FiNas

Banking Dataset Analysis and Classification

EDA

1

Data Preprocessing

2

Model Development

3

Model Evaluation

4

Predict term deposit subscriptions

5

Thursday	Friday	Saturday	Sunday	Monday	Tuesday	Wednesday
EDA &Data Preprocessing						
			Model Development			
				Model Evaluation		
					prepare presentation	

EDA
&
Data Preprocessing

Looking at the DataFrame

	Unnamed: 0	id	age	job	marital	education	default	balance	housing	loan	contact	day	month	duration	campaign	pd
0	56963	31963	47	blue-collar	married	basic.9y	no	NaN	no	yes	cellular	apr	fri	583	2	
1	31753	21378	48	management	divorced	tertiary	no	351.0	yes	no	cellular	7	apr	725	3	
2	60854	17084	38	technician	single	high.school	no	NaN	no	no	cellular	aug	wed	74	2	
3	34207	81693	50	management	divorced	tertiary	no	1270.0	yes	no	cellular	4	may	24	3	
4	73066	63978	54	self-employed	married	high.school	no	NaN	yes	yes	cellular	aug	thu	904	3	

Dropp some columns:

id: not useful and high cardinality.
Unnamed: 0 :not useful and high cardinality

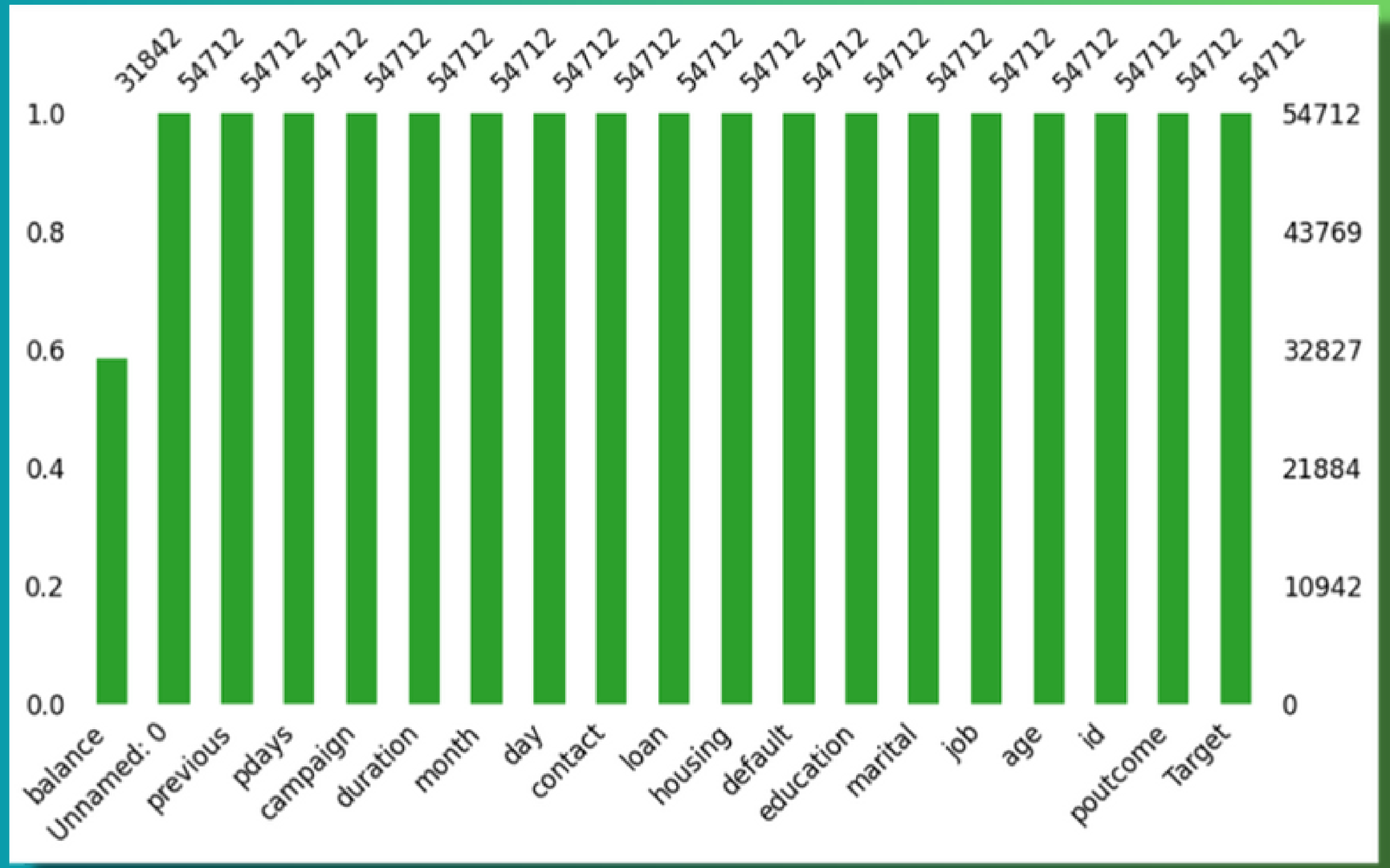
Checking the number of duplicated rows after removing some columns

```
Number of duplicated row on train data:  3
```

Describe numerical data

	age	balance	duration	campaign	pdays	previous
count	54712.000000	31842.000000	54712.000000	54712.000000	54712.000000	54712.000000
mean	40.580147	1368.024590	257.282260	2.679961	425.022829	0.407351
std	10.544504	3057.847866	255.270335	2.966822	476.813012	1.520253
min	17.000000	-8019.000000	0.000000	1.000000	-1.000000	0.000000
25%	32.000000	73.000000	103.000000	1.000000	-1.000000	0.000000
50%	39.000000	449.000000	180.000000	2.000000	88.000000	0.000000
75%	48.000000	1435.000000	318.000000	3.000000	999.000000	0.000000
max	98.000000	98417.000000	4199.000000	58.000000	999.000000	58.000000

Visualizing the null values for each column



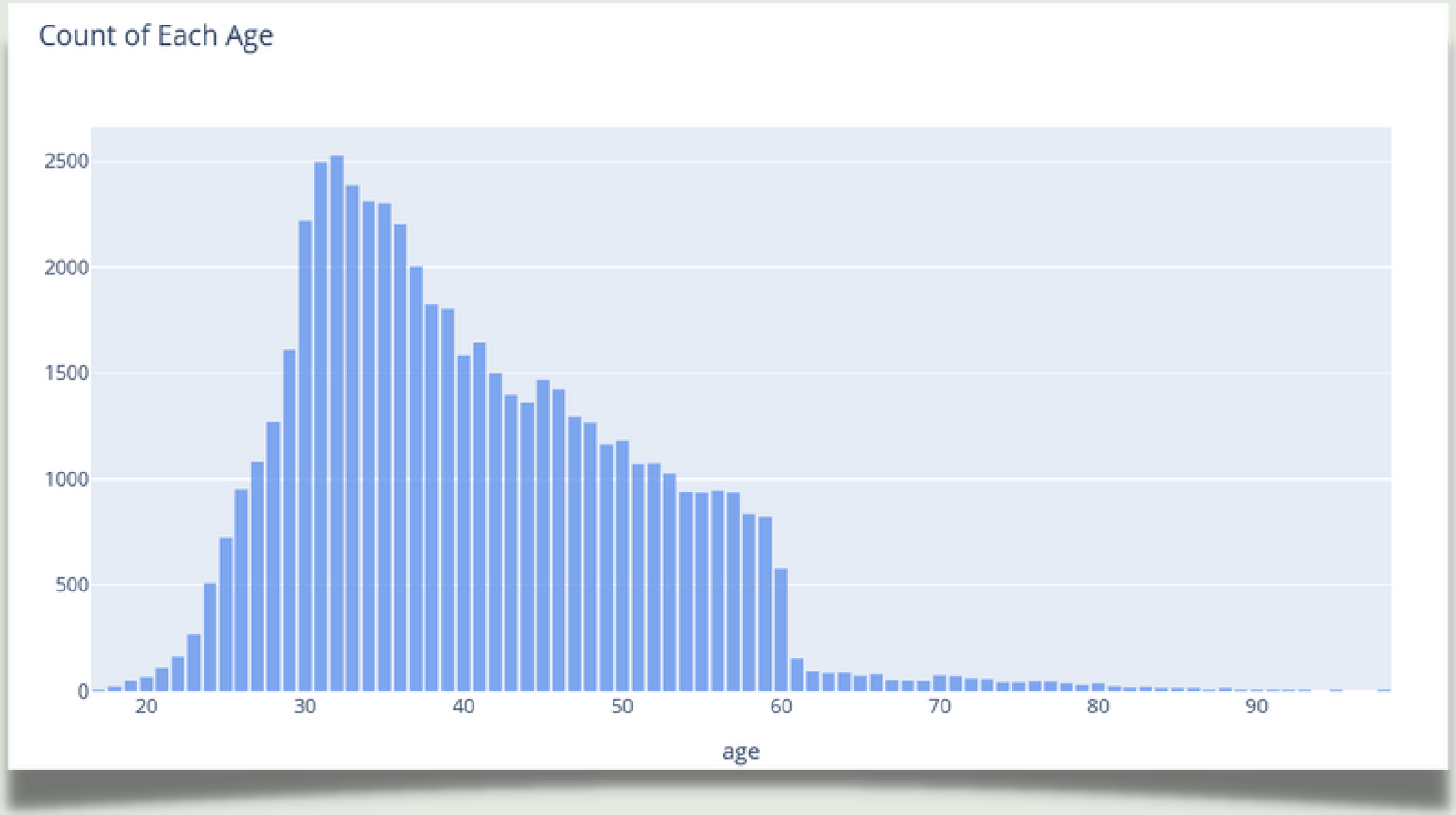
Correlation Heatmap

	age	job	balance	duration	campaign	pdays	previous
age	1	-0.014	0.11	-0.0091	0.0096	-0.012	-0.0041
job	-0.014	1	0.015	-0.0011	0.0064	0.0044	0.013
balance	0.11	0.015	1	0.018	-0.015	0.005	0.02
duration	-0.0091	-0.0011	0.018	1	-0.081	-0.00051	0.0017
campaign	0.0096	0.0064	-0.015	-0.081	1	-0.058	-0.033
pdays	-0.012	0.0044	0.005	-0.00051	-0.058	1	0.53
previous	-0.0041	0.013	0.02	0.0017	-0.033	0.53	1

