**ניתוח קבצים**

ערך NaN – נתונים חסרים. Not A Number

ערכים חסרים:

MCAR – חסר באופן רנדומלי

MAR – חסר באופן רנדומלי תלוי במשתנה מסויים, אחר.

MNAR – חסר לא בצורה רנדומלית. לא ברור למה חסרים.

*איתור ערכים חסרים*

סקירה ראשונית: שימוש ב-pandas

Df.columns – שמות המשתנים/ עמודות/ מאפיינים.

Df.dtypes – סוגי המשתנים – מספר שלם/ עשרוני/ נתונים מוערבים(טקסט)

Df.head() – לראות את השורות הראשונות של הטבלה.

Df.info() – כמה שורות יש בטבלה, לכל עמודה כמה שורות מכילות ערך.

Df.describe(include='all') – השורות בתוצאות:

Count – מספר הערכים הלא חסרים בכל עמודה

Unique – מספר ערכים חד פעמיים

Top – השכיח / Freq – שכיחות השכיח – אם הערך השכיח מופיע בכל השורות – למעשה בעייתי / אין נתונים.

Mean - ממוצע

Std – סטיית תקן

Df.<column name>.head().isnull() – אפשר על כל הטבלה או עמודה – מחזירה bool על כל ערך True אם חסר.

ניתן לעבוד על head() – כדי לצמצם גודל הקלט. בשביל זריזות.

על כל העמודה/טבלה מדובר בהמון תוצאות בוליאני...

Df.<column name>.isnull().sum() – סוכם את מספר הערכים החסרים בעמודה.

Df.isnull().sum() – מחזיר את הסכום של כל הערכים החסרים בכל העמודות.

**נתונים חסרים**

Df.dropna(axis=0) – axis מה למחוק. 0 (ברירת מחדל = מחיקת שורה)/ 1 מחיקת עמודה

פרמטר נוסף how='all'/'any' : all – אם כל הערכים חסרים. Any – אם חסר ערך כלשהו

Thresh = x – השורה/עמודה תמחק רק אם מספר הערכים התקינים שווה או גדול ל-thresh

צריך לשמור את התוצאה כדי להמשיך

Df\_clean=df.dropna…copy()

או לבצע על ה-DF עצמו. Df = df.dropna…

אופציה נוספת שימוש בפרמטר inplace=True

Df.dropna(axis=0, thresh=11, inplace=True)

כדי לא לאבד נתונים. ננסה להכניס נתונים במקום החסרים. בהתאם לסוג העמודה.

Df\_new\_column=Df2.<column name>.fillna(0) סדרה הערכים מלאים. הצבנו 0 בכל מקום שחסר

Df\_new\_column=Df2.<column name>.fillna(df2<column name.mean()) סדרה הערכים מלאים. הצבנו ערך ממוצע של העמודה, בכל מקום שחסר ככה לא ישפיע על הממוצע

או ,median() – חציון של העמודה

Fdf2.fillna(df2.median(), inplace=True) – מילוי כל הערכים בכל הטבלה לפי חציון, בערכים רציפים

למשתנים שמים/סדר – נכניס את הערך השכיח Mode/Mo

Df.<column> = df.<column>.fillna(df.<column>.mode()[0])

df.fillna(method='ffill') – ממלאים ערכים ריקים עם ערכים תקינים משורה שלפניה. שורה ראשונה לא תושפע.

df.fillna(method='bfill') – ממלאים ערכים ריקים עם ערכים תקינים משורה אחריה. שורה אחרונה לא תושפע.

Bfill/ffill מתאימים יותר לערכים רציפים שנאספים לאורך זמן... נגיד טמפרטורה שעתית/ יומית.

**כפילויות נתונים**

איתור

Df2.duplicated() – שורות זהות לחלוטין. T/F לכל שורה. True לשורה שזהה לשורה כלשהי מעליה.

Df2.duplicated().sum() – סוכם את מספר השורות True/.

Df2[Df2.duplicated()] – נראה את השורות הכפולות.

