פקודות לניתוח נתונים

בתחילת הקובץ נרשום:

import numpy as np – מייבא את ספריית נאמפי (ספרייה המאפשרת לעבוד עם מערכים רב-מימדיים, ומספקת פונקציות מתמטיות לעבודה עם המערכים).

import pandas as pd – מייבא את ספריית פנדס (ספרייה שיש לה אוסף של פונקציות שמיועדות לטיפול בנתונים).

שם הפקודה	מה היא עושה	דגשים / הערות
pd.read_csv	קריאת קובץ csv	אם זה שם קובץ אחר אז נכתוב בהתאם (json)
.info()	מידע בסיסי, סוג הנתונים על הקובץ	
.columns	שמות של העמודות	
.shape	נקבל את הצורה של הקובץ בסוגריים, צפייה במספר העמודות והשורות של הקובץ.	ייכתב בסדר הבא - (עמודות, שורה)
.describe()	מראה מידע סטטיסטי על כל אחד מהעמודות	
type()	סוג הקובץ	
.head()	נותן את השורות הראשונות של הקובץ	אם לא מכניסים ערך לתוך הסוגריים זה ייתן את ה 5 שורות הראשונות
.teal()	נותן את השורות האחרונות של הקובץ	אם לא מכניסים ערך לתוך הסוגריים זה ייתן את ה 5 שורות האחרונות
.iloc[]	תפקידה לבחור נתונים על סמך מיקומם בטבלה - אינדקס השורות והעמודות. לכן היא מקבלת רק מספרים בתור פרמטר.	לא כולל המספר האחרון [0:100] >99
	iloc[[-1,2,45]]. אפשר כמה שורות iloc[[0,3,6,24],[0,5,6]]. (צד ימין - העמודות, ושמאל - השורות)	[עמודות,:שורות] [3:,5]לשים לב לפסיק
.loc[]	מאפשרת לבחור שורות לפי האינדקס ועמודות לפי שם העמודה.	כולל המספר האחרון [0:100]> 100
.count()	מספר השורות המכילות ערך	
.isnull()	הצגת שורות המכילות ערכים חסרים	null \ na – אין ערך. 2 השמות אותו פירוש
.notnull()	הצגת שורות שאינן מכילות ערכים חסרים	
.sum()	סוכם את כל האפשרויות	אם לא נשתמש במתודה sum נקבל True \ False
.dropna()	הסר שורות עם ערכים חסרים בהתאם לפרמטרים ברירת מחדל – שומט שורות אם לפחות בעמודה אחת יש NuN	
.dropna(how='all')	שומט שורות רק אם בכל העמודות שלה יש NuNs	
.dropna(k=)	שומט שורות אם יש כמות ערכים חסרים בשורה (אני קובעת כמות ערכים=K)	K= מספר ערכים
.dropna(axis=1)	הסרה של עמודות במקום שורות	

שם הפקודה	מה היא עושה	דגשים / הערות
.fillna()	השלמת ערכים חסרים (בדר"כ 0 או 1-)	
inplace	"תכניס במקום", כמו לעשות השמה. (מתי שרוצים שזה יופיע בדאטה פריים)	
ffill	ממלא את הערך החסר בערך שהיה לפניו	
bfill	ממלא את הערך החסר בערך שהיה אחריו	
.groupby()	מקבץ את הרשומות לקבוצות (ע"פ מה לקבץ את הרשומות – ע"פ העמודה שאני בחרתי)	
.apply()	הפעלת פונקציה על עמודה	
.copy()	נותן לעבוד על חלק/עותק משלי בדאטה פריים, בלי לפגוע בדאטה המקורי.	
.max()	מקסימום	לא יעבוד אם יש ערכים חסרים (Nan)
.interpolate()	אינטרפולציה של ערכים על פי שיטות שונות	

היסטוגרמה

שם הפקודה	מה היא עושה	דגשים / הערות
.hist()	יצירת היסטוגרמה	
.astype()	המרת סוג משתנה (לדוגמא int)	
.seed()	שחזור נתונים	
.plot()	מאחד נתונים להיסטוגרמה משותפת. מקבלים את הנתונים בסוג של קווים לאורך	
.sort_values	ממיין. סידור סט הנתונים לפי עמודה (בסדר יורד או עולה)	
bins	כמות טווחים רצויה	
alpha	נותן שקיפות מה שעוזר לנו כשאנחנו רוצים לראות את הנתונים החופפים בין הקבוצות	
nu	שם משתנה לממוצע	
sigma	שם משתנה לסטיית תקן	
sample	שם משתמש למדגם בגודל מסוים	