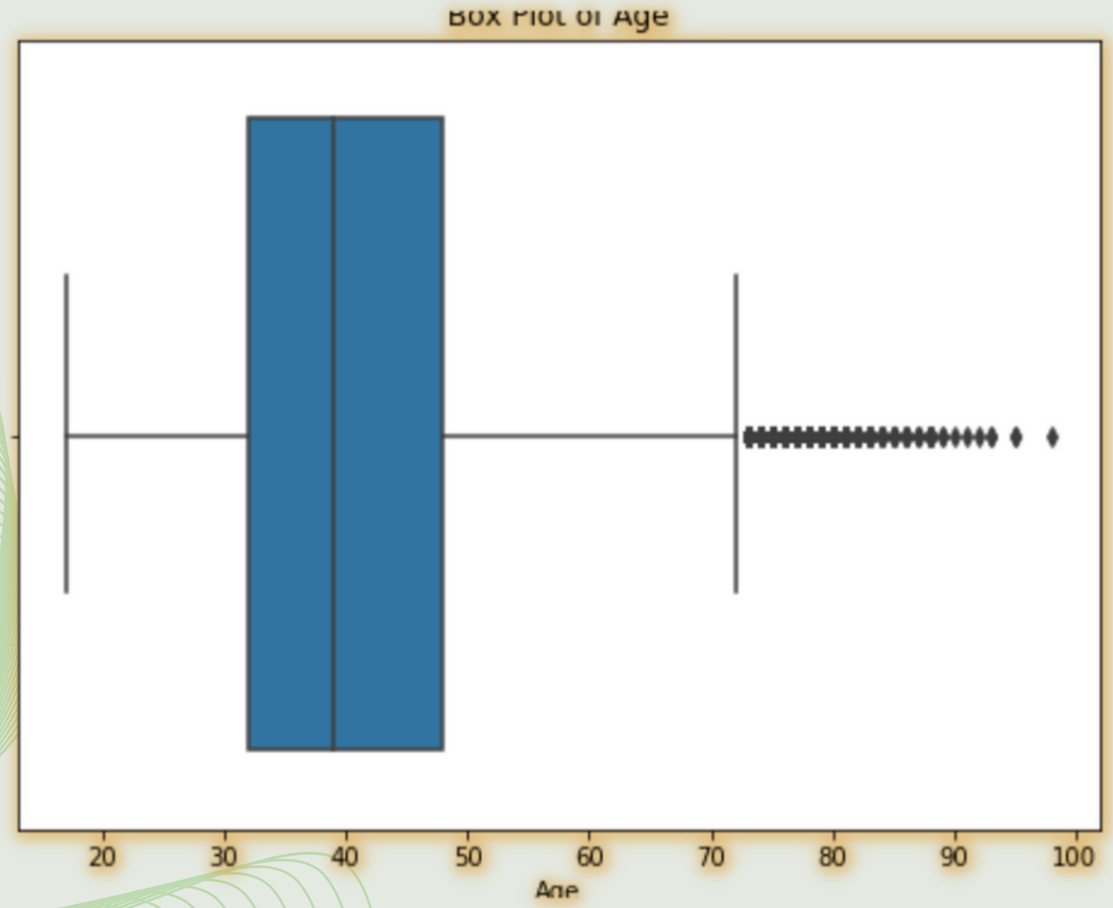
1.0
0.8
0.6
0.4
0.2
0.0

• Age Column

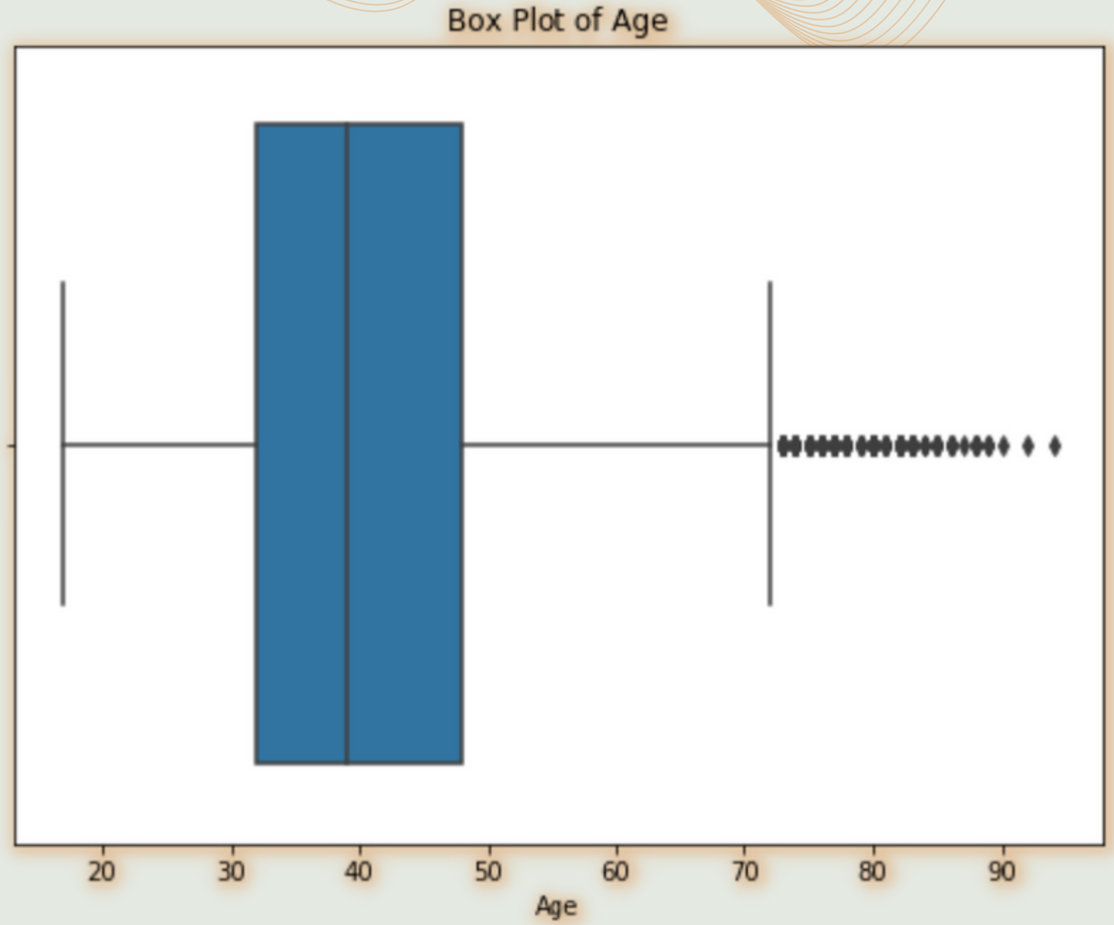
Filter the DataFrame to get only rows where the 'age' column value is less than 94



