



# כריית מידע - פרויקט חלק 1

## *Data information - שלב 1*

### הצגת המידע:

תחילה, הצגנו את המידע על ידי הפקודה `df.head()`. ראינו שיש כ-30517 נתוני מידע ולכל אחד יש כ-20 פיצ'רים.

תיקנו את הערכים שהיו כתובים כ-unknown ל `np.nan` והצגנו את `df.info()`:

```
df.info()

<class 'pandas.core.frame.DataFrame'>
Int64Index: 30517 entries, 512491 to 516748
Data columns (total 20 columns):
 #   Column                Non-Null Count  Dtype  
---  -
 0   status                30517 non-null  object  
 1   age                   30517 non-null  int64   
 2   n_child               10697 non-null  object  
 3   education             29273 non-null  object  
 4   profession            30324 non-null  object  
 5   device                21646 non-null  object  
 6   account_balance       30517 non-null  int64   
 7   loan                  30517 non-null  bool     
 8   mortgage              30517 non-null  bool     
 9   credit                30517 non-null  bool     
10  positive              30517 non-null  bool     
11  campaign_type         30517 non-null  object  
12  consent               30517 non-null  bool     
13  n_contact             30517 non-null  int64   
14  l_date                30517 non-null  object  
15  l_call_duration       30517 non-null  int64   
16  p_outcome             5574 non-null   object  
17  n_p_contact           30517 non-null  int64   
18  p_days                30517 non-null  int64   
19  subscribed            30517 non-null  bool     
dtypes: bool(6), int64(6), object(8)
memory usage: 3.7+ MB
```

יש 6 פיצ'רים מסוג `int`.

שאר הפיצ'רים הם מסוג `bool/object`.



לפי מבט ראשוני על המידע ניתן לראות כי לפיצ'רים הבאים חסרים ערכים:

- n\_child - missing 19,820 values.
- education - missing 1,244 values.
- profession - missing 193 values.
- device - missing 8,871 values.
- p\_outcome - missing 24,943 values.

בנוסף פיצלנו את ה Data frame ל 3 סוגים שונים:

1. target - the target value of the data - 'Subscribed'.
2. num\_data - the numeric data. (All the int features)
3. nom\_data - the nominal data. (The rest of the features object\bool).

## שלב 2 - Data statistics & Visualization

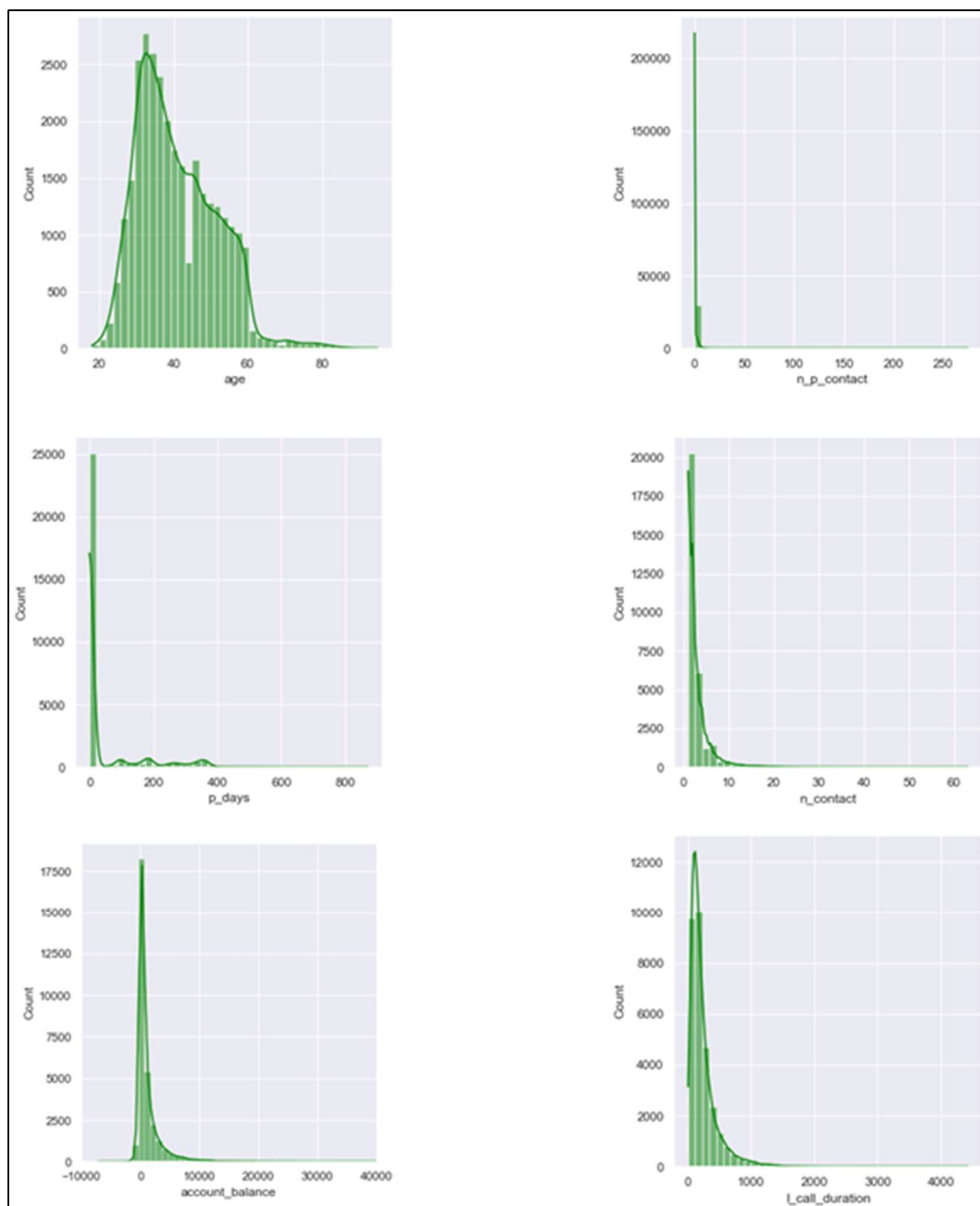
### Numeric Data:

הצגת המידע הסטטיסטי:

num_data.describe()						
	age	account_balance	n_contact	l_call_duration	n_p_contact	p_days
count	30517.000000	30517.000000	30517.000000	30517.000000	30517.000000	30517.000000
mean	40.873546	1228.707966	2.769604	233.294262	0.58397	40.320706
std	10.591058	2738.410757	3.085730	232.690931	2.48213	100.489272
min	18.000000	-7207.000000	1.000000	2.000000	0.00000	-1.000000
25%	33.000000	74.000000	1.000000	95.000000	0.00000	-1.000000
50%	39.000000	411.000000	2.000000	164.000000	0.00000	-1.000000
75%	48.000000	1278.000000	3.000000	286.000000	0.00000	-1.000000
max	95.000000	91924.000000	63.000000	4428.000000	275.00000	871.000000



## הצגת הגרפים:





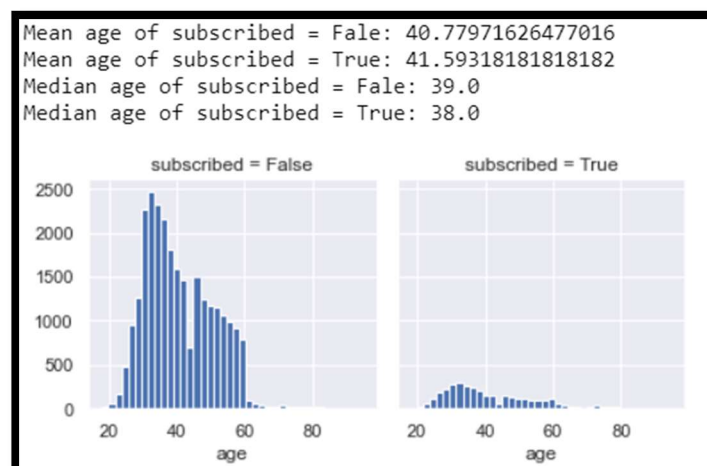
## חישוב Skewness:

Feature	Skewness
Age	0.6962
P_days	2.6168
L_call_duration	3.3062
N_contact	4.7919
Account_balance	8.3480
N_p_contact	48.0634

ניתן לראות מהגרפים וטבלת ה Skewness שמסודרת בסדר עולה, כי age מתפלג בצורה שהכי קרובה להתפלגות אחידה (הכי סימטרי) לעומת n\_p\_contact שמתפלג בצורה הכי לא סימטרית. כמו כן, לשאר הפיצ'רים הנומרים ששונים מ age, יש ערכים חריגים (הרוב נמצא באיזור 0 ומעט בערכים הגדולים מ 0).  
טיפולנו בחריגים בחלק של – Data transformation.

