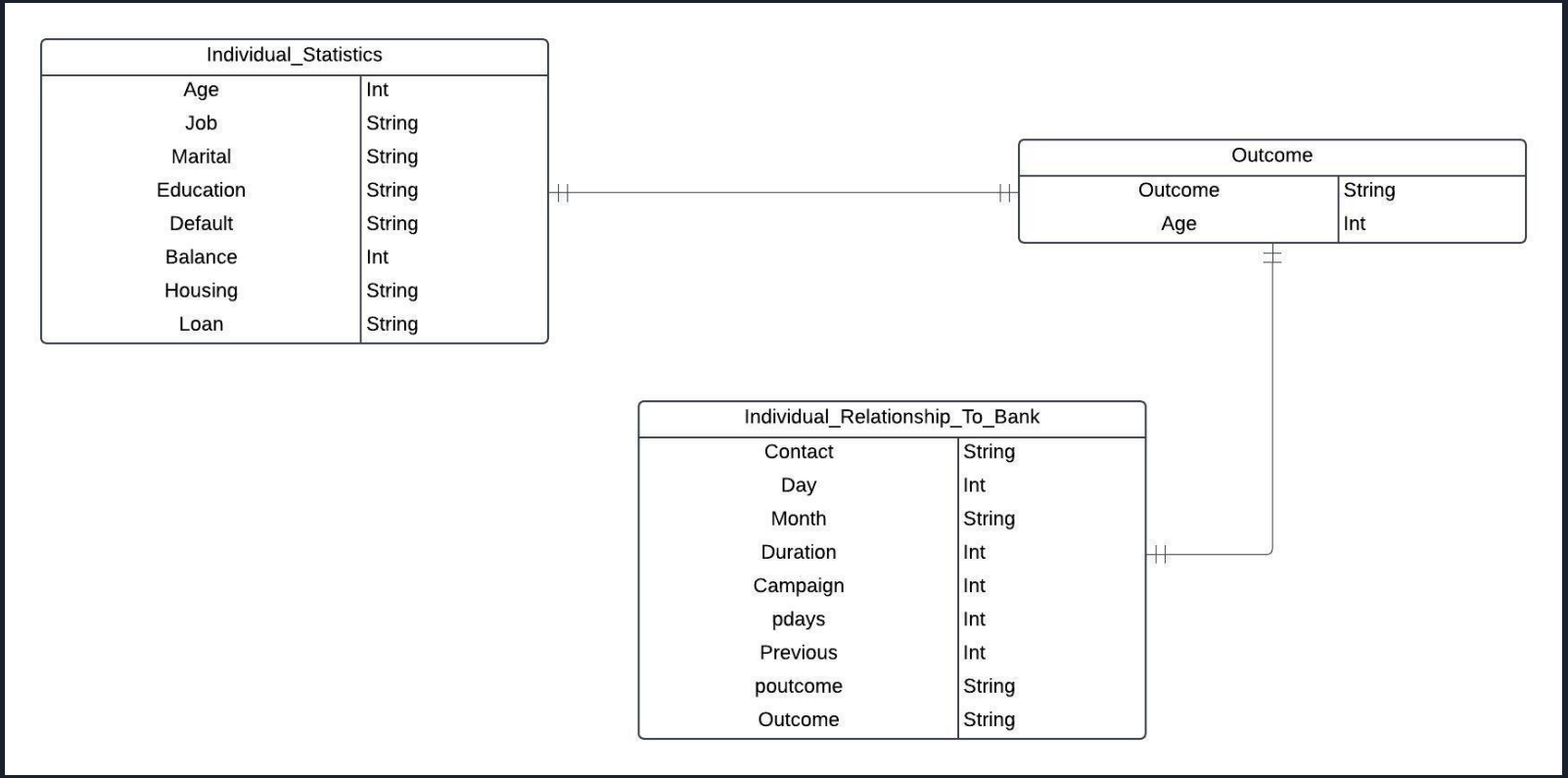
```
bank_df_relationship.to_sql("individual_relationship", con=engine, if_exists="replace",
                            index=False)
```

```
521
```

```
bank_df_y.to_sql("individual_outcome", con=engine, if_exists="replace",
                 index=False)
```



Data Model


Individual_Statistics
Age
Int
Job
String
Marital
String
Education
String
Default
String
Balance
Int
Housing
String
Loan
String
Outcome
Outcome
String
Age
Int
Individual_Relationship_To_Bank
Contact
String
Day
Int
Month
String
Duration
Int
Campaign
Int
pdays
Int
Previous
Int
poutcome
String
Outcome
String

Individual Outcome

```
1  select * from individual_outcome
```

	y text	age bigint
1	no	30
2	no	33
3	no	35
4	no	30
5	no	59
6	no	35
7	no	36
8	no	39
9	no	41
10	no	43
11	no	39
12	no	43

Total rows: 1000 of 4521

Individual Relationship

```
1  select * from individual_relationship
```

	contact text	day bigint	month text	duration bigint	campaign bigint	pdays bigint	previous bigint	poutcome text	y text
1	cellular	19	oct	79	1	-1	0	unknown	no
2	cellular	11	may	220	1	339	4	failure	no
3	cellular	16	apr	185	1	330	1	failure	no
4	unknown	3	jun	199	4	-1	0	unknown	no
5	unknown	5	may	226	1	-1	0	unknown	no
6	cellular	23	feb	141	2	176	3	failure	no
7	cellular	14	may	341	1	330	2	other	no
8	cellular	6	may	151	2	-1	0	unknown	no
9	unknown	14	may	57	2	-1	0	unknown	no
10	cellular	17	apr	313	1	147	2	failure	no
11	unknown	20	may	273	1	-1	0	unknown	no
12	cellular	17	apr	113	2	-1	0	unknown	no

Total rows: 1000 of 4521 Query complete 00:00:00.193 Ln 1, Col 38

Individual Statistics

```
1   select * from individual_statistics
```

	age bigint	job text	marital text	education text	default text	balance bigint	housing text	loan text
1	30	unemployed	married	primary	no	1787	no	no
2	33	services	married	secondary	no	4789	yes	yes
3	35	management	single	tertiary	no	1350	yes	no
4	30	management	married	tertiary	no	1476	yes	yes
5	59	blue-collar	married	secondary	no	0	yes	no
6	35	management	single	tertiary	no	747	no	no
7	36	self-employed	married	tertiary	no	307	yes	no
8	39	technician	married	secondary	no	147	yes	no
9	41	entrepreneur	married	tertiary	no	221	yes	no
10	43	services	married	primary	no	-88	yes	yes
11	39	services	married	secondary	no	9374	yes	no
12	43	admin.	married	secondary	no	264	yes	no
13	36	technician	married	tertiary	no	1109	no	no

Total rows: 1000 of 4521 Query complete 00:00:00.166 Ln 1, Col 36