• Age Column



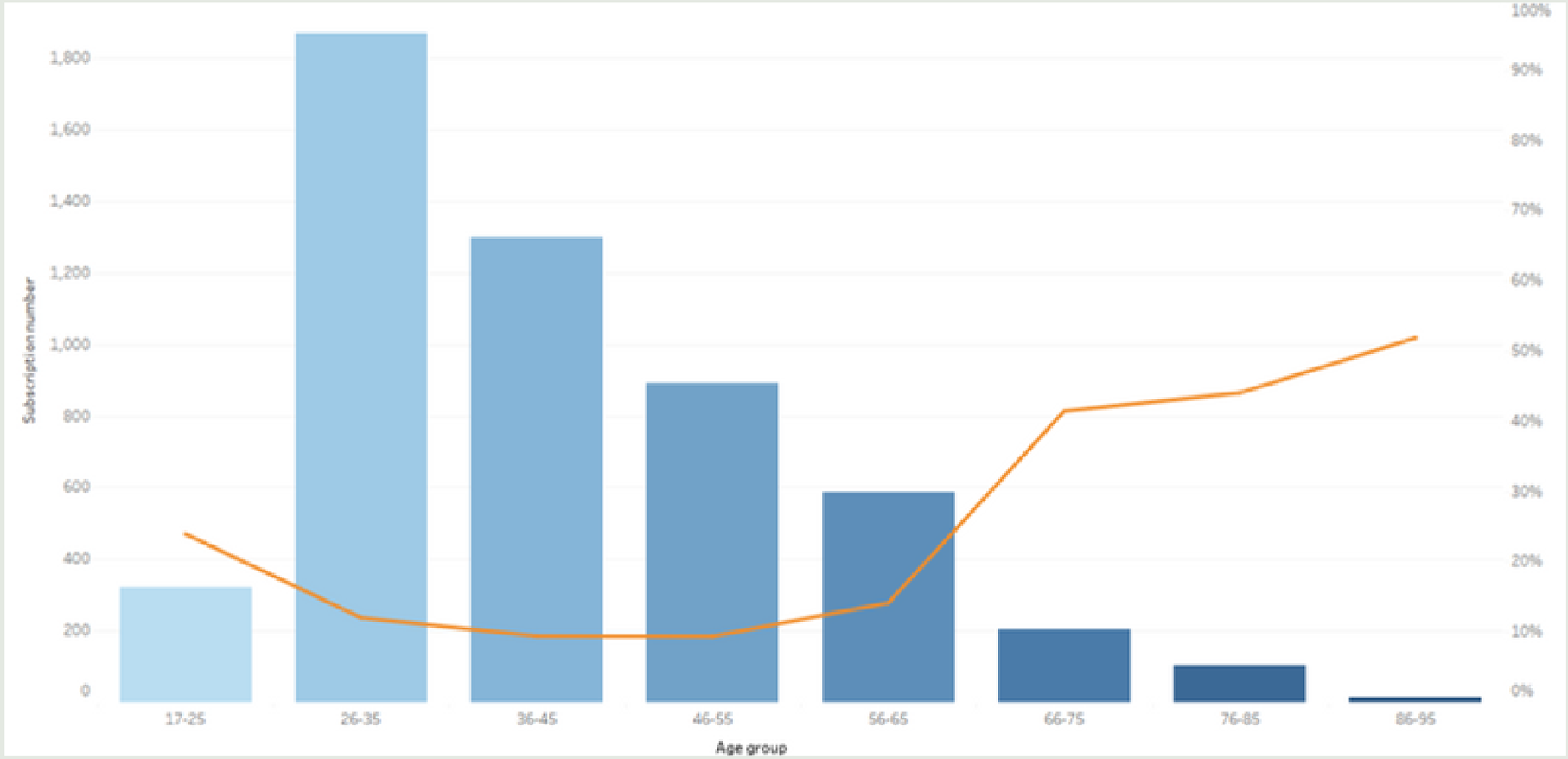
train:
min: 17
max: 98



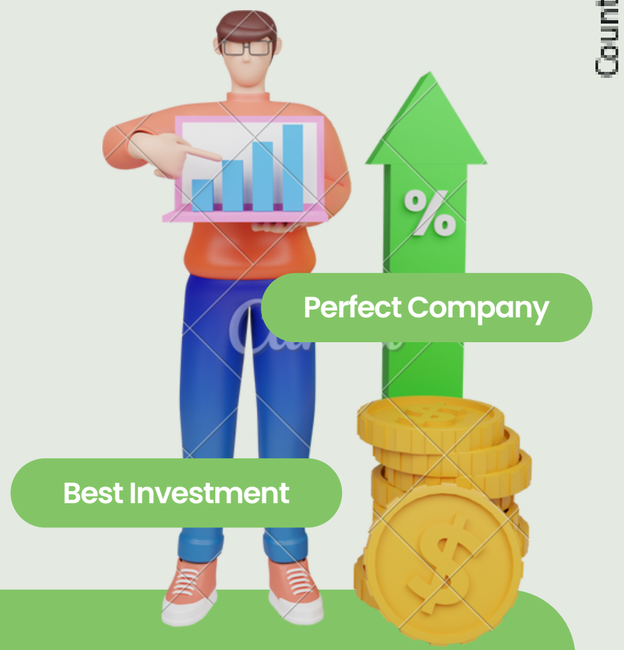
test:
min: 17
max: 94

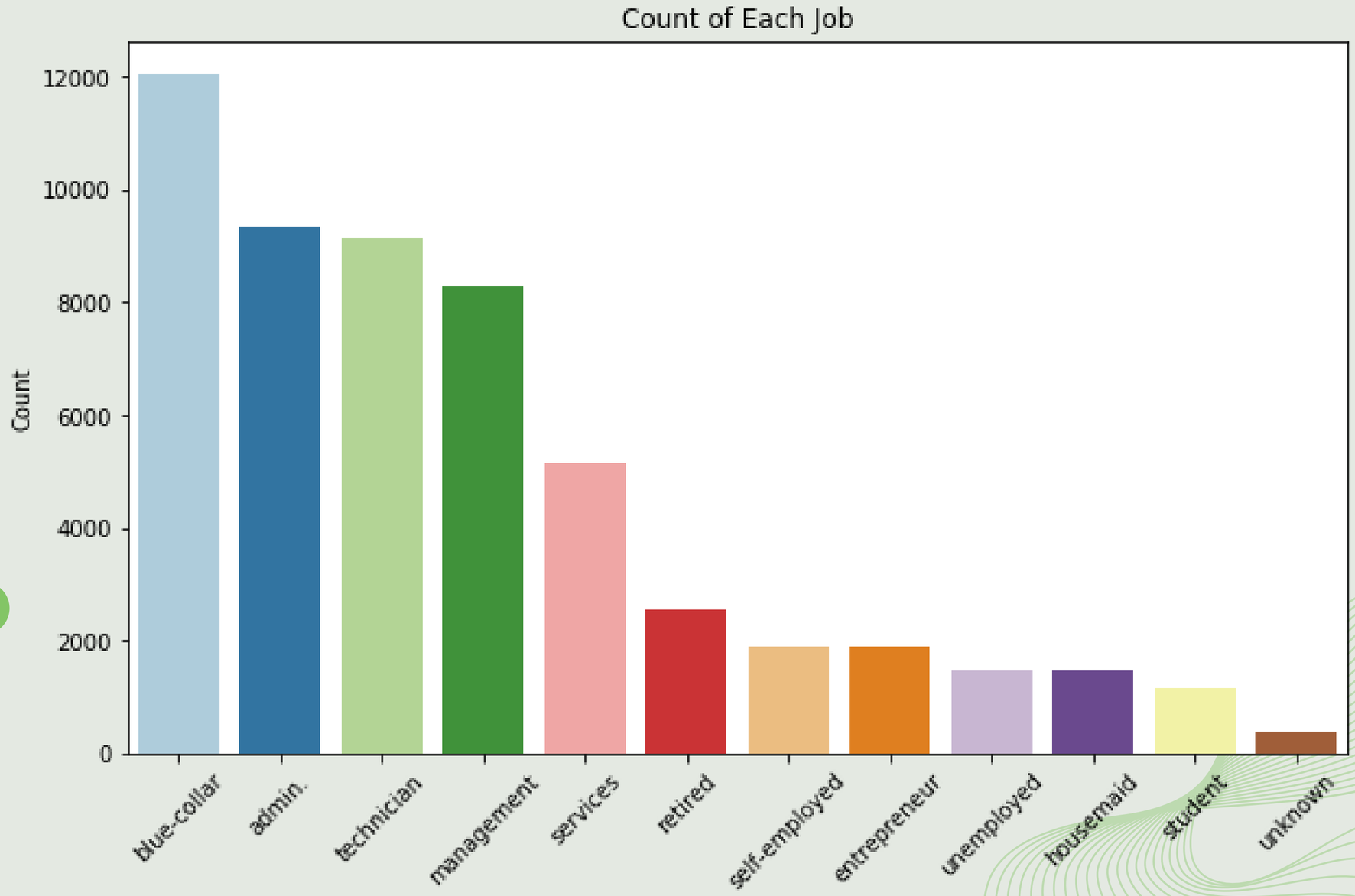
• Age Column

The age groups above 56 have the largest number of subscriptions

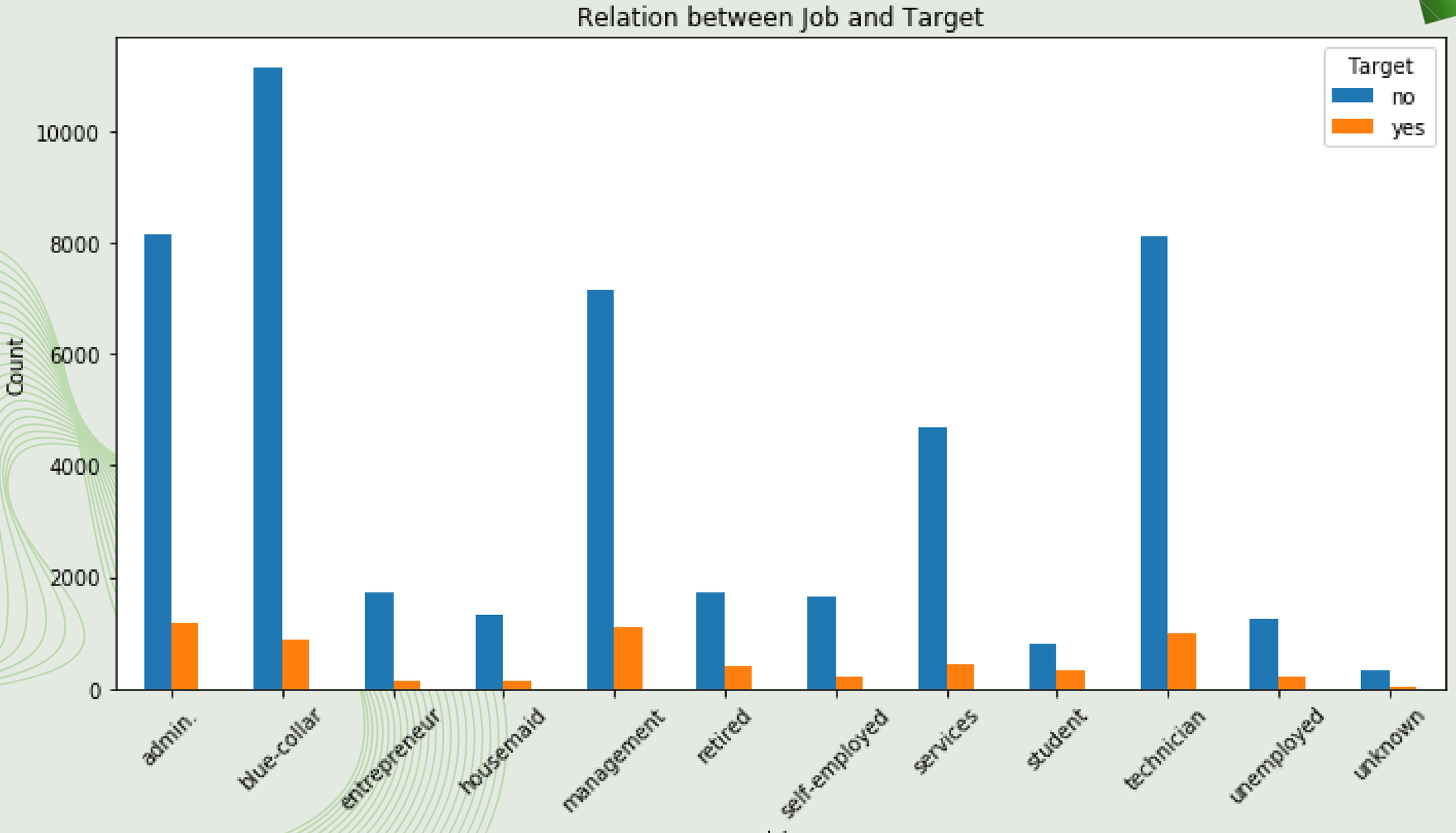


Job Column


Perfect Company
Best Investment

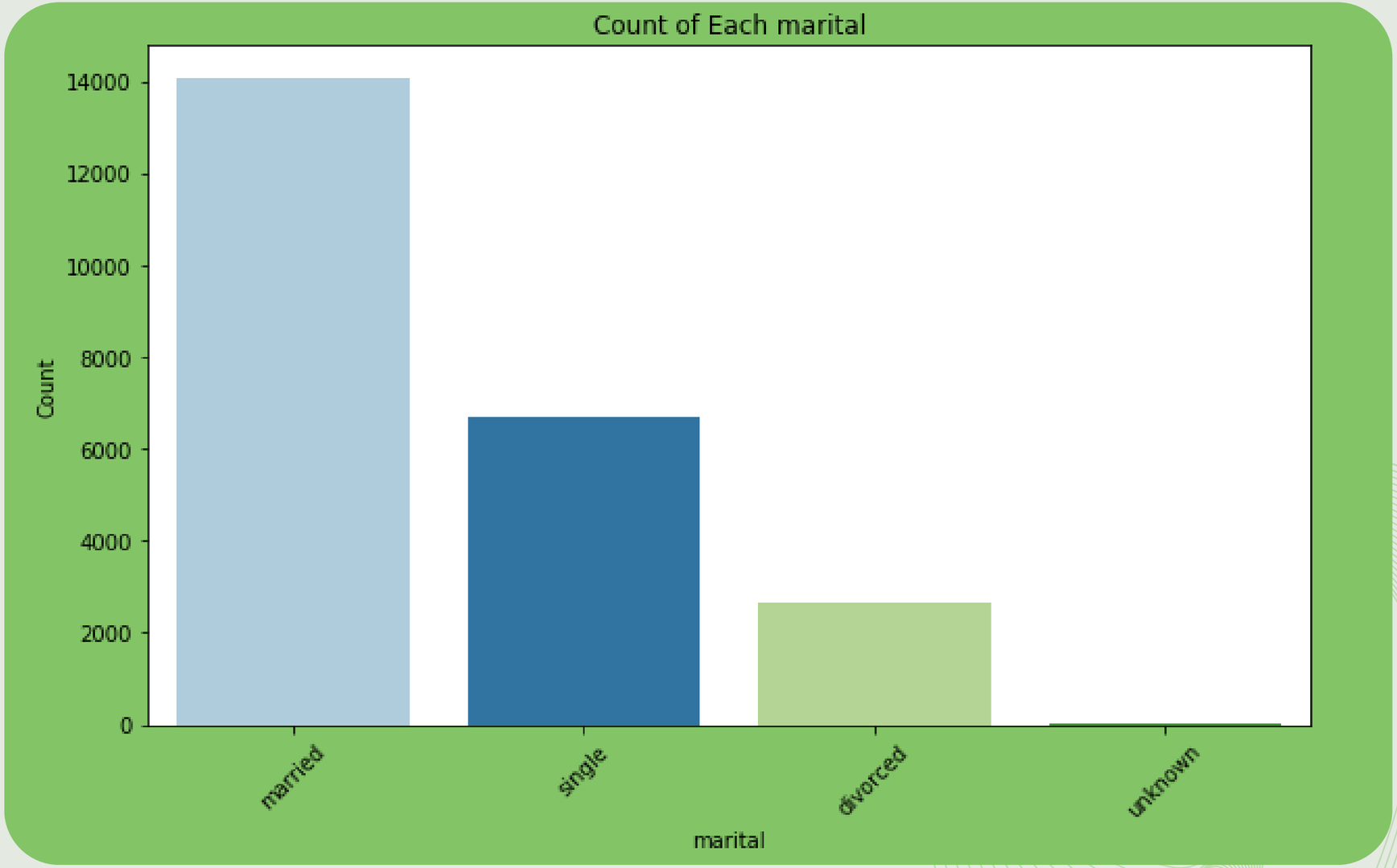

Count of Each Job
Count
12000
10000
8000
6000
4000
2000
0
blue-collar
admin.
technician
management
services
retired
self-employed
entrepreneur
unemployed
housemaid
student
unknown

• Job Column

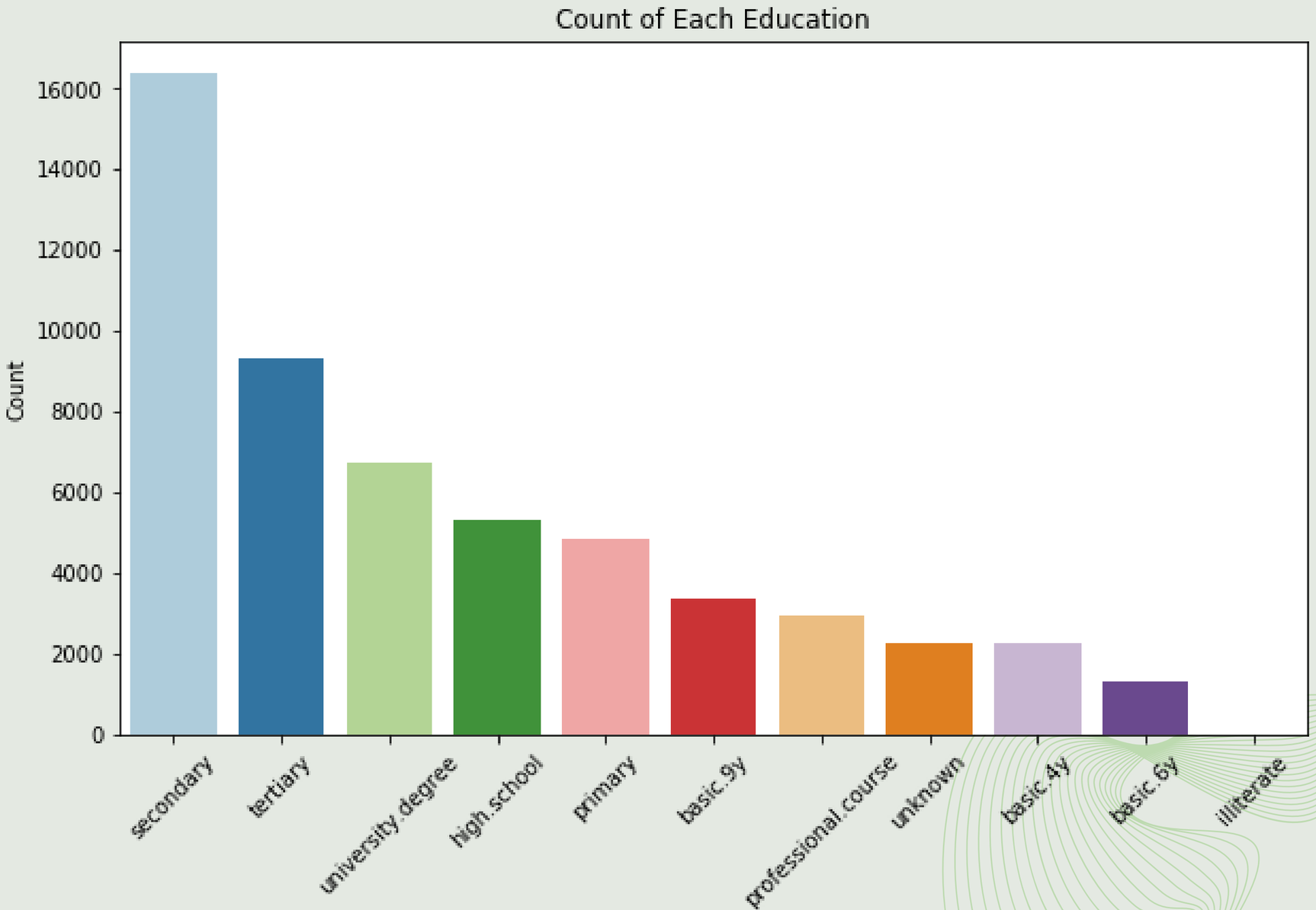

Relation between Job and Target
Target
no
yes
Count
10000
8000
6000
4000
2000
0
admin.
blue-collar
entrepreneur
housemaid
management
retired
self-employed
services
student
technician
unemployed
unknown

Marital Column

- Fill 'unknown' marital values as 'single' for rows where 'job' is 'student'
- Calculate the most frequent marital status for each age
- fill unknown values in 'marital' based on age