## חקירת הפיצ'רים הרלוונטיים בהתאם לערך המטרה – subscribed:

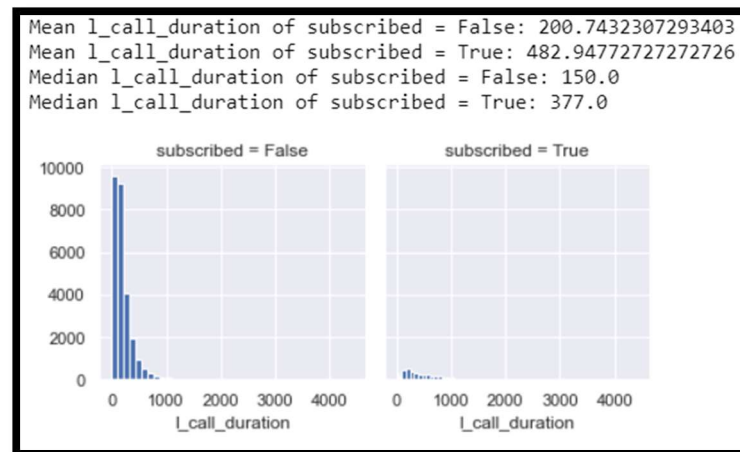
Age vs subscribed





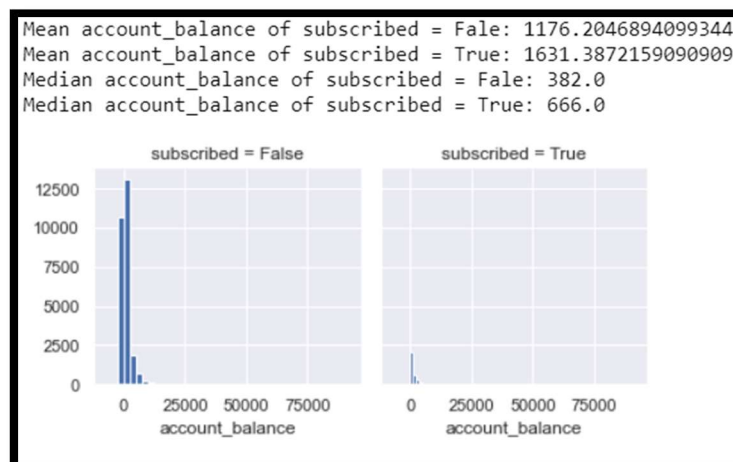
לא ניתן ללמוד יותר מדי מהגיל הממוצע או החציון של הגילאים שבהם הסכימו לרכוש ביטוח. מכיוון שההתפלגות של הגרפים יצאה זהה ובנוסף הממוצעים + החציונים יצאו זהים.

L\_call\_duration vs subscribed



ניתן לראות כי אורך השיחה השפיע על רכישת הביטוח – ככל שאורך השיחה עלה ככה יותר אנשים רכשו ביטוח, אורך שיחה ממוצע של 482 וחציון של 377. הסקנו יותר מסקנות לפיצ'ר זה ב scatter plot.

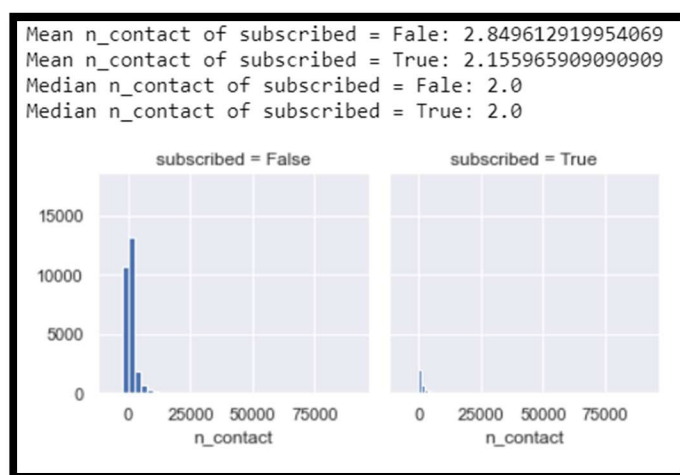
account\_balance vs subscribed





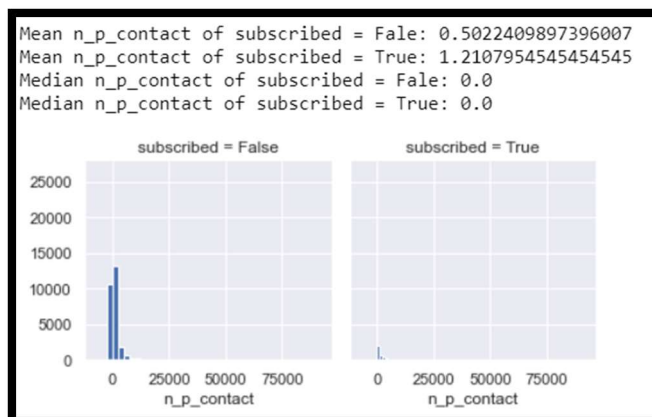
ניתן לראות כי ממוצע ההכנסה השנתית השפיע על רכישת הביטוח – ככל שממוצע ההכנסה השנתית היה גבוה יותר ככה יותר אנשים רכשו ביטוח. לא הצלחנו ללמוד מההשפעה של מהממוצע של ממוצע ההכנסה השנתית (כנראה בגלל שיש יותר ערכים קיצוניים), אך ניתן ללמוד מהחציון שהוא 666 - ככל שלאנשים היה יותר כסף הם בסוף רכשו ביטוח.

n\_contact vs subscribed



לא ניתן ללמוד יותר מדי ממספר השיחות שבוצעו ללקוח במהלך הקמפיין הנוכחי – מהו המספר המתאים ביותר בכדי שלקוח ירכוש בסוף ביטוח. מכיוון ש הממוצעי שיחות וגם ה median של הלקוחות שרכשו בסוף ביטוח ולא רכשו יצאו זהים, ובנוסף ההתפלגות של הגרפים זהה.

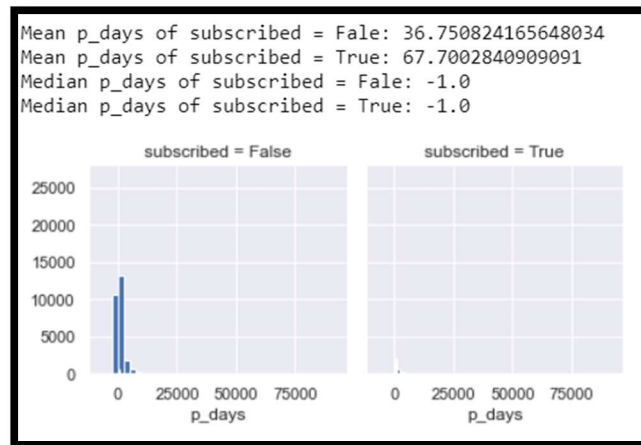
n\_p\_contact vs subscribed





ניתן לראות כי מספר הפעמים שהתקשרו לכל לקוח בקמפיין הקודם השפיע על רכישת הביטוח.  
ככל שמספר הפעמים שהתקשרו לכל לקוח בקמפיין הקודם היה לפחות פעם אחת, ככה זה השפיע עליו לרכוש ביטוח בקמפיין הנוכחי.  
(המסקנה הזו נבעה מהממוצעים, אך לא ניתן ללמוד מהחציון).

P\_days vs subscribed



עבור p\_days הסקת המסקנות הייתה מסובכת יותר, כי קיימים ערכים ב data שעבורם p\_days == -1.

בהמשך חקירת המידע הבנו שערכים אלו הם עבור אנשים שלא השתתפו בקמפיין הקודם.

אז כרגע על פי הגרפים הללו ניתן לראות שעבור הגרף שבו subscribed == False לקוחות שדיברו איתם לפני הרבה זמן מהקמפיין הקודם לא רכשו ביטוח.



## Nominal Data:

הצגת המידע הסטטיסטי:

	status	n_child	education	profession	device	loan	mortgage	credit	positive	campaign_type	consent	l_date	p_outcome
count	30517	10697	29273	30324	21646	30517	30517	30517	30517	30517	30517	30517	5574
unique	3	4	3	11	2	2	2	2	2	1	2	308	3
top	married	1	master	engineer	cellular	False	True	False	True	phone call	False	15-May	failure
freq	18414	3058	15622	6538	19718	25632	17055	29987	28059	30517	26997	782	3330
Percent of top value	60.34	10.021	51.191	21.424	64.613	83.993	55.887	98.263	91.945	100.0	88.465	2.563	10.912

ראשית, ניתן לראות שיש 13 פיצ'רים. אלו הם הפיצ'רים מהסוג שלא integer כלומר bool/object.

שמנו לב שקיימת בעיה עם העמודה l\_date – פעם אחרונה שיצרו קשר עם הלקוח. העמודה מוצגת בשילוב של מספר יום + חודש: day\_month. החלטנו ליצור עמודה חדשה – month\_l\_date ולחקור את פיצ'ר זה ברמת החודש, ולא לחקור לרמת היום, מכיוון שאנו חושבים שכך יהיה יותר קל לנהל את ה data ולבצע חישובים.