Df2[Df2.duplicated()].shape – נקבל את "צורת" הטבלה. כמה שורות/ עמודות

Df2[Df2.duplicated([columns name])] – יכול לבדוק שורות כפולות לפי עמודות ספציפיות

טיפול

Df2.drop\_duplicates() – ברירת מחדל מוחק שורות כפולות זהות לחלוטין

Df2.drop\_duplicates(subset=['Sex']) – ברירת מחדל מוחק שורות כפולות בהתחשב בעמודות המצויינות ב-subset – ברירת מחדל – שומר את השורה הראשונה.

Df2.drop\_duplicates(subset=['Sex'], keep='First'] – לבחור איזו שורה כפולה לשמור.

אפשר גם 'last' – לשמור את הרשומה האחרונה

Keep='false' – לא שומרים שום שורה כפולה.

**נתונים חריגים**

איתור

Df.describe(include='all')

ממוצע/ חציון/ אחוזון 25/75 (משמשים ל-IQR), סטיית תקן...

* איתור בגרפים –

plt.hist(df.Fare, bins=50)

plt.xlabel("Fare")/ plt.ylabel("Frequency")

נזהה קבוצות עם ערכים חריגים

Sum(df.Fare>X) – סיכום מספר השורות עם ערכים חריגים.

* מרחק מחציון (כלל IQR)

אחוזון 25 = Q1/ חציון (50%) = Q2/ אחוזון 75 = Q3

IQR – התחום בין Q1 ל – Q3 (מכיל 50% מהערכים).

החריגים יחשבו קטנים מ – Q1 – 1.5IQR ו/או/ הגדולים מ- Q3 + 1.5IQR

יש מי שמגדיל ל - +-2IQR

תצוגה גרפית seaborn.BoxPlot.

Import seaborn as sns

Sns.boxplot(df.column)

אפשר לשמור את האינדקסים של שורות החריגים לפי Q1/Q3 שנחשב ידנית:

Q1 = np.percentile(df.column, 25)

Q3 = np.percentile(df.column, 75)

IQR = Q3 – Q1

Column\_name\_outliers\_index = df[(df.column\_name < Q1 – 1.5\*IQR | df.column\_name > Q3 + 1.5\*IQR].index

ואפשר לבדוק עם len על הרשימה, לראות כמה שורות חריגות יש.

אפשר לבדוק על כל עמודה אינדקס שורות חריגות ולזכור את כל השורות המכילות ערכים חריגים.

ואז לבדוק עם set, אם יש אינדקסים כפולים, כלומר שורות עם מספר ערכים חריגים.

* מרחקי סטיית תקן מ-הממוצע.

מרחק של 3 סטיות תקן מהממוצע. מתאים למשתנים שמניחים שמתפלגים נורמלית.

Z\_score = (df.column – df.column.mean()) / df.column.std()

Outliers = abs(z\_score) > 3

Sum(outliers)

אפשר לבדוק min / max עם min(df.columns[outliers]) לדוגמה.

טיפול

אם הערכים הגיעו עם הנתונים כרגיל – נשאיר אותם.

אם מחליטים למחוק – נשנה אותם ל-NaN ונטפל בהם לפי טיפול בנתונים חסרים.

Df.column[df.column>X] = np.nan

ואז שימוש ב-describe() לראות מה עוללנו.

אפשר גם להציב NaN לפי 1.5IQR.

**המרת נתונים**

טיפול במשתנה תקינים. נגיד ערכים מילוליים – יכול להיות שיהיה לנו קל יותר לעבוד עם מספרים.

מספרים רציפים אולי נעדיף לעבוד עם X קטגוריות.

בידוד וקידוד בינארי...

Male/female – המרה ל- 0/1

אופציה נוספת df.Sex = df.Sex.astype('category')

Replace\_map = {'female':1, 'male':2}

Df.replace(replace\_map, inplace=True)

אופציה נוספת

Df\_test = pd.get\_dummies(df, columns=['Sex'], prefix=['Sex'])

יצור טבלה חדשה. יוריד את העמודה sex

הטבלה החדשה תכיל 2 עמודות חדשות sex\_female, sex\_male שיכילו 1/0 לפי מין האדם באותה שורה.  
מכיוון שסכום משתני dummies בשורה שווים ל – 1, ניתן לוותר על עמודה 1.

לקידוד ערכי סדר – ניתן לבנות מספר עמודות, לייצוג כמספר בינארי.