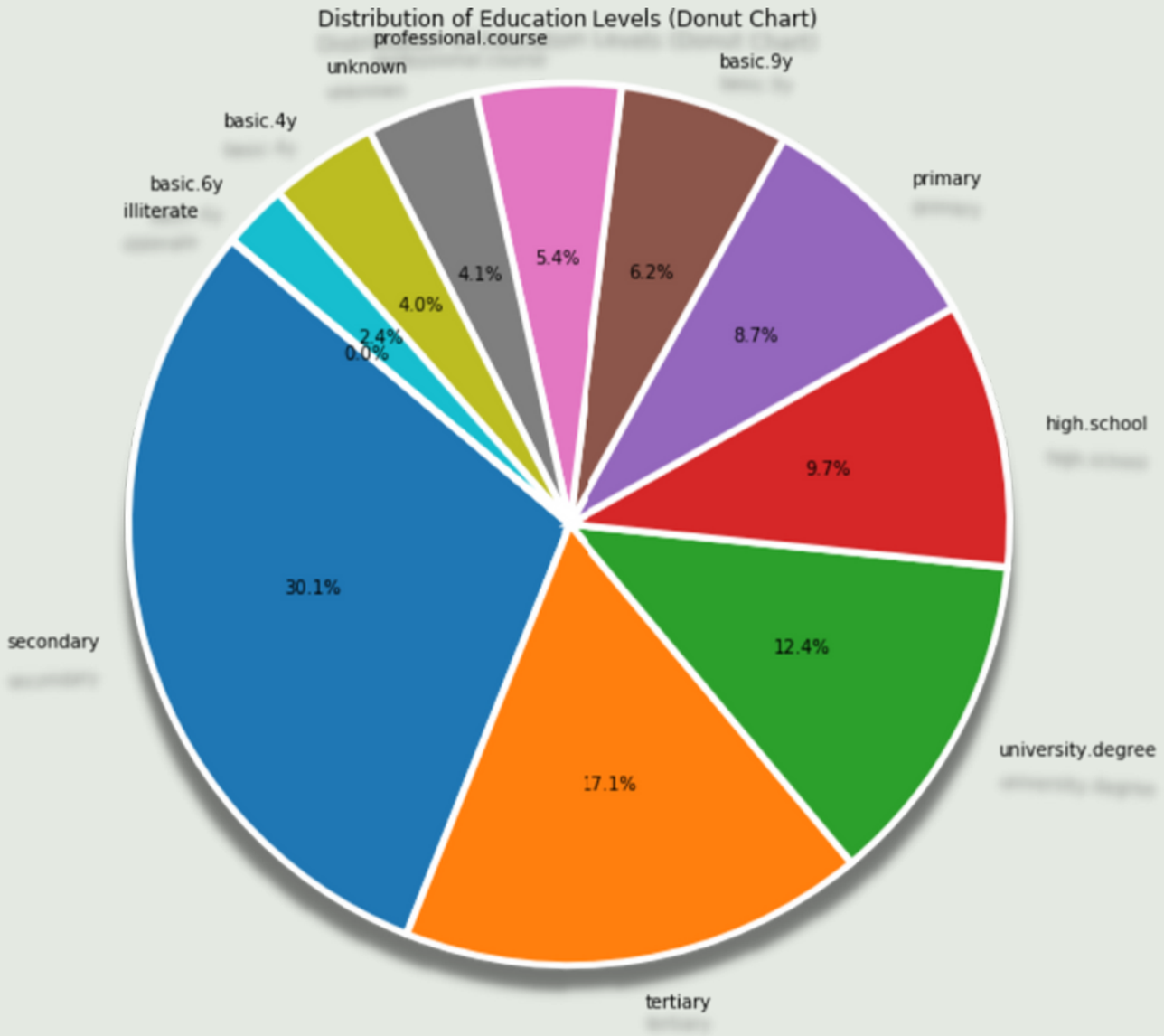


• Education Column



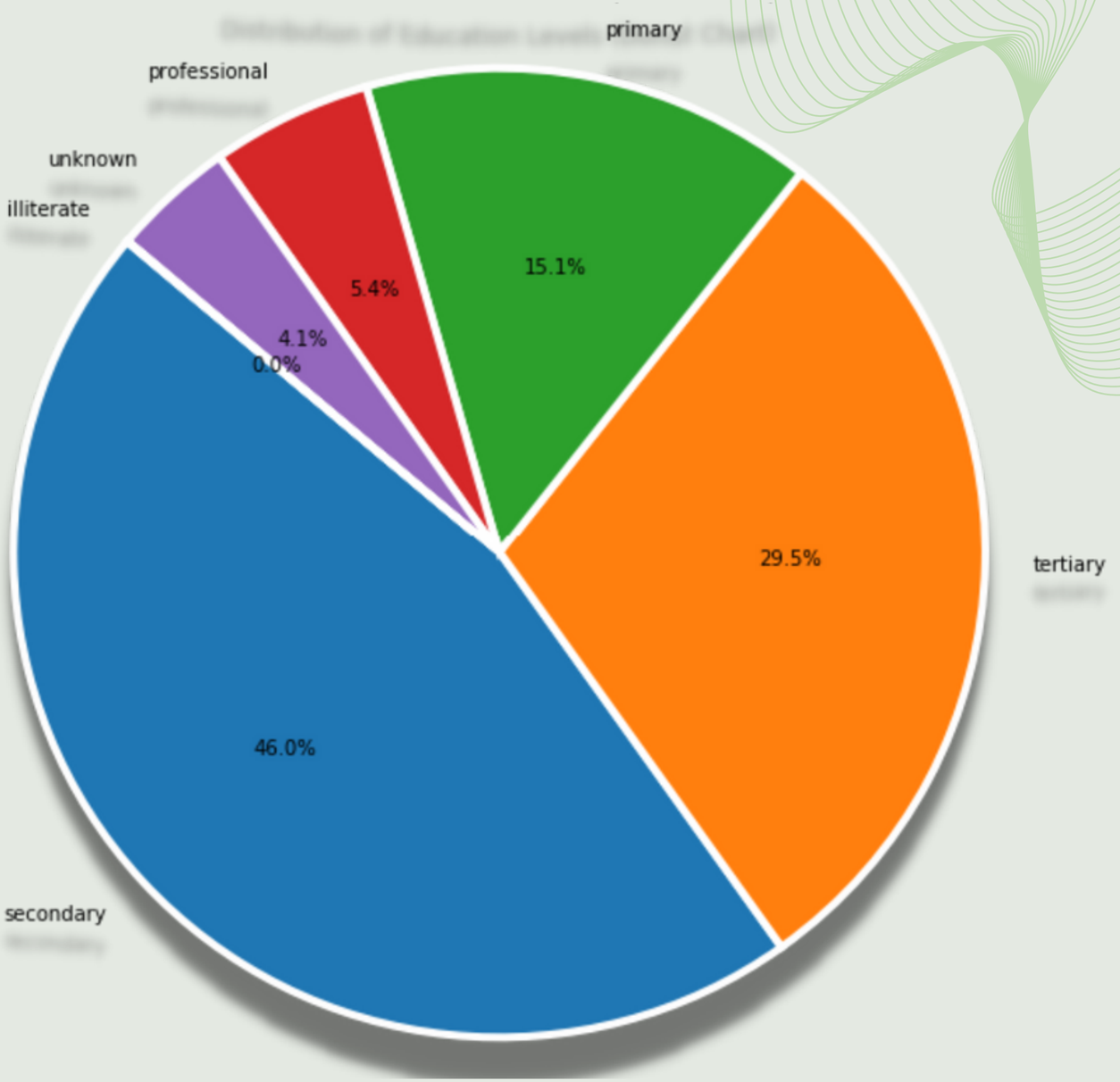
• Education Column

secondary	16285
tertiary	9272
university.degree	6694
high.school	5261
primary	4725
basic.9y	3347
professional.course	2901
unknown	2214
basic.4y	2149
basic.6y	1291
illiterate	12
Name: education, dtype:	int64



• Education Column

```
secondary      26292
tertiary       16405
primary         8858
professional    2929
unknown          216
illiterate        12
Name: education, dtype: int64
```



• Education Column

- Calculate modes of education for each unique age value
- replace 'unknown' values with corresponding modes

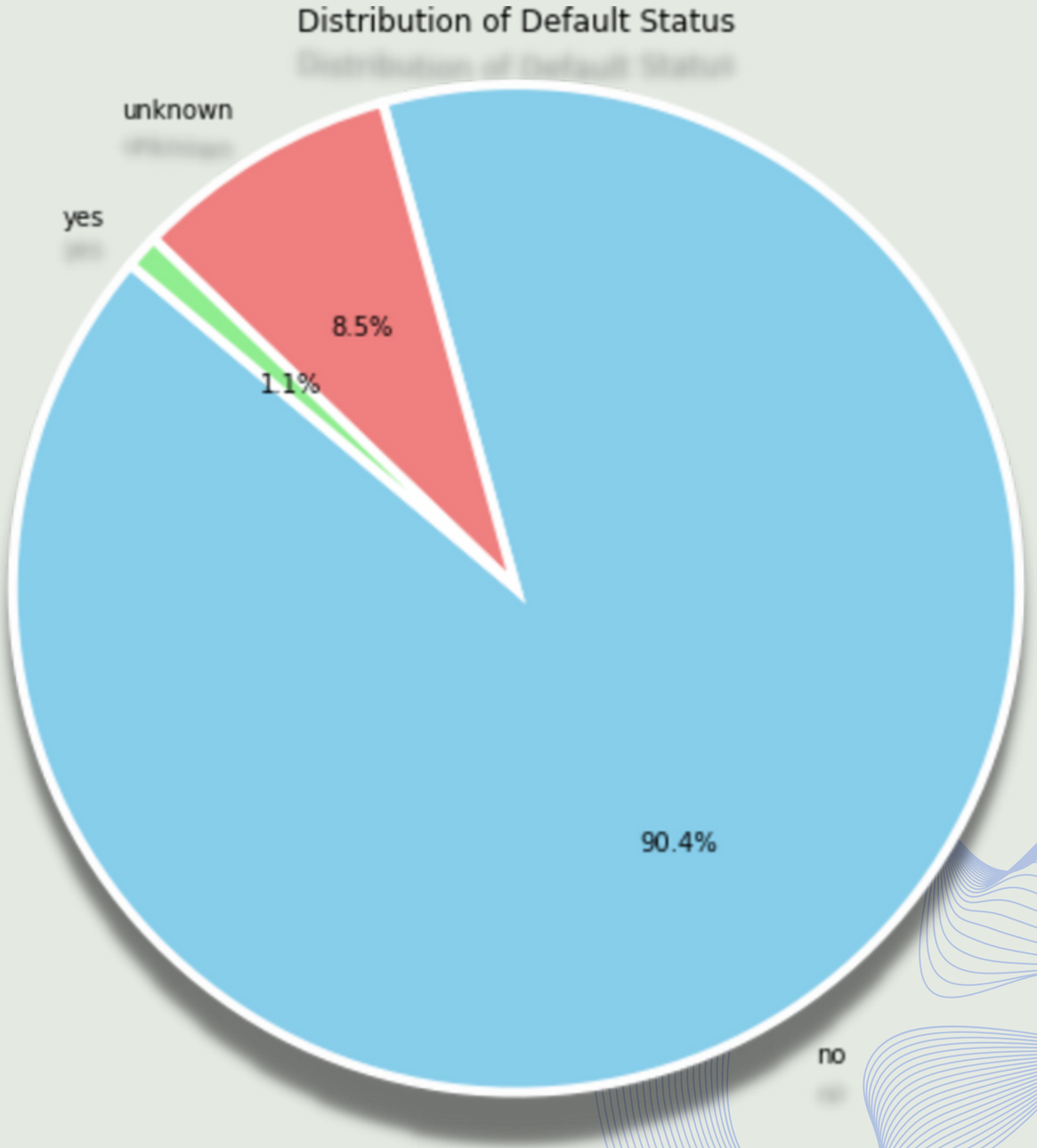
```
secondary       26474
tertiary        16405
primary          8877
professional     2929
unknown            15
illiterate         12
Name: education, dtype: int64
```

- Find the mode of the "education" column and replace "unknown" values with the mode

```
secondary       26489
tertiary        16405
primary          8877
professional     2929
illiterate         12
Name: education, dtype: int64
```

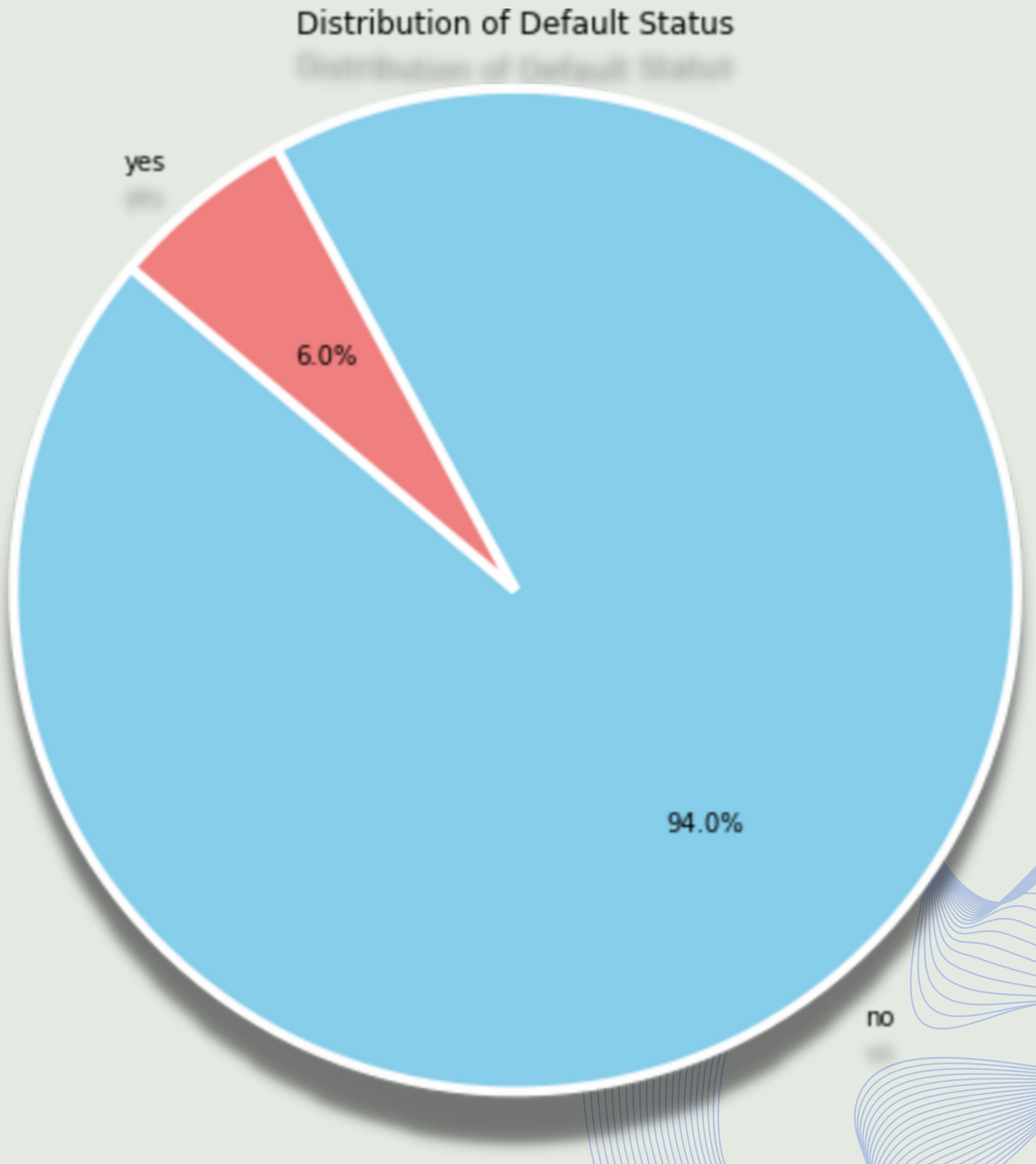
• Default Column

```
no         49368
unknown     4754
yes          590
Name: default, dtype: int64
```



Default Column

- checks if the "default" value is "unknown".
- checks if either the "loan" or "housing" column has a value of "yes", the "default" value for that row is set to "yes".
- If neither the "loan" nor "housing" column has a value of "yes", the "default" value for that row is set to "no".