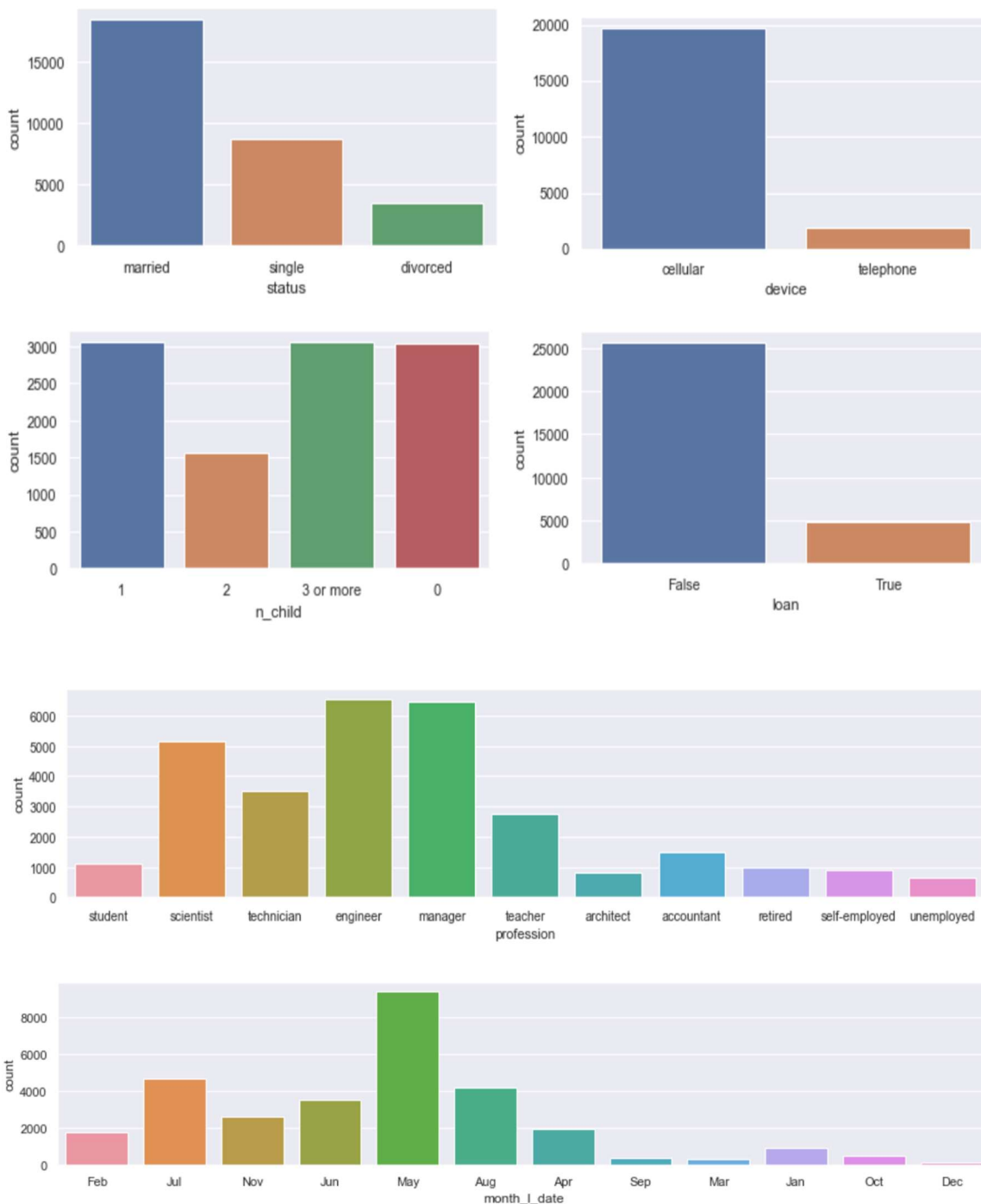
כמו כן, לא הרסנו את העמודה המקורית שאם נבין שהחלטה זו לא יעילה עדיין נוכל להשתמש בעמודה l\_date.

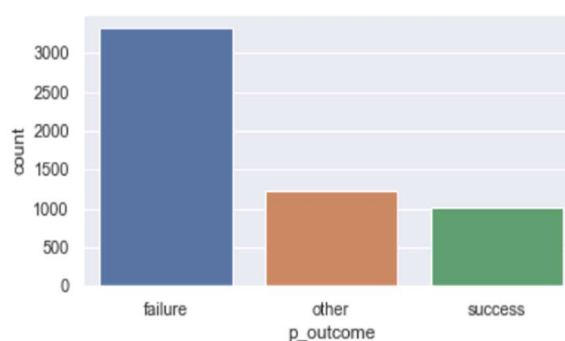
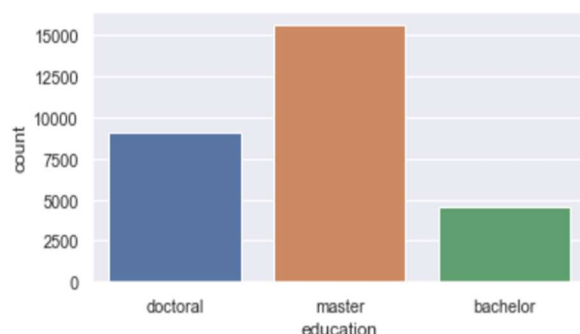
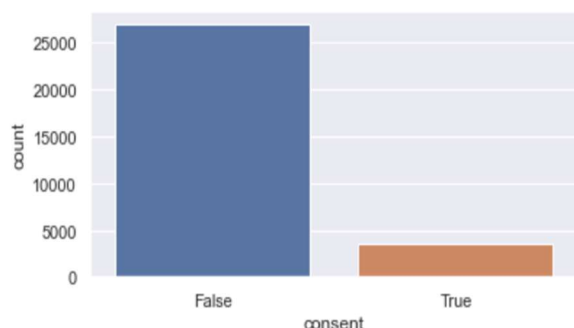
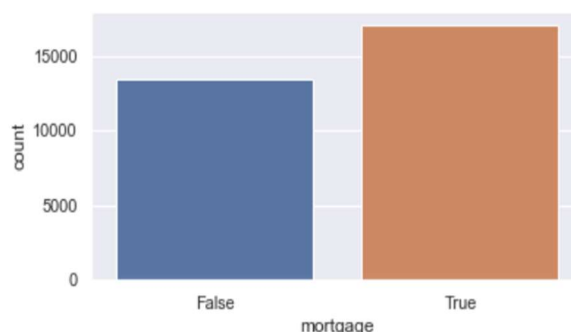
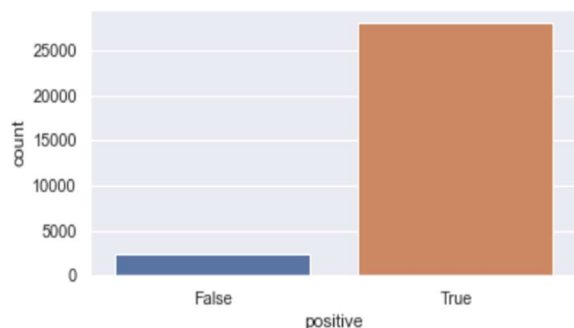
תחילה הצגנו כל פיצ'ר על ידי גרף.





## הצגת הגרפים:





### מסקנות:

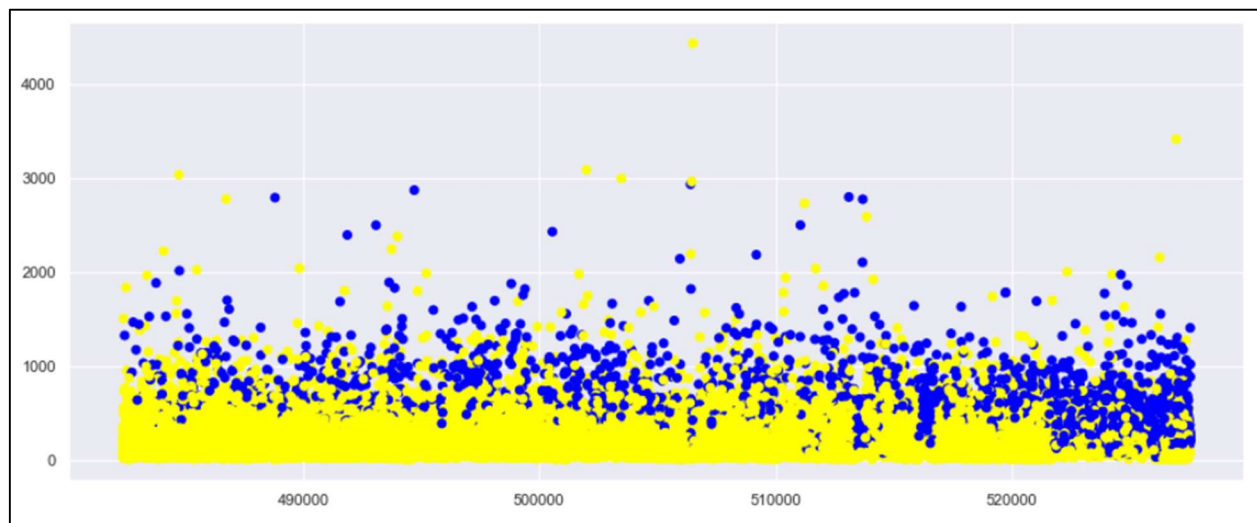
- ניתן לראות כי רוב המשתתפים נשואים.
- ניתן לראות כי רוב המשתתפים עם רמת השכלה master.
- ניתן לראות כי רוב המשתתפים לא נמצאים במינוס בחשבון הבנק.
- ניתן לראות כי לרוב המשתתפים יש אשראי.
- ניתן לראות כי רוב המשתתפים לא לקחו הלוואה.
- בn\_childs ובp\_outcome יש הרבה ערכים חסרים ולכן לא ניתן להסיק מסקנות.
- ניתן לראות שרוב הפניות למשתתפים התבצעו טלפון האישי (השתמשו בעובדה זו בתהליך ה pre\_processing).