שם הפקודה	מה היא עושה	דגשים / הערות
.corr()	קורלציה	
.plot.scatter()	מאפשרת להציג תרשים פיזור (של נקודות) שליטה במראה של התרשים: color= - שליטה בצבע alpha= - שקיפות בערכים שנעים בין 0 ל-1 marke= - זו הצורה	שייכת לספריית פנדס scatter plot – עובד עם ערכים חסרים גם
import matplotlib.pyplot as plt – הספרייה matplotlib מציעה מגוון של עיצובים מוכנים מראש שכדאי להשתמש בהם כדי להקנות לתרשימים שלנו מראה מקצועי. היא הספרייה האם של seaborn המשמשת להצגת גרפים ותרשימים. (את ה scatter רושמים אחרת בספרייה זו)		
import seaborn as sns – ספריית Seaborn מבוססת על Matplotlib, ומאפשרת ליצור תרשימים אטרקטיביים ומשוכללים בממשק ידידותי הרבה יותר.		
sns.regplot()	הוספת קו רגרסיה ל scatter plot	
סוגי קורלציות	'person' , ' spearman' , 'kendall'	פרסון טוב למקרים שהדאטה הגיע מאיזושהי התפלגות נורמלית
pd.plotting.scatter_matrix	מטריצה	
sns.pairplot()	מאפשר לתאר את היחסים בין העמודות המספריות של מסד הנתונים.	
sns.heatmap()	מאפשרות לנו לייצג בקלות רבה מטריצות של נתונים. טבלאות ציר ומטריצות של קורלציות. מראה את הקורלציה בצורת מספרים	
annot=True	מציג את המספרים בתוך התאים	במקרה של False הנתונים לא יופיעו.
cmap=	סכימת הצבעים	ככל שהצבע יותר כהה(חזק) כך הקורלציה יותר גבוהה
.round()	כמה ספרות להציג אחרי הנקודה	
.style.background_gradient()	משנה את הצבע	
.subplots_adjust()	כוונן את פריסת חלקת המשנה	
.reset_index(drop=true)	מחזיר את האינדקס המקורי	
.figure()	צור דמות חדשה, או הפעל דמות קיימת	
.plot.density()	קובע את הצפיפות של הציור	
sns.regplot()	נתוני עלילה ומודל רגרסיה ליניארי מתאימים	
plt.subplots_adjust ()	מכוון את הרווח בין החלקות.	

שם הפקודה	מה היא עושה	דגשים / הערות
sns.countplot()	מציג את מספר הדוגמאות מכל סוג בקטגוריה.	לדוגמה, מספר הזכרים והנקבות במסד הנתונים.
plt.show()	מראה את הקוד	
pd.get_dummies()	יוצר עמודות חדשות על עמודות קיימות	לדוגמא: עמודה של מין – הפקודה תיצור לי עמודה של זכר ועמודה של נקבה
.Marital_Status		
.aggregate()	צבירה באמצעות פעולה אחת או יותר מעל הציר שצוין.	
.unstack()	שנה את ההיררכיה של תיעוד הנתונים	
.pivot_table()	צור טבלת ציר לקיבוץ ולסיכום תיעוד הנתונים	
import sklearn as sk – sklearn הוא ספרייה מפייתון, המספק כלים שונים להתאמת מודלים, עיבוד מקדים של נתונים, בחירת מודלים והערכתם, ותכשירים רבים אחרים. from sklearn import tree - המטרה היא ליצור מודל שמנבא את הערך של משתנה יעד על ידי למידה של כללי החלטה פשוטים שמקורם בתכונות הנתונים. ניתן לראות בעץ קירוב קבוע באופן חלקי - from sklearn.tree import DecisionTreeClassifier		
.tree.DecisionTreeClassifier()	מודל מסוג עץ	
.title()	כותרת לתרשים	
.value_counts()	סופר כמה ערכים יש לי מכל סוג	
.plot.pie()	תרשים עוגה. לוקח את המספר והופך אותו לאחוזים. (מחלק את העיגולים לאחוזים)	
_feature_importances.	בודק מה מכל הפיצ'רים הם החשובים, ומשאיר אותם. מוריד כפילויות ומשאיר את החשובים.	
tree.plot_tree()	יצירת עץ החלטה	
.export_text()	בנה דוח טקסט המציג את הכללים של עץ החלטה	
.fit()	תאמן את המודל, קח את המודל מסוג החלטה ותאמן אותו עם הדאטה () שרשום בתוך הסוגריים. בונה עץ החלטה = לאמן	
.predict()	המנבא. בניית מודל עץ ההחלטה על מערך הנתונים שלנו	
sk.model_selection.train_test_split()	פיצול	