• Housing & Loan Column

```
df['housing'].value_counts()
```

```
yes        29666
no         24503
unknown      543
Name: housing, dtype: int64
```

```
df['loan'].value_counts()
```

```
no         45564
yes         8605
unknown      543
Name: loan, dtype: int64
```

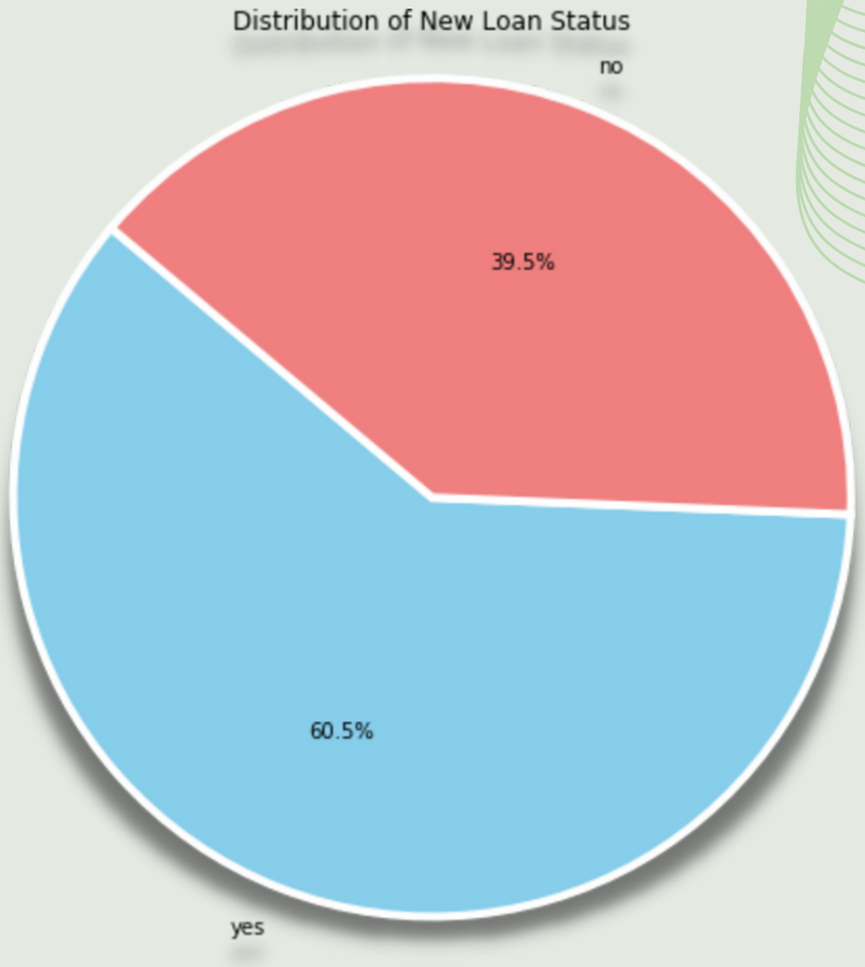
New_Loan Column

- If either the "housing" column or the "loan" column has a value of 'yes', the corresponding "new_loan" value is set to 'yes'.
- If both the "housing" and "loan" columns have values of 'no', the correspondi "new_loan" value is set to 'no'.
- If the "housing" column is 'unknown' and the "loan" column is 'yes', or if the "housing" column is 'yes' and the "loan" column is 'unknown', the corresponding "new_loan" value is set to 'yes'.
- If the "housing" column is 'unknown' and the "loan" column is 'no', or if the "housing" column is 'no' and the "loan" column is 'unknown', the corresponding "new_loan" value is set to 'no'.

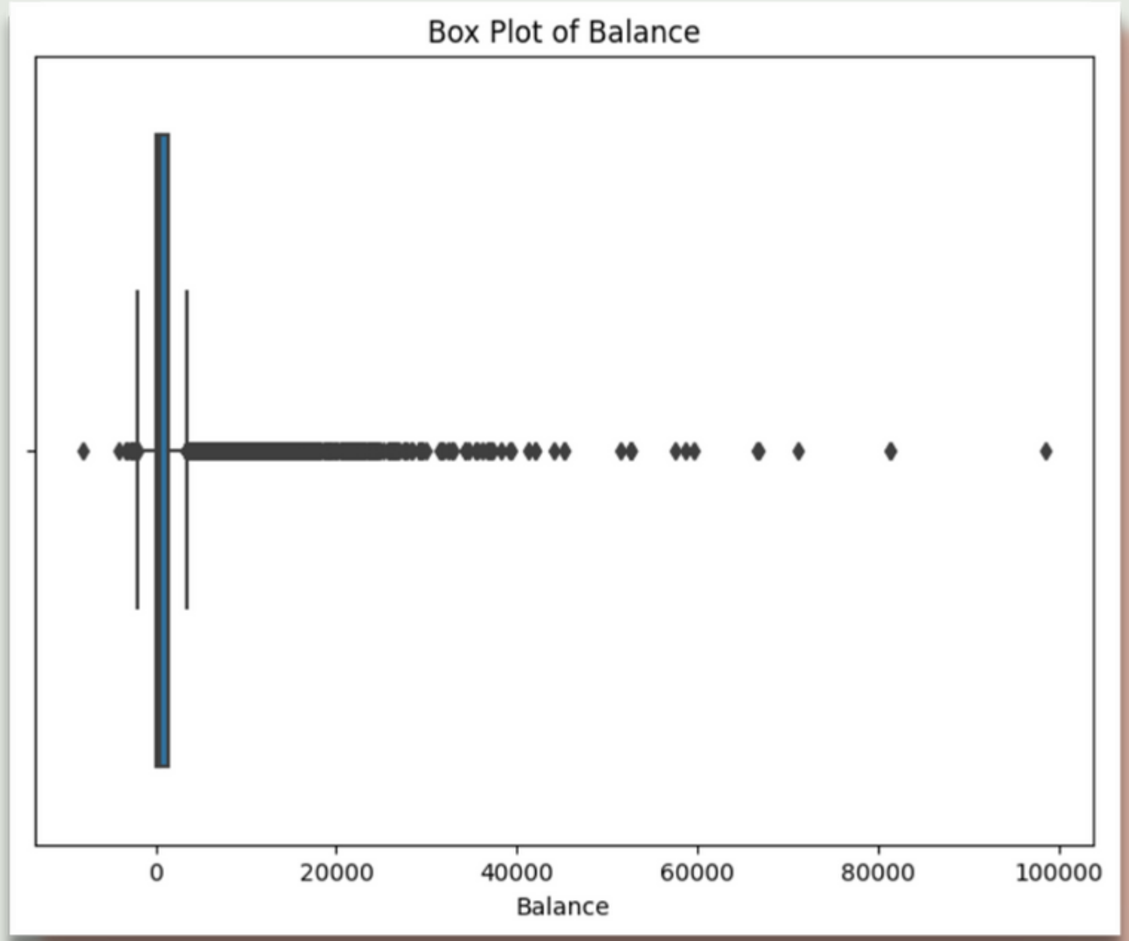
New_Loan Column

- refining the values in the "new_loan" column by considering the "default" column as well. If the "default" column is 'no', it sets the "new_loan" value to 'no', and if the "default" column is 'yes', it sets the "new_loan" value to 'yes'.

- Drop 'housing' and 'loan' columns



• Balance Column



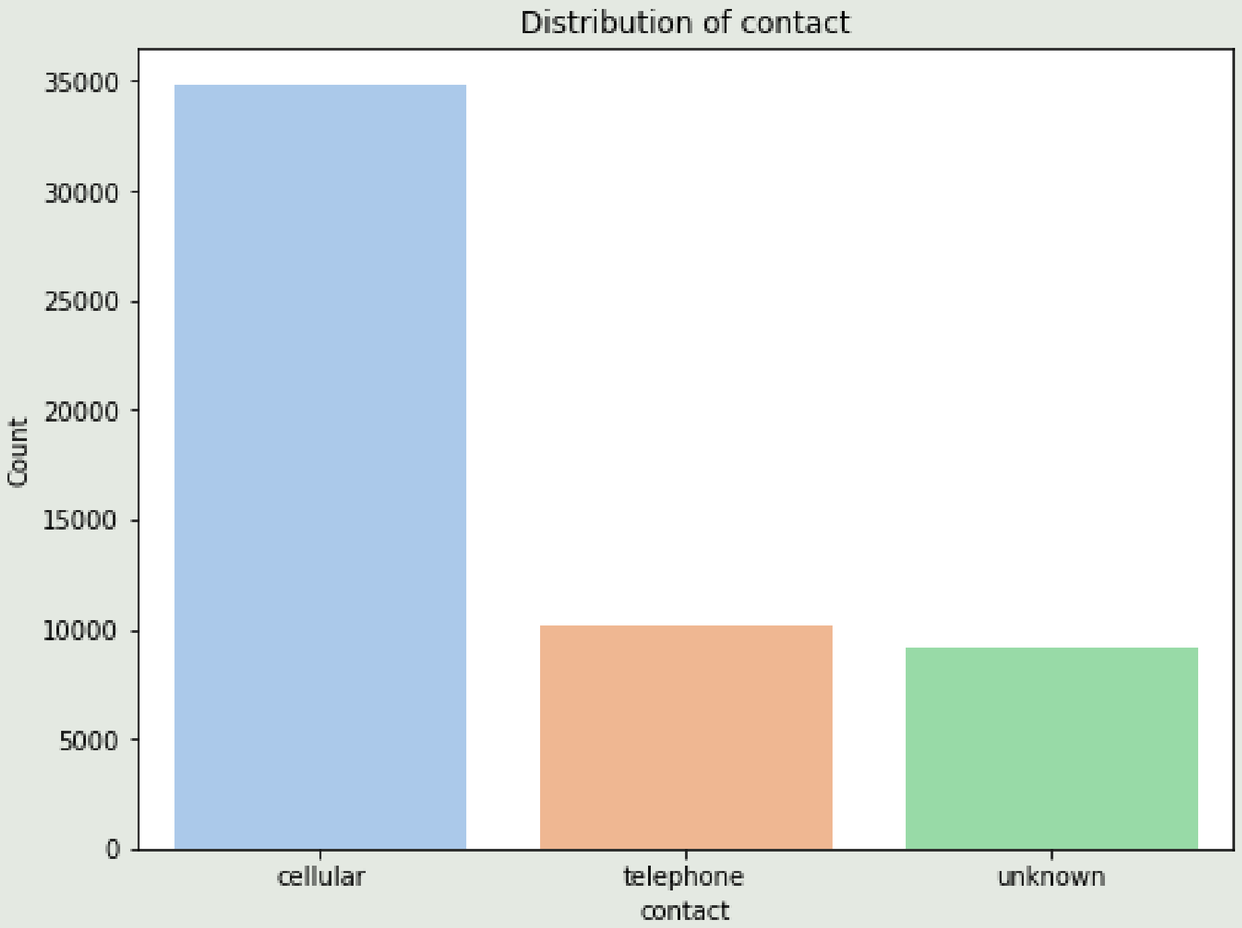
Calculate the average balance for each age and fill null values with average balance

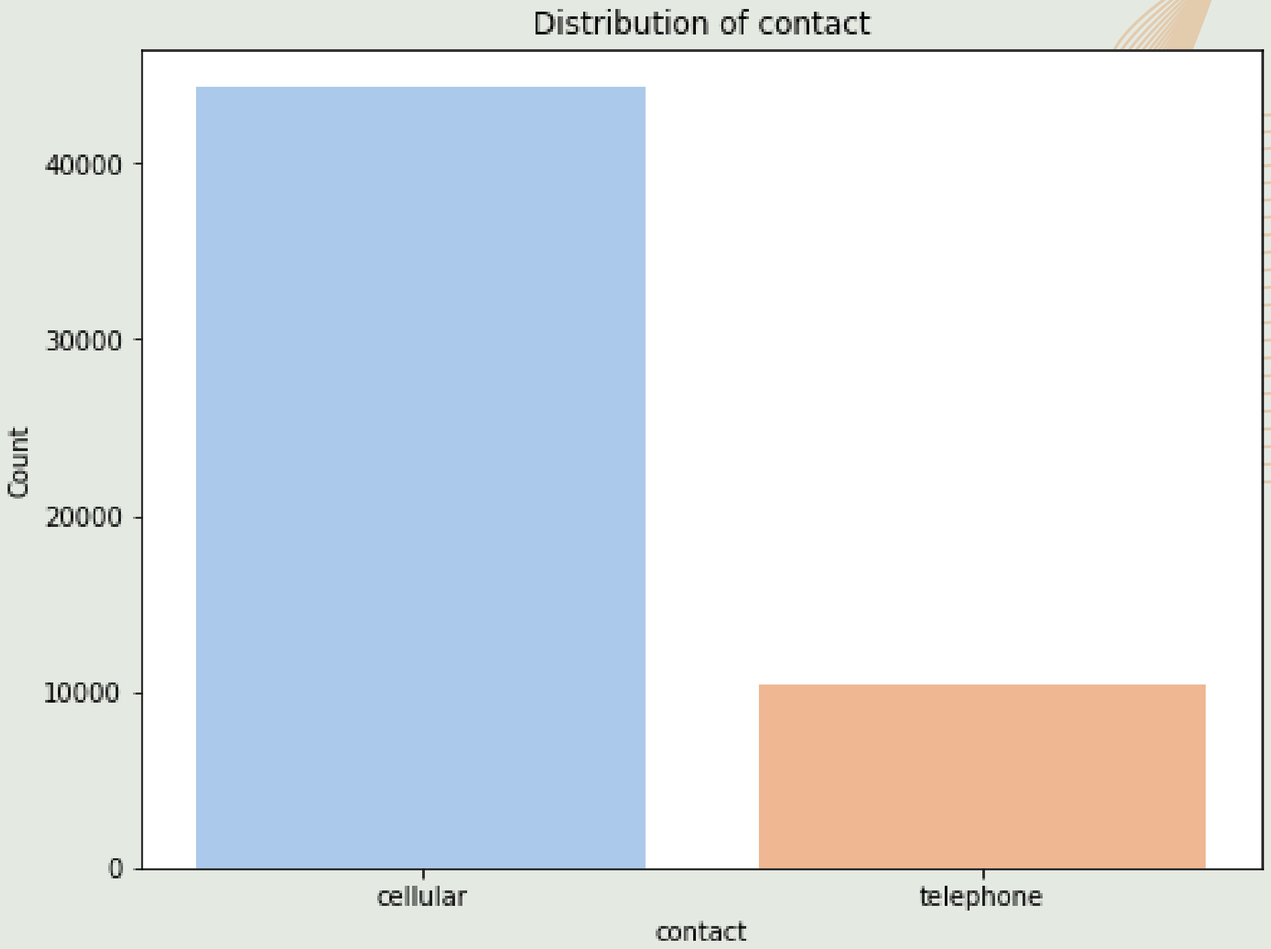
```
df.balance.describe()
```

```
count    31842.000000
mean      1368.024590
std       3057.847866
min      -8019.000000
25%         73.000000
50%        449.000000
75%       1435.000000
max      98417.000000
Name: balance, dtype: float64
```