- לא הוספנו גרף עבור campaign type מכיוון שהוא 100% טלפוני (בהמשך מחקנו את עמודה זו).
- לא הוספנו גרף עבור consent מכיוון שבהמשך מחקנו עמודה זו (והסברנו מדוע).
- ניתן לראות כי לרוב המשתתפים פעם האחרונה שיצרו איתם קשר היה בחודש מאי.

## Scatter plot:

בחרנו לעשות scatter plot עבור subscribed vs call\_duration.



**Subscribed → False = yellow**

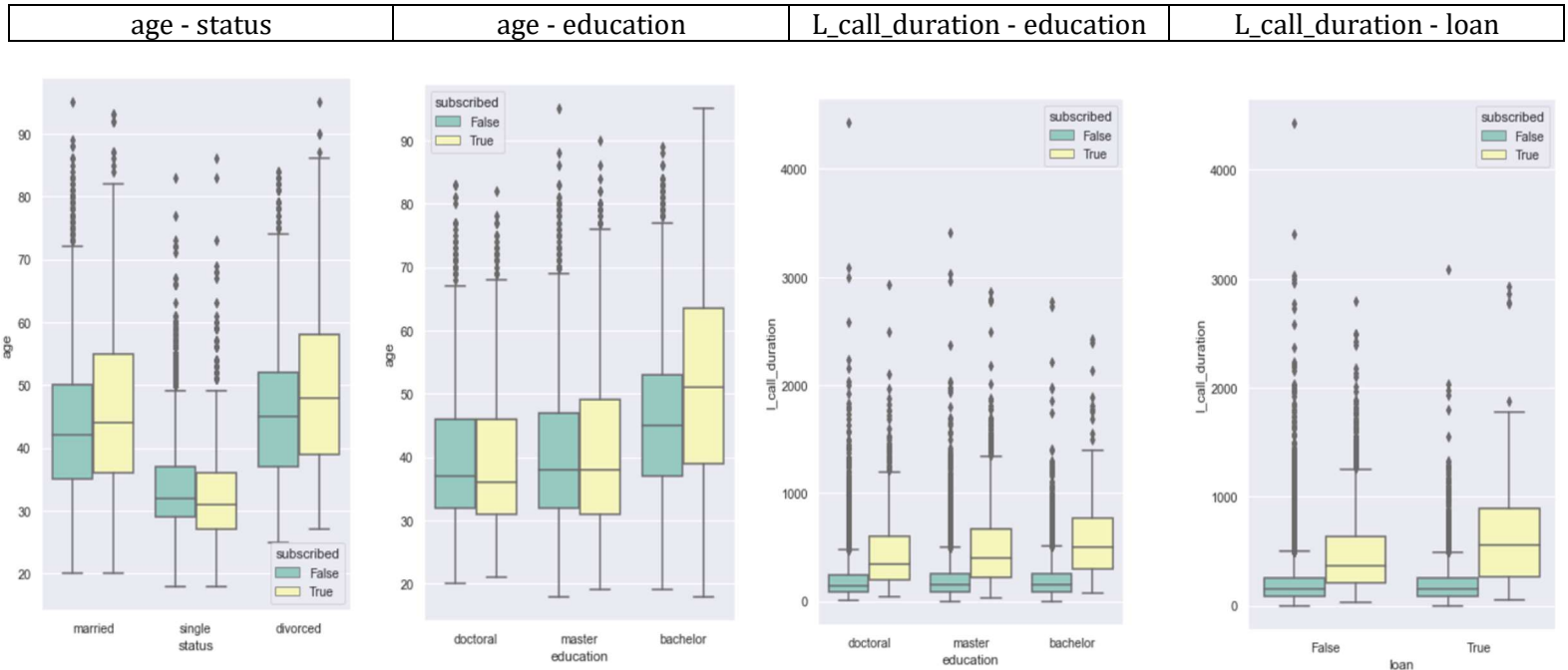
**Subscribed → True = blue**

- אנו משערים כי הזמן נתון בשניות. ניתן לראות בצורה ברורה כי רוב השיחות שלא היו מספיק ארוכות (בין 0 ל- 1000 שניות) באמת לא רכשו ביטוח בסופן, ולעומת זאת ניתן לראות כי שיחות ארוכות יותר (מעל 1000) הסתיימו ברכישת ביטוח. נתון מעניין נוסף שניתן ללמוד מגרף זה שהאחוזים של מי שרכש ביטוח בשיחות שאורכן גדול מ-2000 הוא לא טוב (אזור ה-50%), כלומר צריך לשקול להפסיק לנסות לשכנע בשיחות שאורכן מתארך מעל 2000 מבחינת אחוזים.



בנוסף רואים כי האחוזים של מי שרכש ביטוח בשיחות שאורכן 1000-2000 היה מאוד גבוה.

## Box plot:

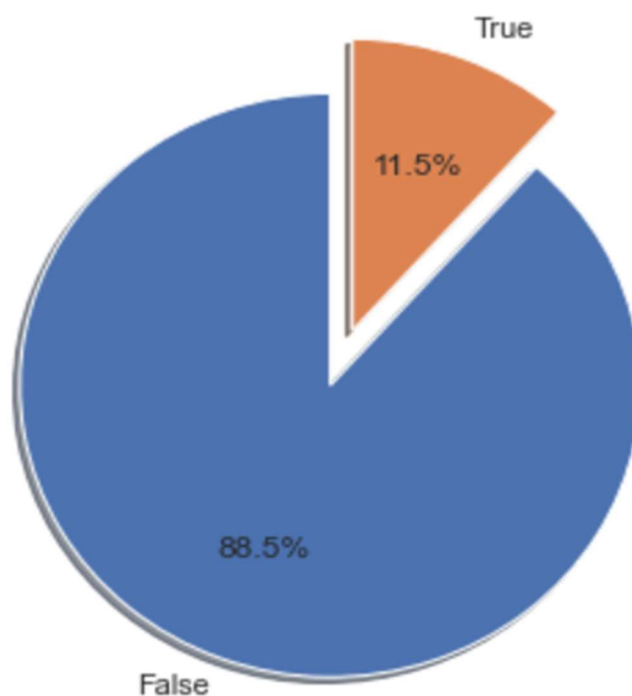


- age-status – ניתן לראות שמי שלא נשוי/גרש שאין שינוי בגיל בין מי שרכש ביטוח או לא רכש ביטוח, לעומת זאת בנשואים/גרשים ניתן לראות כי המבוגרים יותר באותה קבוצה רכשו ביטוח.
- age vs education – ניתן לראות שמי שמבוגר יותר כדי שיהיה יותר סיכוי שירכוש ביטוח עדיף למצוא מבוגר שעושה תואר ראשון, לעומת זאת מי שצעיר כדי שיהיה יותר סיכוי שירכוש ביטוח עדיף למצוא צעיר שעושה דוקטור.
- L\_call\_duration vs education – ניתן לראות שללא קשר לרמת השכלה ככל שהשיחה התארכה יותר כך זה השפיע על רכישת הביטוח.
- L\_call\_duration vs loan – ניתן לראות שללא קשר ללקיחת הלוואה ככל שהשיחה התארכה יותר כך זה השפיע על רכישת הביטוח.



Target:

הצגת התפלגות ערך המטרה – subscribed:

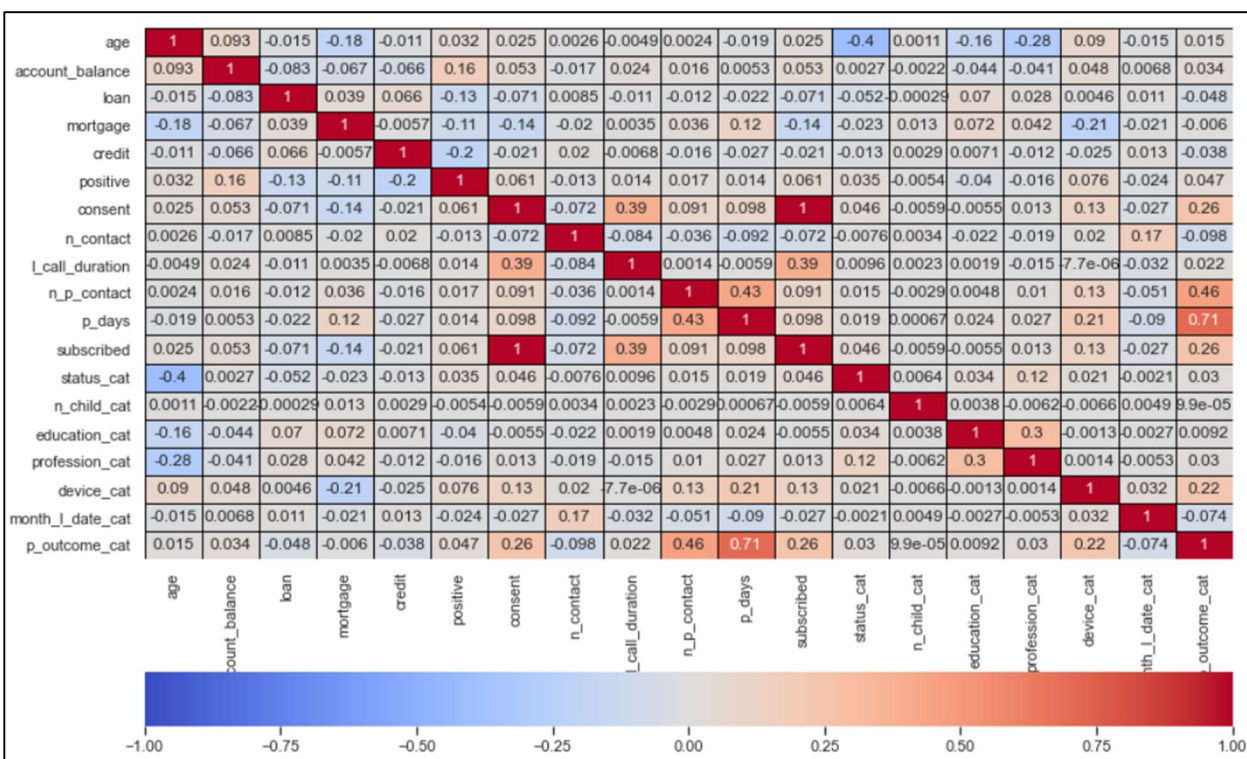


ניתן לראות כי רוב המשתתפים לא רכשו את הביטוח בסופו של דבר.



## שלב 3 - Data correlation

המרנו את הערכים הנומינליים לערכים נומריים ע"י הפקודה `cat.codes` ואז חישבנו את הקורלציה בין כל הפיצ'רים:



- ניתן לראות כי בין רוב הפיצ'רים אין קורלציה גבוה (לא הפוכים ולא מתואמים).
- לפיצ'ר `p_outcome` יש קורלציה יחסית גבוהה עם `n_p_contact` – 0.46.
- לפיצ'ר `p_outcome` יש קורלציה מאוד גבוהה עם `p_days` – 0.71. השתמשו בקורלציה הזאת על מנת להשלים את הערכים החסרים ב `p_outcome`.
- ל `subscribed` ו `consent` יש קורלציה של 1, בהמשך פירטנו מדוע מחקנו את עמודה זו.



---

## שלב 4 - Data cleaning

---

### Pre-Processing

- הערה: בתהליך זה לא מחקנו את העמודות המקוריות, הוספנו עמודות חדשות בשם: xxxxx\_Pre\_Proc (כאשר – xxxxx הוא השם של הפיצ'ר) שבהם מילאנו את הערכים החסרים כתוצאה מהתהליך.

ניתן לראות בקוד כי בתחילה בדקנו באילו שורות יש הכי הרבה ערכים חסרים בפיצ'רים (כלומר יש הכי הרבה null) וראינו כי המקסימום ערכי null בשורה הוא 4. יש 33 שורות כאלו, ומכיוון שמדובר ב 4 פיצ'רים null מתוך 20 החלטנו כי אנחנו לא מוחקים את השורות האלו, כי יש 16 פיצ'רים שיש להם ערכים וזו כמות די גבוהה. כעת נעבור על הפיצ'רים שיש בהם ערכים חסרים ונמלא אותם בשיטה המתאימה ביותר.

### N\_child:

לפיצ'ר זה חסרים המון ערכים לכן שיטת החציון/ממוצע לא נכונה פה. בנוסף השיטה למלא לפי הערך שיש לו רוב גם לא נכונה פה מכיוון שזה לא רוב המידע. כמו כן, לפיצ'ר זה אין קורלציה גבוהה במיוחד עם פיצ'ר אחר אז גם לא ניתן לתקן לפי קורלציה.

כדי לבדוק האם יש עוד משהו שאפשר לעשות, מחקנו את כל השורות שבהן לפיצ'ר זה אין ערך ולאחר מכן בדקנו האם יש קורלציה עם המידע, אך גם פה גילינו שאין קורלציה גבוהה.

לכן אנו מבינים שפיצ'ר זה לא תורם לנו מספיק מידע ובסופו של דבר החלטנו למחוק אותו.



## Education:

לפיצ'ר זה קיימים 1244 שורות שחסרים בו ערכים.  
 1244 זה 4% מסך השורות, ולדעתינו זוהי כמות יחסית גבוהה.  
 לפיצ'ר זה אין קורלציה גבוהה עם פיצ'ר אחר (למעט קורלציה – 0.29 עם profession  
 אך בעייתי לעבוד עם פיצ'ר זה כי גם לו חסרים ערכים).  
 לכן בחרנו לתקן פיצ'ר זה ע"י שיטה של מרחק אוקלידי מינימלי.

```
Filling missing values using similarity between objects

In [45]: education_miss = df[df['education'].isnull()].drop(['status', 'n_child', 'education', 'profession', 'device', 'month_1_date', 'l_date',
education_exist = df[df['education'].notnull()].drop(['status', 'n_child', 'education', 'profession', 'device', 'month_1_date', 'l_date',

In [46]: ary = scipy.spatial.distance.cdist(education_miss, education_exist, metric='euclidean')

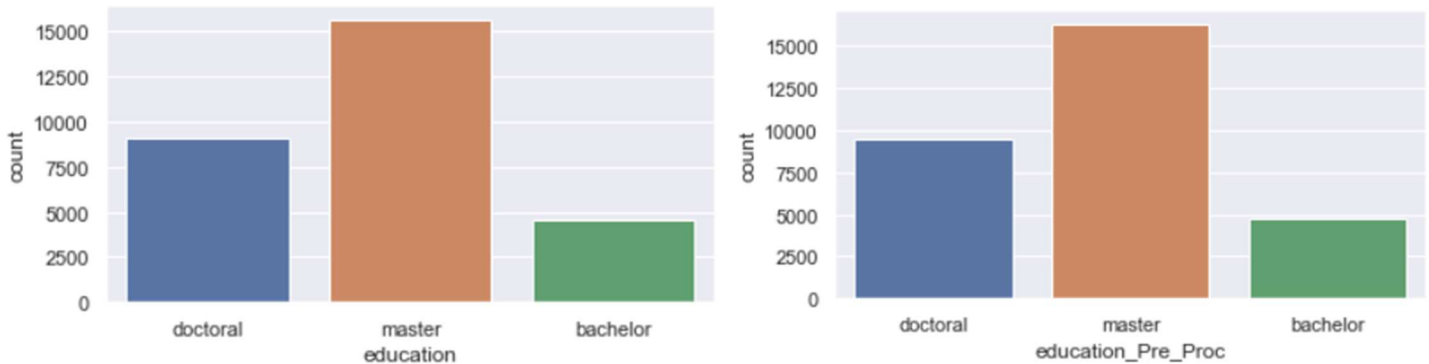
In [47]: df.groupby(['education', 'education_cat']).size()

Out[47]: education  education_cat
bachelor      0                4590
doctoral      1                9061
master        2               15622
dtype: int64

In [48]: df['education_Pre_Proc'] = df['education']
for i, j in enumerate(education_miss.index):
    value = int(df.loc[education_exist[ary[i]==ary[i].min()].index]['education_cat'].mean())
    if value == 0:
        df.loc[j, 'education_Pre_Proc'] = 'bachelor'
    elif value == 1:
        df.loc[j, 'education_Pre_Proc'] = 'doctoral'
    else:
        df.loc[j, 'education_Pre_Proc'] = 'master'
```

יצרנו 2 data frame חדשים:  
 Education\_miss: שבו קיימים כל הערכים הנומרים שיש להם null ב education.  
 Education\_exist: שבו קיימים כל הערכים הנומרים שאין להם null ב education.  
 יצרנו מערך ary שמכיל את המרחקים האוקלידיים בין שני ה data frames.  
 ובעזרתו, ובעזרת מציאת המרחקים הקצרים מילאנו את הערכים החסרים ב education.  
 לסיום הצגנו את שני הגרפים:  
 Education – העמודה המקורית לפני pre processing.  
 Education\_pre\_proc – העמודה לאחר ה pre processing.





ניתן לראות כי אחרי מילוי הערכים החסרים שמרנו על ההתפלגות הצפויה.