שם הפקודה	מה היא עושה	דגשים / הערות
sk.metrics.accuracy_score()	פונקציה זו מחשבת דיוק של תת קבוצות: קבוצת התוויות החזויות לדוגמא חייבת להתאים במדויק לקבוצת התוויות המתאימה ב- y_true.	
sk.metrics.plot_confusion_matrix()		
sk.metrics.classification_report()	בנה דוח טקסט המציג את מדדי הסיווג העיקריים	
.classes_		
from sklearn.svm import SVC – מודל לינארי. המטרה של SVC לינארי (סיווג וקטור תמיכה) היא להתאים לנתונים שאתה מספק, להחזיר hyperplan "בכושר הטוב ביותר" המחלק או מסווג את הנתונים שלך. משם, לאחר קבלת ה-hyperplane, תוכל להזין כמה תכונות למסווג שלך כדי לראות מה המחלקה "החזויה". **from sklearn.datasets import load_digits** – דאטה שמנסה לזהות מספרים. טען והחזר את מערך הספרות (סיווג) **from sklearn.ensemble import RandomForestClassifier-** יער אקראי מתאים למספר מסווגי עץ החלטות על דגימות משנה שונות של מערך הנתונים ומשתמש בממוצע לשיפור דיוק הניבוי ובקרת התאמת יתר. גודל מדגם המשנה נשלט באמצעות max_samplesהפרמטר אם bootstrap=True(ברירת מחדל), אחרת כל מערך הנתונים משמש לבניית כל עץ.		
max_depth	מקסימום עומק. משתנה שמכניסים לפונקציה, ומגדיר לעץ החלטה עד איזה עומק להגיע.	
.target		

איך עושים פעולות מסוימות

הפעולה	תיאור הפעולה
הוצאה של עמודה ספציפית	*הוצאה של העמודה ע"י נקודה ושם העמודה. *אם שם העמודה מכיל יותר ממילה אחת וסימנים כלשהם – נכניס את שם העמודה לסוגריים מרובעים ונתחום אותה בגרשיים [' '] *מדפיס את ה 5 העמודות הראשונות והאחרונות של הקובץ
עבודה על חלק מהדאטה פריים	לעשות ()copy.

NuN – לא קיים ערך. אין אינפורמציה בקובץ המקורי.

בתרשים עץ – True הולך שמאלה, ו -False הולך ימינה.

---

markdown

עיקרי שפת ה markdown הם:

- כל מה שנכתב מתורגם כטקסט רגיל
- תחילת שורה ב # מהווה כותרת ראשונה
- תחילת שורה ב ## מהווה כותרת שניה , וכו'
- תחילת שורה ב * יתורגם כ bullet
- סימון מילה ב **text** יתורגם כ bold text
- סימון מילה ב *text* יתורגם כ italic text

רשימה של קיצורי הדרך הכי שימושיים:

Ctrl+Enter : הרצת שורת קוד מסומנת

Ctrl+Shift+Enter : הרצת chunk של קוד

-Alt : כתיבת אופרטור השמה

Ctrl+Shift+M : כתיבת אופרטור שרשור

Ctrl+Shift+C : סימון שורה בתור הערה

Alt+Shift+K : קיצורי דרך

Ctrl+L : ניקוי הקונסול

F1 : קבלת עזרה על פקודה

tab : השלמה אוטומטית בזמן כתיבה