• Contact Column

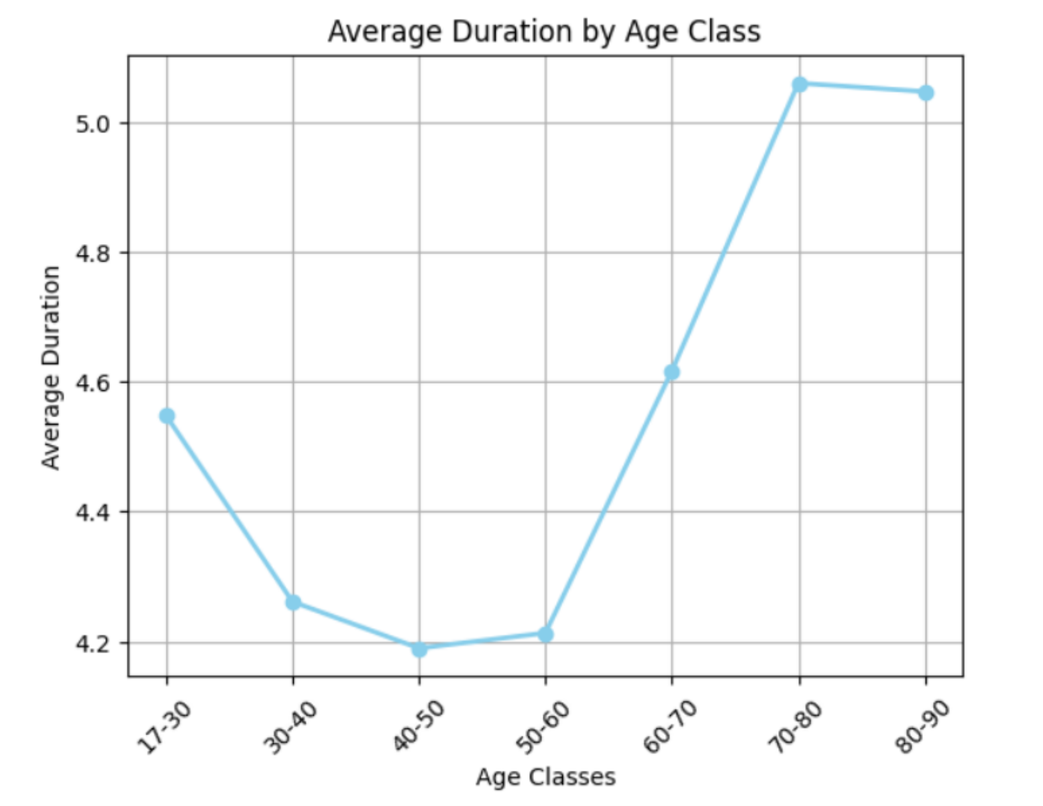
Replace "unknown" values with the mode of the "contact" column