### Profession:

תחילה בדקנו כמה ערכים חסרים. חסרים 193 ערכים שזה 0.63% מהמידע.  
בגלל שיש ממש מעט ערכים חסרים החלטנו לתקן פיצ'ר זה לפי החציון.

```
In [52]: df.groupby(['profession', 'profession_cat']).size()
Out[52]: profession      profession_cat
accountant              0              1491
architect              1              833
engineer              2              6538
manager              3              6458
retired              4              988
scientist              5              5138
self-employed         6              882
student              7              1096
teacher              8              2759
technician            9              3503
unemployed           10              638
dtype: int64

In [53]: df['profession_Pre_Proc'] = df['profession_cat']
df['profession_Pre_Proc'].replace(-1, np.nan, inplace = True)

In [54]: df['profession_Pre_Proc'].median()
Out[54]: 3.0

In [55]: df['profession_Pre_Proc'].mean()
Out[55]: 4.480543463922965

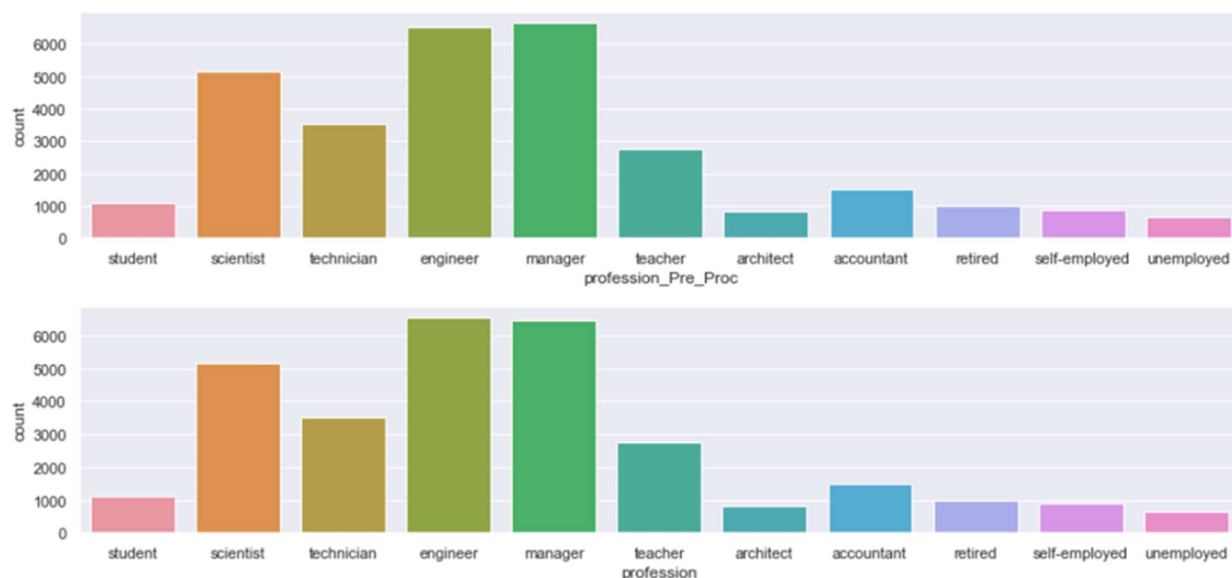
In [56]: #Filling the null values with the median
df['profession_Pre_Proc'] = df['profession_Pre_Proc'].fillna(value=df['profession_Pre_Proc'].median())

In [57]: # Changing the cat.codes back to the origin code:
for i, row in df.iterrows():
    if df.loc[i, 'profession_Pre_Proc'] == 0.0:
        df.loc[i, 'profession_Pre_Proc'] = 'accountant'
```

עשינו קטלוג של כל הערכים ולאחר מכן חישבנו את החציון והממוצע ומילאנו את כל הערכים החסרים בחציון (בגלל שהחציון והממוצע קרובים יחסית), לאחר מכן עשינו תיקון מהערך הנומרי לערך הנומינלי בהתאמה.



הצגנו שני הגרפים של profession לאחר התיקון ולפני התיקון:



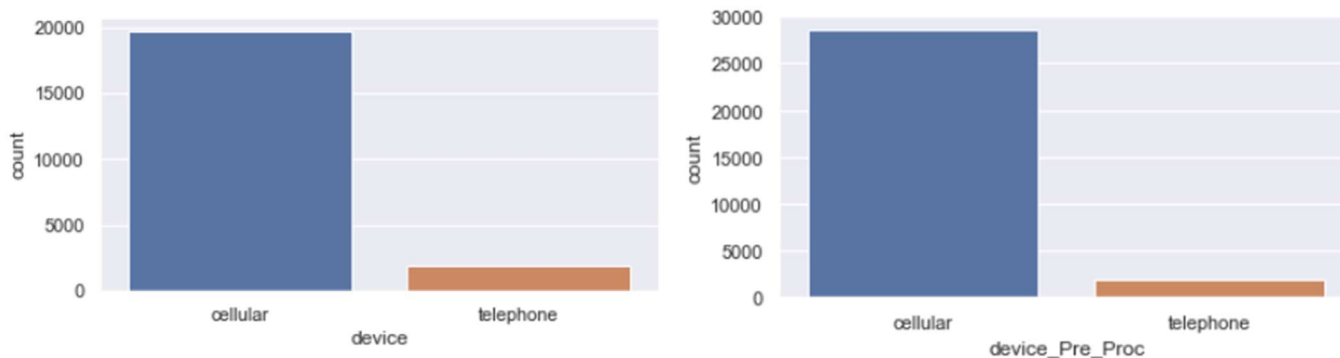
ניתן לראות ש engineer ו manager עדיין מהווים מקסימום מהערכים.

### Device:

קיימים 8871 שורות שבהם חסר הערך device.  
בנוסף, cellular מהווה 65% מהערכים בעמודת ה device.

בגלל שרוב הערכים הם cellular מתוך 2 ערכים החלטנו למלא את כל הערכים החסרים בערך cellular.

הצגת הגרף של device לפני התהליך ואחרי התהליך:





## P\_outcome:

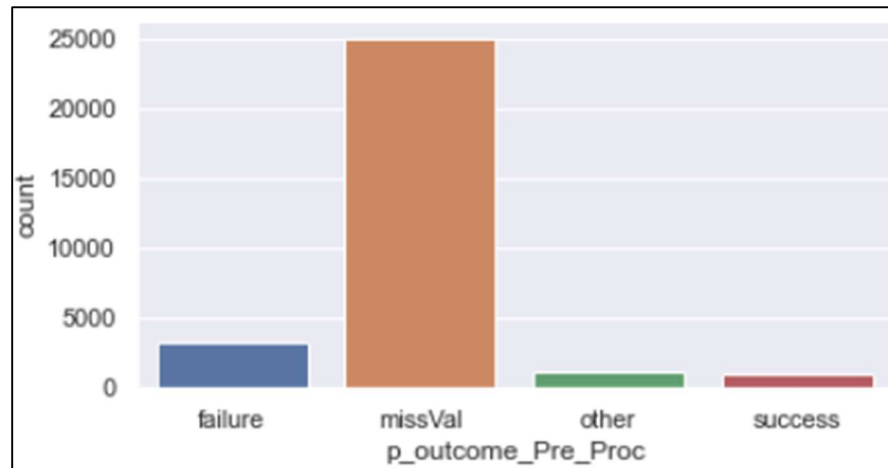
לפיצ'ר זה חסרים המון ערכים – קיימים 24943 שבהם חסר ערך של p\_outcome. תחילה חשבנו שאולי כדאי למחוק את העמודה הזו כי ב 81% מהdata חסר ערך של p\_outcome, אך לאחר בדיקה מעמיקה ניתן לראות כי יש קורלציה מאוד גבוה בין פיצ'ר זה לבין p\_days – קורלציה של 0.7 ובנוסף יש ל p\_outcome קורלציה של 0.26 עם הערך מטרר subscribed. ביצענו בדיקה נוספת להשוואת הערכים בין p\_outcome\_cat ל p\_days:

```
In [68]: df.groupby(['p_outcome_cat', 'p_days']).size()

Out[68]: p_outcome_cat  p_days
-1                    -1      24940
                  98         1
                  168         1
                  528         1
0                    1         3
...
2                   550         1
                  555         1
                  561         1
                  651         1
                  771         1
Length: 1064, dtype: int64
```

ניתן לראות כי ב 24940 מקומות שבהם p\_outcome\_cat הוא -1 (כלומר null) אז p\_days הוא -1. שזה כמעט כל ה data שחסר בו p\_outcome. לכן הבנו שכל הנראה לאנשים שלא השתתפו בקמפיין הקודם הוחלט לשים ב p\_outcome ערך – unknown וגם ב p\_days שמו -1 זאת אומרת שניתן להסיק כי כל הערכים החסרים הם בעצם אנשים אשר לא השתתפו בקמפיין הקודם. ומכיוון שיש קורלציה עם הערך מטרר, החלטנו לא לוותר על העמודה הזו ולמלא את כל הערכים החסרים בערך חדש missVal.

הצגת העמודה p\_outcome\_Pre\_Proc לאחר התהליך:



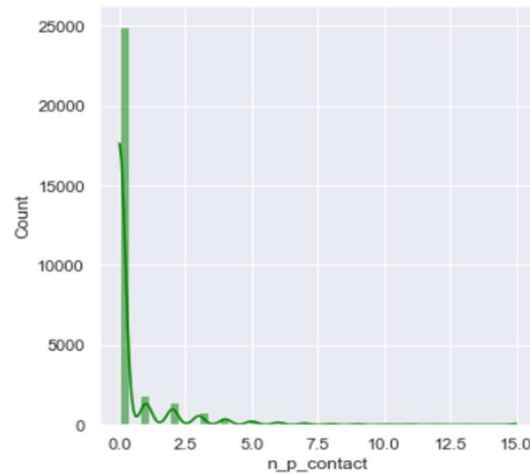
## Outliers

בהסתכלות על הגרפים של הערכים הנומרים ובנוסף מניתוח ה skewness ניתן לראות שלמעט age המתפלג קרוב להתפלגות אחידה, לשאר הערכים יש הרבה ערכים קיצוניים המשפיעים על ההתפלגות. לכן החלטנו לשאר הערכים הנומרים לבצע ניתוח של הערכים הקיצוניים ולשנות אותם בהתאם.