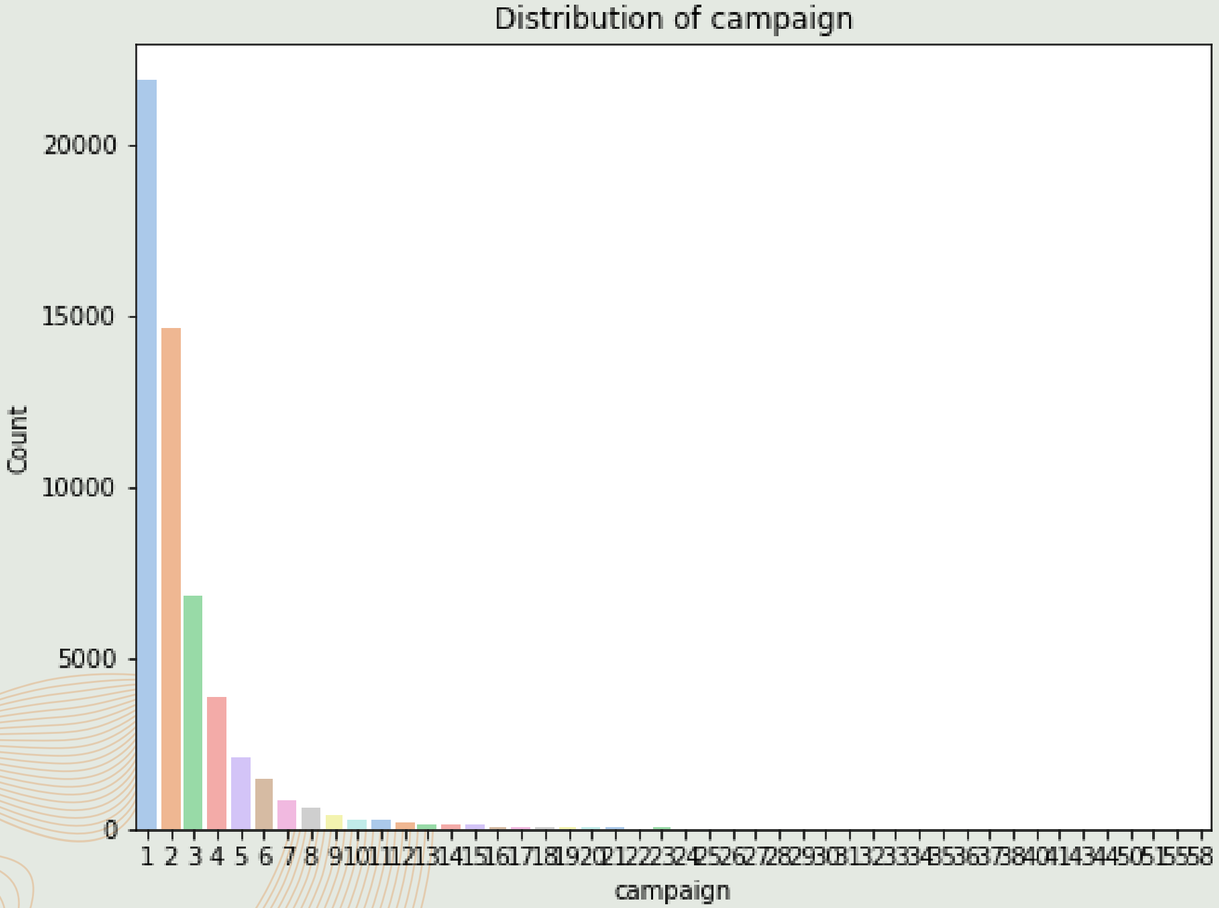


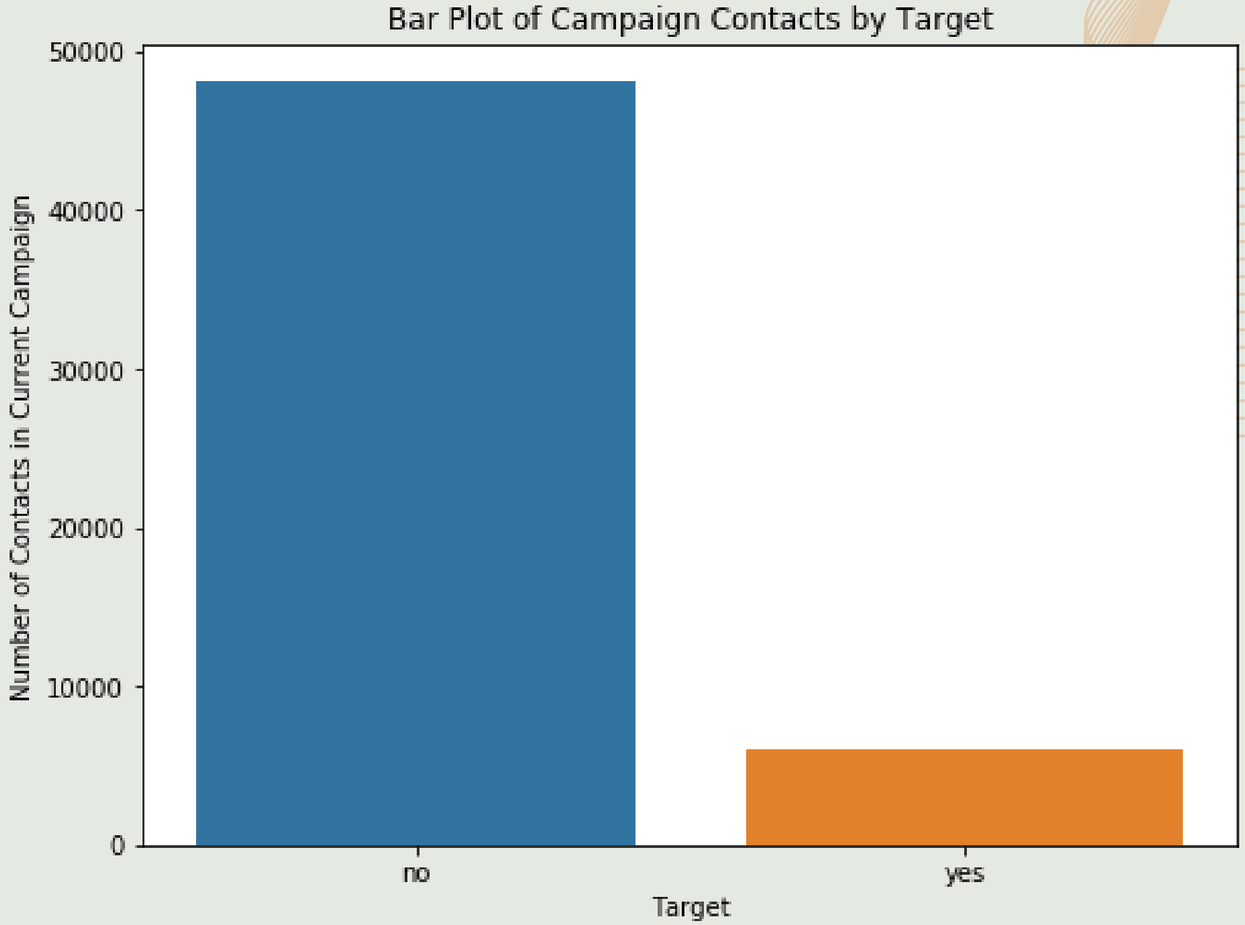
• Duration Column

convert seconds duration to minutes



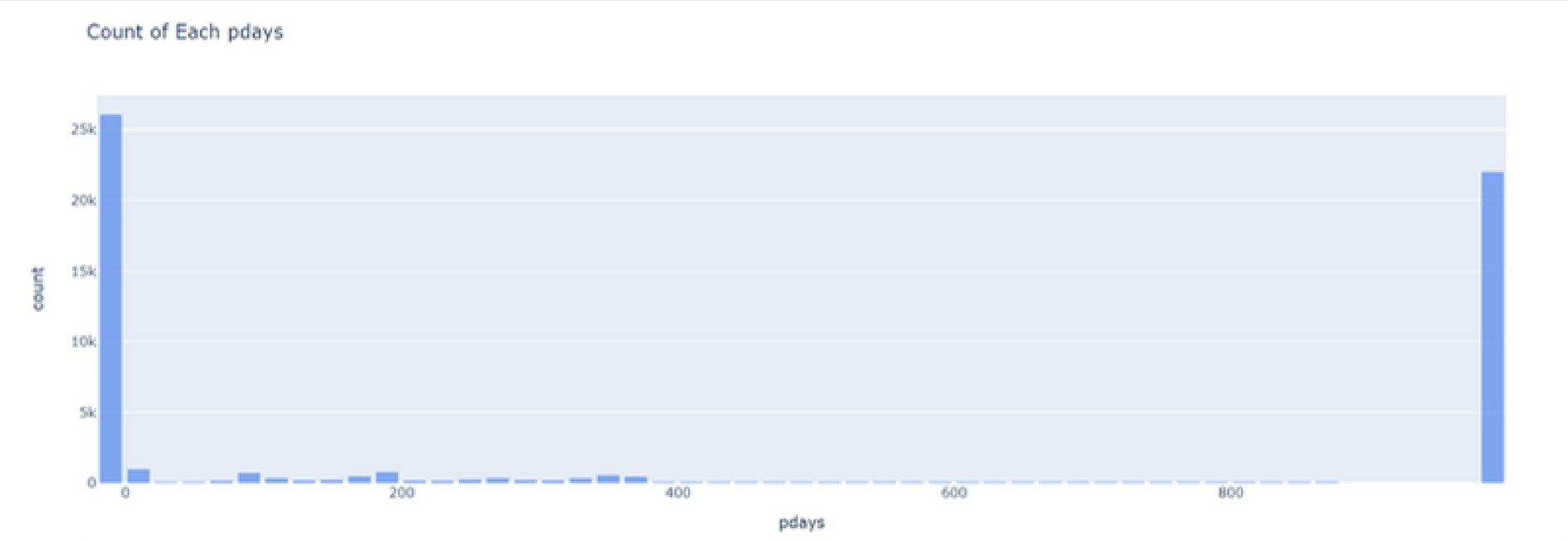
• Campaign Column



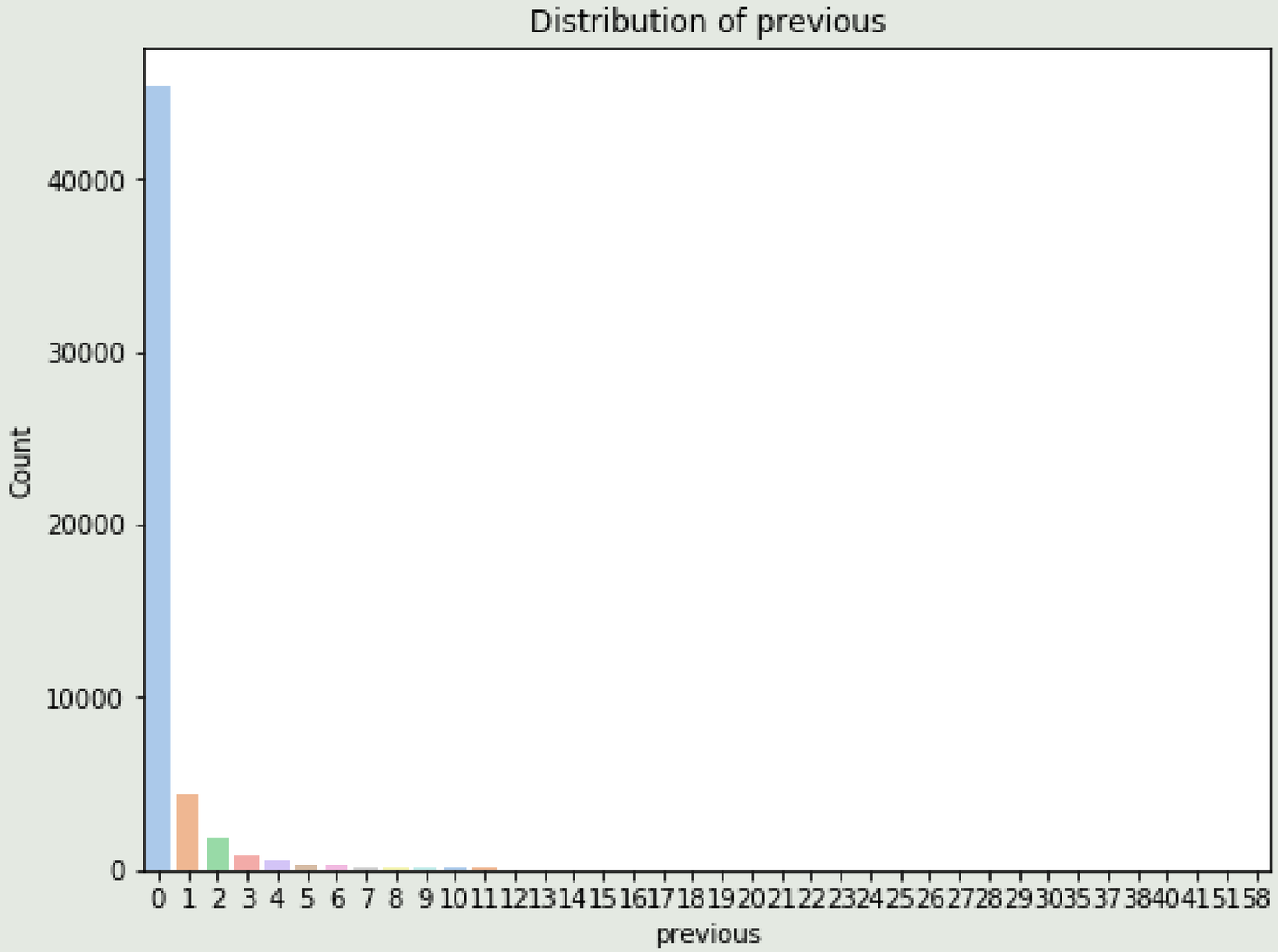


pdays Column

Loop through the 'pdays' column and replace 999 with -1



• Previous Column

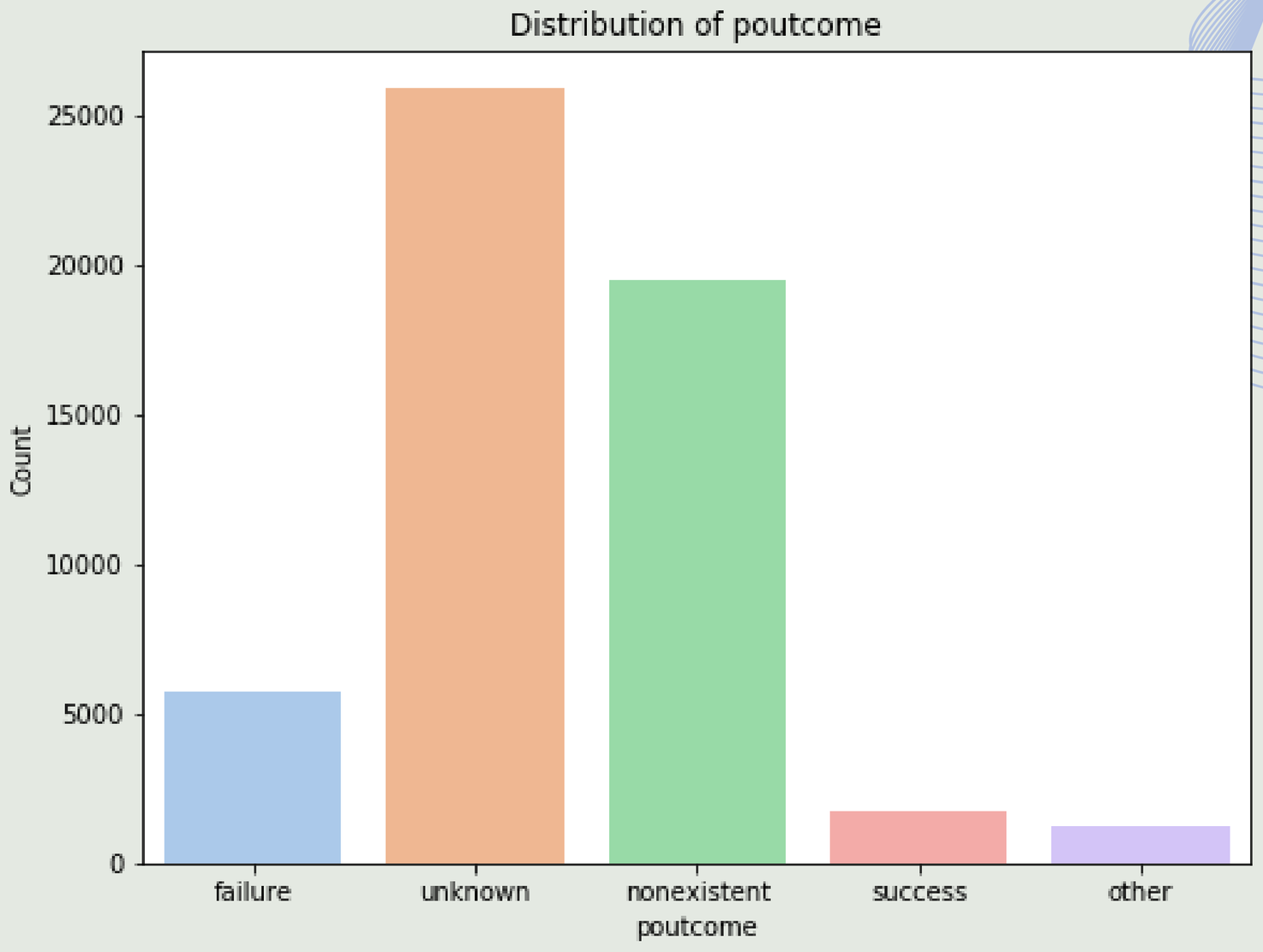


Identify values with less than 3 counts and Replace low count values with the mean

• Poutcome Column

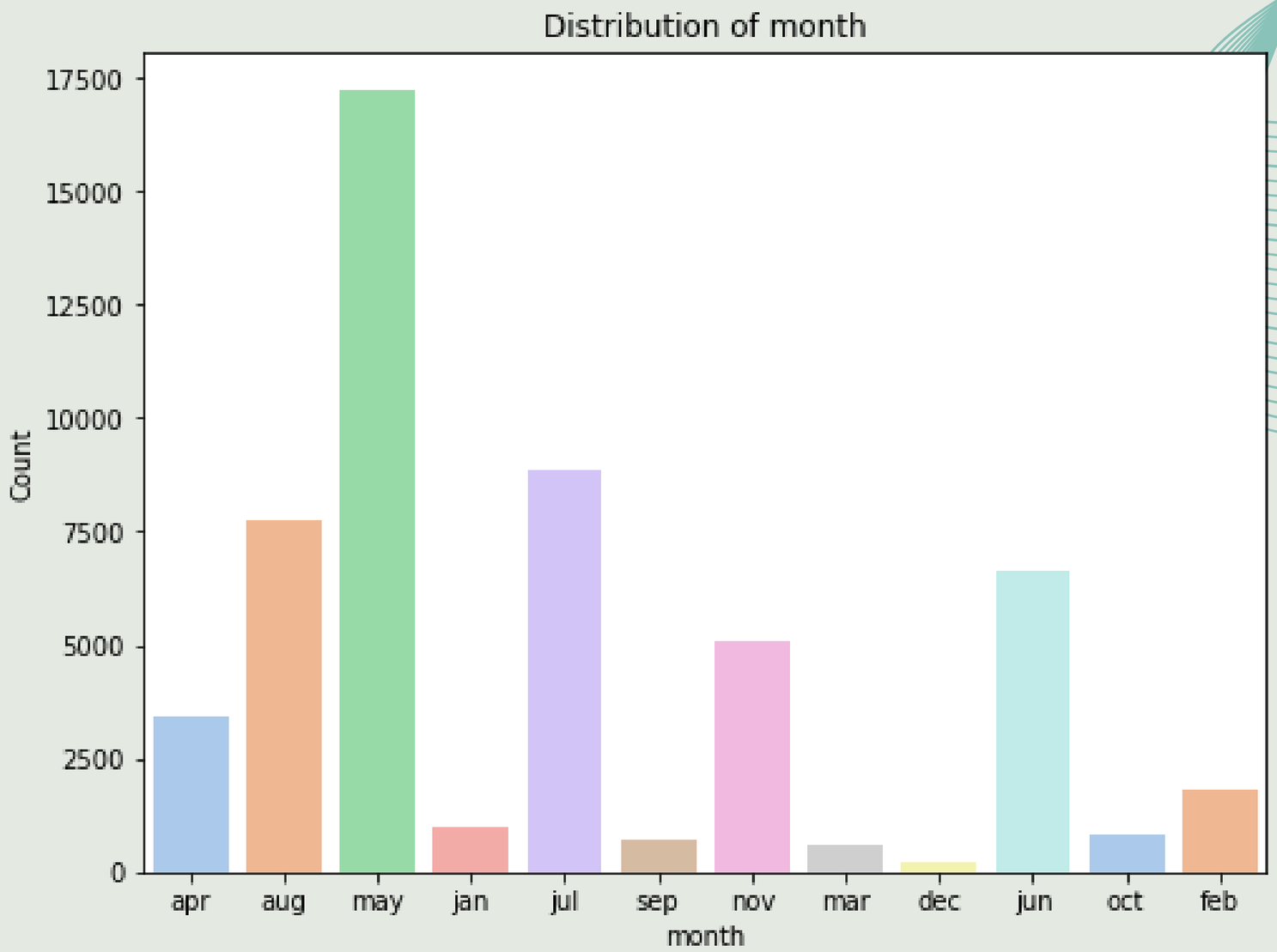
```
df['poutcome'].value_counts()
```

```
unknown        26057
nonexistent    19704
failure         5809
success         1855
other           1287
Name: poutcome, dtype: int64
```



• Month Column

- If the "month" value is invalid but the "day" value is a valid month abbreviation, it replaces the "month" value with the "day" value.
- drop day column



- # Data Scaling

  Standardize (normalize) the numerical columns using the StandardScaler ['age', 'balance', 'campaign', 'duration', 'previous']