### n\_p\_contact:

בניתוח של הגרף שעשינו ניתן לראות כי רוב מוחלט של הערכים הרלוונטיים לפיצ'ר זה קטנים מ-15 (קיימים רק 78 שורות שבהם יש ערך n\_p\_contact הגדול מ-15 והם מהווים 0,25% מהdata), לכן החלטנו להמיר את כל הערכים שגדולים מ-15 (שזה 419 ערכים) לערך 15.

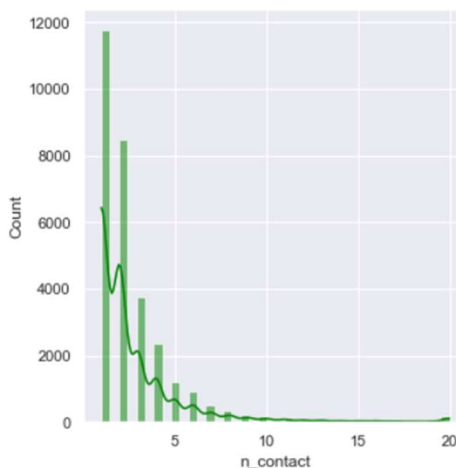
n\_p\_contact לאחר השינוי:



### n\_contact:

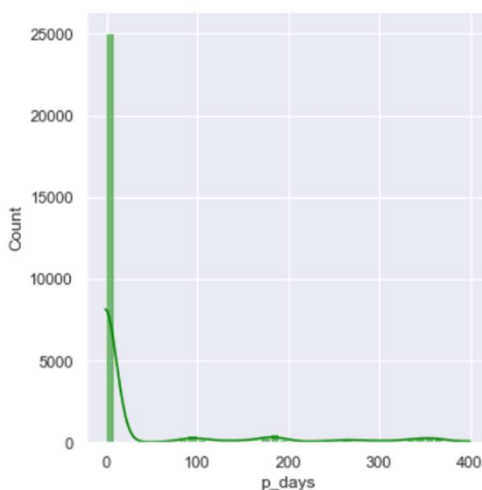
בניתוח של הגרף שעשינו ניתן לראות כי רוב מוחלט של הערכים הרלוונטיים לפיצ'ר זה קטנים מ 20.

רק ל 166 שורות יש ערך n\_contact הגדול מ 20.  
לכן החלטנו להמיר את כל הערכים שגדולים מ 20 לערך 20.  
n\_contact לאחר השינוי:

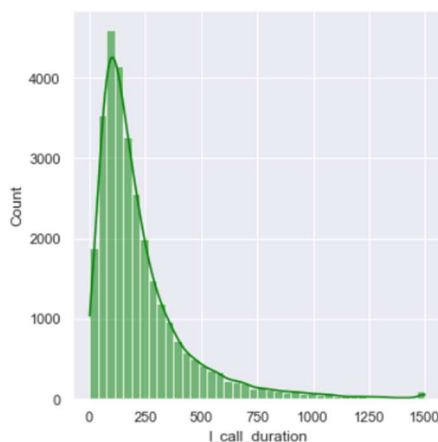


p\_days:

בניתוח של הגרף שעשינו ניתן לראות כי רוב מוחלט של הערכים הרלוונטיים לפיצ'ר זה קטנים מ-400, רק ל-168 שורות יש ערך הגדול מ-400. לכן החלטנו להמיר את כל הערכים שגדולים מ-400 (שזה 164 ערכים) לערך 400. גרף p\_days לאחר השינוי:

l\_call\_duration:

בניתוח של הגרף שעשינו ניתן לראות כי רוב מוחלט של הערכים הרלוונטיים לפיצ'ר זה קטנים מ-1500, רק ל-95 מהערכים יש l\_call\_duration הגדול מ-1500. לכן החלטנו להמיר את כל הערכים שגדולים מ-1500 לערך 1500. גרף l\_call\_duration לאחר השינוי:



מחיקת פיצ'רים שלא תורמים מידע:

החלטנו למחוק עמודות שלא תורמות לנו מידע/המידע שהן מייצגות נמצא כבר בפיצ'רים האחרים ולכן ניתן לוותר עליהן:

- campaign\_type – campaign\_tape זהה לכולם ולכן פיצ'ר זה לא רלוונטי.
- n\_child – מחקנו פיצ'ר זה בתהליך ה pre processing.
- consent – פיצ'ר זה בקורלציה מלאה עם subscribed.

```
subscribed  consent
False       False    26997
True        True     3520
dtype: int64
```

לפי הבנתנו פיצ'ר זה מתאר שמי שרכש ביטוח, הסכים לרכוש ביטוח. אין בפיצ'ר זה ערך מוסף שניתן ללמוד על המשתתפים מדוע הם הסכימו לרכוש ביטוח, ואיך נדע לחזות בעתיד האם לקוח ירכוש בסוף ביטוח או לא ירכוש ביטוח. גם מכיוון שיש 100% התאמה בין הרכישה להסכמה, ניתן לראות שההסכמה נבעה רגע לפי הרכישה, כלומר מילאו את העמודה הזו בעת הרכישה (יחד עם העמודה subscribed), לכן לא הרגשנו שיש טעם להשאיר את העמודה הזו.

---

## שלב 5 - Data reduction

---

### PCA

הרצנו את האלגוריתם PCA לצמצום המימדים.  
בחרנו שה n\_components יהיה 0.999 כלומר אנחנו רצינו לשמר 99% מהשונות. התוצאה שקיבלנו:



	0	1	2
0	-928.818493	-53.216928	259.982810
1	110.456134	91.328429	-13.639327
2	1602.954729	-168.625127	-50.290632
3	-699.128602	-183.322088	-0.030485
4	-2151.092535	-189.384820	-60.042958

## שלב 6 - Data transformation

### Discretization

בשלב זה החלטנו לחלק חלק מהפיצורים בעמודה חדשה המתארת את הפיצור לפי קבוצות. לדעתנו, יהיה יותר נוח להסתכל על ה data לפי קבוצות (עבור פיצורים מסויימים) כי כך יהיה יותר קל לנתח את ה data בהמשך. הפיצורים שלהם החלטנו לחלק לפי קבוצות הן:

- profession
- age
- account\_balance

והסיבה לכך היא שבפיצורים אלה יש התפלגות על הרבה ערכים, והיה נוח לחלקם אותם לפי קבוצות.

#### profession:

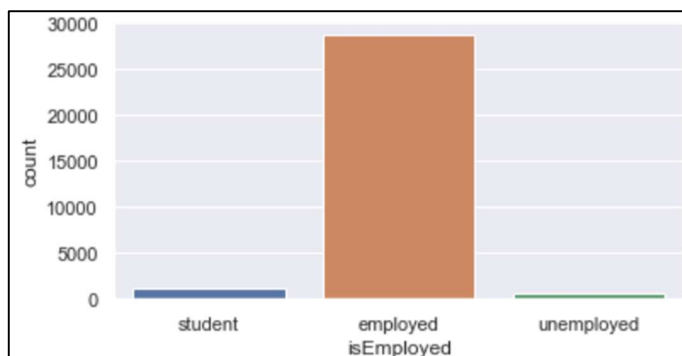
profession הוא פיצור שהכיל הרבה ערכים, כ – 11 ערכים. החלטנו ליצור פיצור חדש שיתן לנו מידע קצת יותר מעניין. לפיצור זה קראנו isEmployed ויש לו 3 שדות:





- Employed.
- Unemployed.
- Student.

בעצם הפיצ'ר החדש מתאר לנו האם הלקוח עובד או לא עובד או סטודנט.  
הצגת גרף isEmployed:



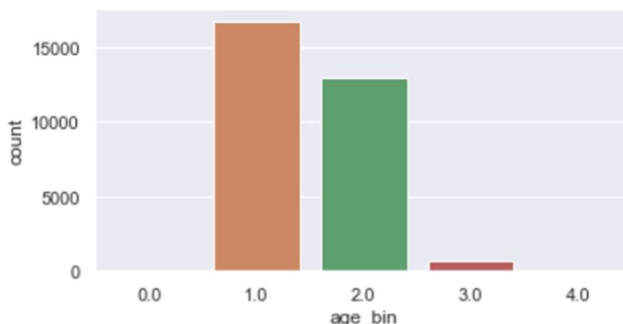
מהפיצ'ר החדש ניתן לראות שרוב הלקוחות אכן עובדים.

age:

החלטנו ליצור פיצ'ר חדש – age\_bin.  
בפיצ'ר זה חילקנו את age לקבוצות גיל:

קבוצת גיל	ערך
$age \leq 20$	0
$20 < age \leq 40$	1
$40 < age \leq 60$	2
$60 < age \leq 80$	3
$80 < age$	4

age\_bin:



מהפיצ'ר החדש ניתן לראות שהקבוצת גיל הגבוהה ביותר היא בין 20 ל – 40.