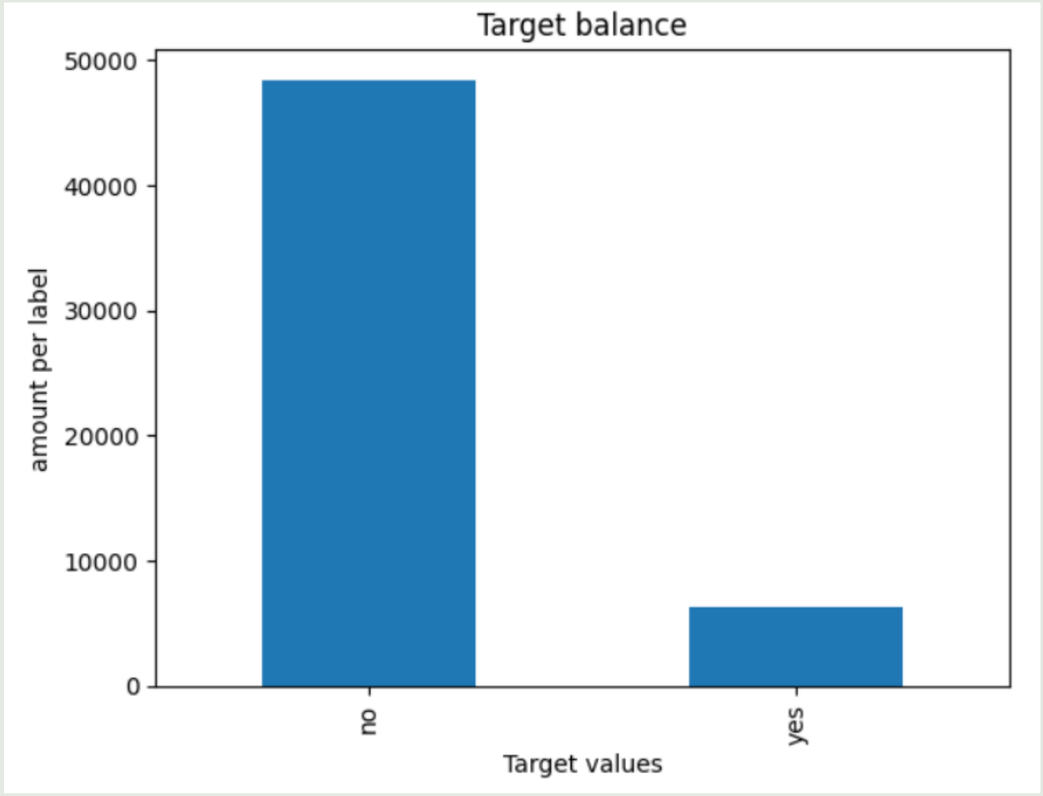
- # Data Encoding

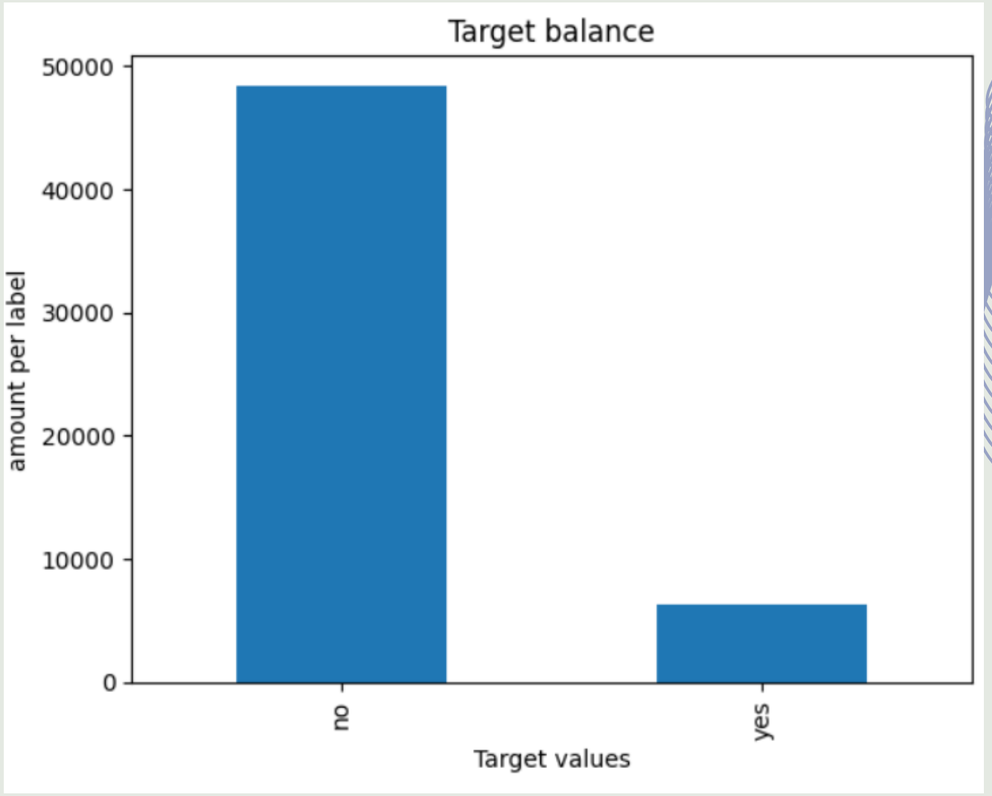
  Label encoding on specified categorical columns ['job', 'marital', 'education', 'default','contact','month','new_loan', 'poutcome']

- # Data splitting

  Split the data into 80% training and 20% validation sets

• Data over-sampling






```
no      48433
yes     47991
Name: Target, dtype: int64
no      48433
yes      6279
Name: Target, dtype: int64
```

Model Development & Evaluation

Random Forest Classifier

Adaboost

XGboost

ExtraTreesClassifier

• Random Forest

RF	TRAIN ACCURACY	VALIDATION ACCURACY	F1 SCORE VALIDATION	KAGGLE SCORE
Under sampling	0.903906	0.903999	0.894909	0.23105
SMOTE	0.904978	0.878187	0.886110	0.50396
SMOTETomek	0.932226	0.887142	0.892276	0.42881
Over sampling	0.962328	0.965310	0.965284	0.56908

• Adaboost

ADABOOST	TRAIN ACCURACY	VALIDATION ACCURACY	F1 SCORE VALIDATION	KAGGLE SCORE
n_estimators=50, random_state=42	0.898238	0.897743	0.883434	0.38931
learning_rate=1.0, n_estimators=150, random_state=42	0.888644	0.902312	0.889668	0.39683

```
print(best_params)
```

```
{'learning_rate': 1.0, 'n_estimators': 150}
```

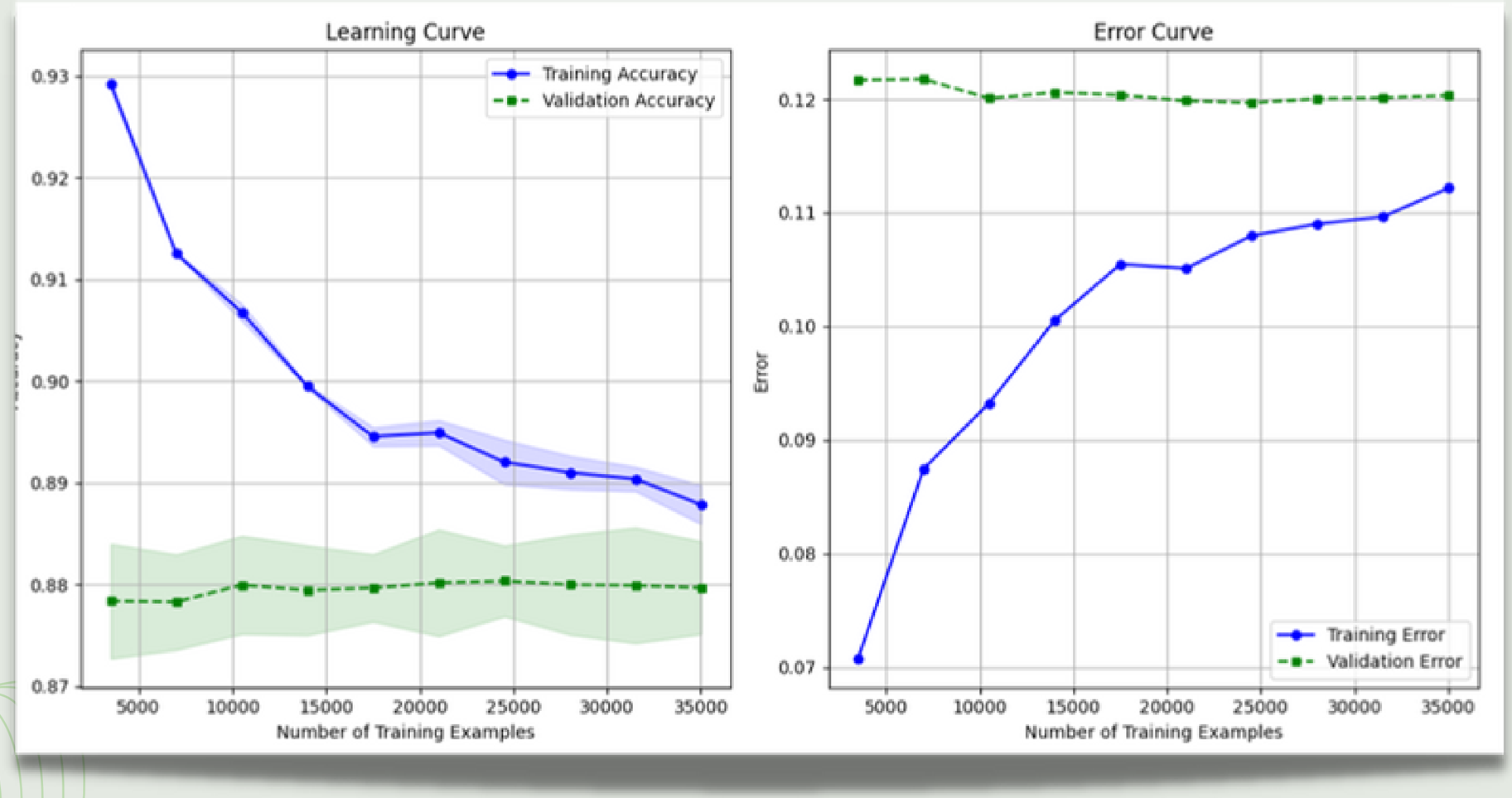
• XGBOOST

XGBOOST	F1 SCORE TRAIN	F1 SCORE VALIDATION	KAGGLE SCORE
n_estimators=30, random_state=42	0.91903	0.90031	0.46042
max_depth': 4 'n_estimators': 50 'scale_pos_weight': 4.0	0.89561	0.89147	0.58366

```
gcv.best_params_
```

```
{'max_depth': 4, 'n_estimators': 50, 'scale_pos_weight': 4.0}
```

- # Learning curve & Error curve



XGboost

• ExtraTreesClassifier

EXTRATREES CLASSIFIER	F1 SCORE TRAIN	F1 SCORE VALIDATION	KAGGLE SCORE
min_samples_split=10, n_estimators=200	0.95318	0.88730	0.39218
'class_weight': 'balanced', 'min_samples_leaf': 1, 'min_samples_split': 5, 'n_estimators': 150	0.8968	0.8974	0.54776

Future work

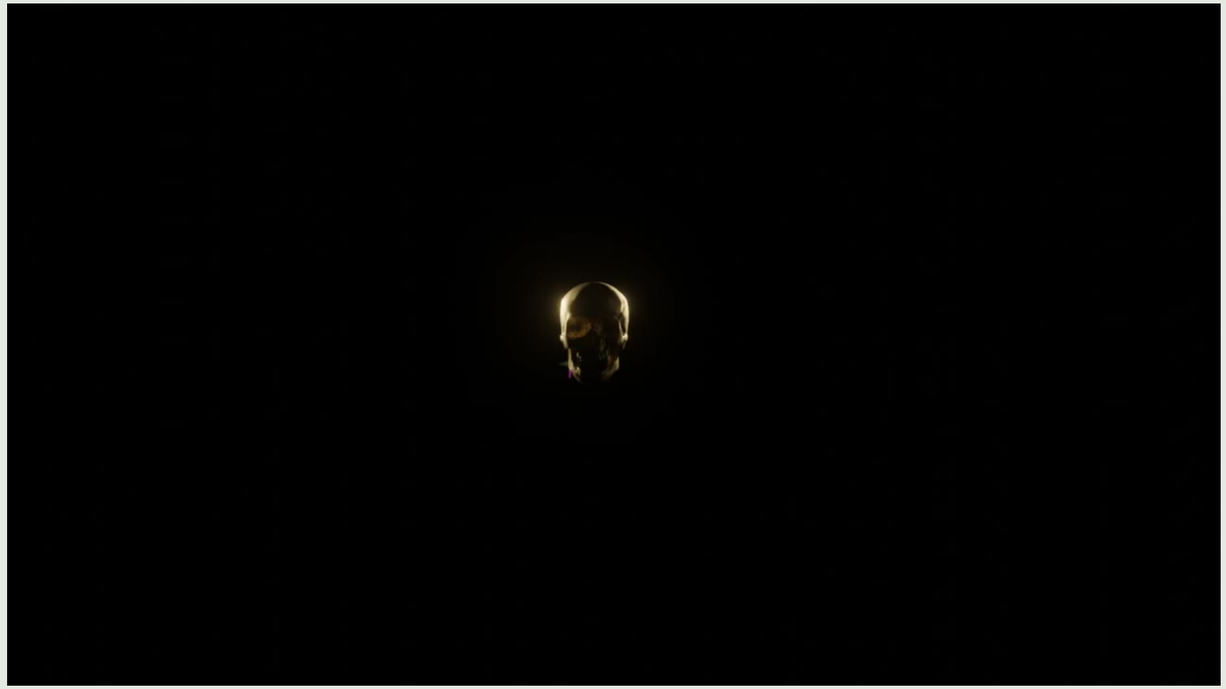
- Enhancing Performance
- Explore additional features
- Different ways for filling features

References

- Daniel, B.(2019).Cross-validation.(2-6),Meguro-ku, Tokyo 152-8550, Japan
- Hamed, T.(2016).Sampling Methods in Research Methodology; How to Choose a Sampling Technique for Research.(2-8), Vol. 5, No. 2, 2016, Page: 18-27, ISSN: 2296-1747© Helvetic Editions LTD,
- Nabila, F ,and M. A. Jabbar,(2016) Random Forest Modeling for Network Intrusion Detection System. Procedia Computer Science 89 (2016) 213 – 217.
- Barry, V.(2013)Decision trees.Overview wires.(wiley.com/compstats)448 -453
- https://www.linkedin.com/pulse/feature-scaling-dataset-spliting-arnab-mukherjee/

FiNas

Leen Almallah
Raneem Alomari
Muna Alsaber
Mujahed Alissa



Eng. Hasan Aloqool
Eng. Khawla Alquraan
Eng. Haneen Alakhrass

We Thank You for Your Time