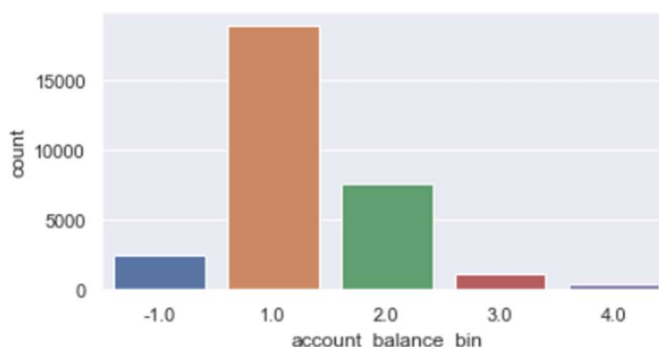
### account balance:

החלטנו ליצור פיצ'ר חדש – `account_balance_bin`.

בפיצ'ר זה חילקנו את `account_balance_bin` לקבוצות משכורת:

קבוצת גיל	ערך
$account\ balance \leq 0$	-1
$0 < account\ balance \leq 1000$	1
$1000 < account\ balance \leq 5000$	2
$5000 < account\ balance \leq 10000$	3
$10000 < account\ balance$	4

`account_balance_bin`:



מהפיצ'ר החדש ניתן לראות שהמרב חשבון של רוב הלקוחות הוא בין 0 ל 1000.



## MinMax normalization

בשלב זה בנינו עמודות נומריות חדשות מנורמלות בין 0 ל-1. הסיבה להוספת עמודות אלו היא שאם בהמשך יהיה לנו צורך בחישובי אלגוריתמים המתבססים על כך שה data הנומרי מתפלג על אותו ציר מספרים, אז נוכל להשתמש בעמודות מנורמלות אלה.

כמו כן נירמלנו את כל העמודות על פי אותו אלגוריתם – minmax.

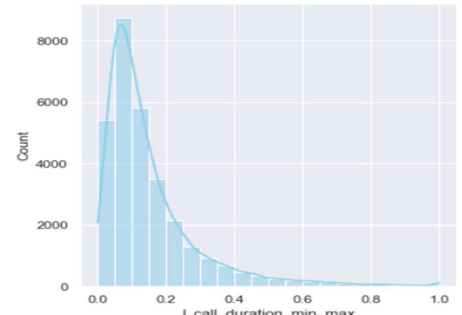
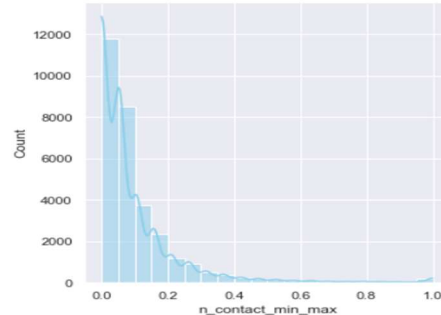
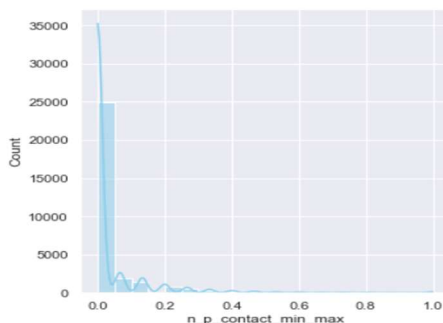
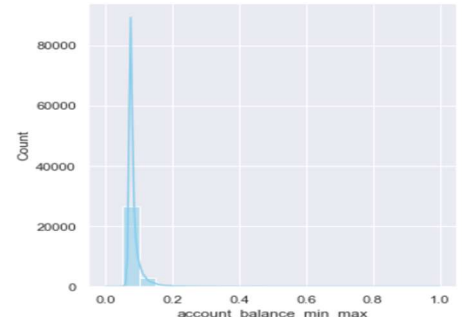
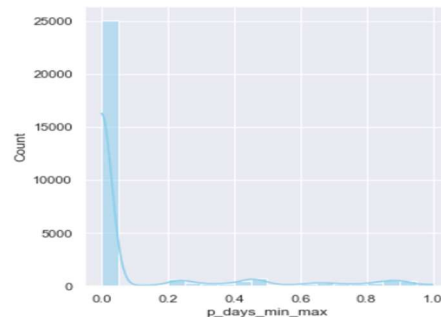
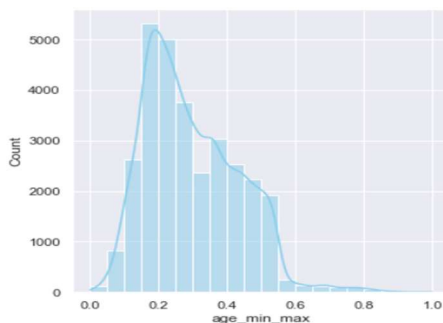
העמודות החדשות מסומנות כ- xxx\_min\_max כאשר xxx הוא שם הפיצ'ר.

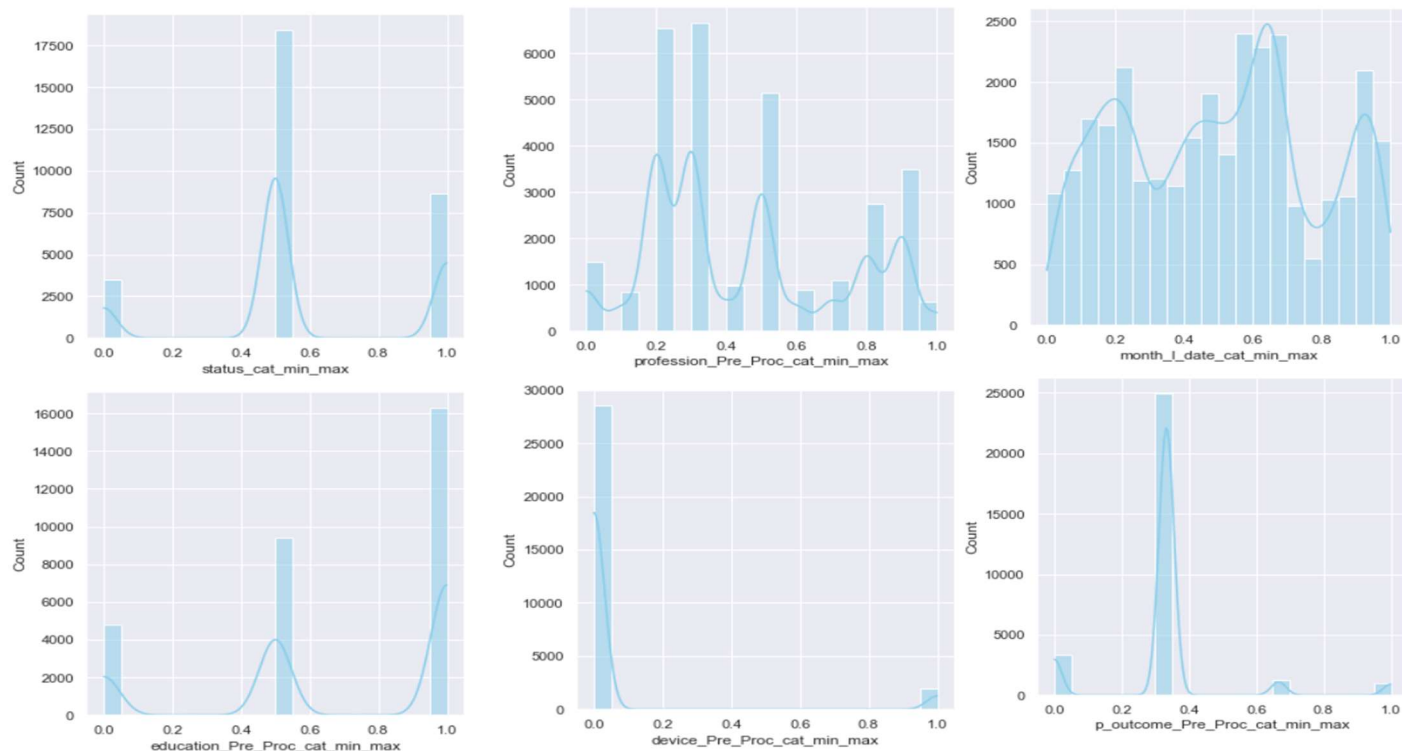
העמודות שנרמלנו הן:

age  
n\_p\_contact  
p\_days  
n\_contact  
account\_balance  
l\_call\_duration

status\_cat  
education\_Pre\_Proc\_cat  
profession\_Pre\_Proc\_cat  
device\_Pre\_Proc\_cat  
month\_l\_date\_cat  
p\_outcome\_Pre\_Proc\_cat

הצגת גרפים לאחר נרמול:





ניתן לראות שהטווח של הערכים השתנה בין 0 ל 1.  
ובנוסף הגרפים נראים בצורה דומה לגרפים המקוריים.