

**המחלקה למדעי המידע**

**Machine Learning – פרויקט סיכום**

שם קורס: Machine Learning

מספר קורס: 35887-01

שנה אקדמית: תש"פ

סמסטר (בו ניתן הקורס): ב'

תאריך הגשה: 7.9.2020

מוגש למרצה: ד"ר אבשלום אלימלך

שם הסטודנטית: מאשה שמידוב

ת.ז. 320908361

טלפון: 054-2185432 כתובת מייל: mashmidov@gmail.com‏

שאלה 1:

בהתחלה הפעלת dataset.isnull().sum() כדי לראות באיזה עמודות יש חוסרים. לפי התוצאה נראה שאין חוסרים, אבל לאחר ריפרוף בעמודות החוסרים הם לא תאים ריקים אלה עם סימן "?" ולכן לא עלו בבדיקה.

ביצעתי החלפה של סימן "?" בNaN על ידי replace:

dataset2=dataset.replace('?',"NaN")

ושמרתי את זה בdataset2 ועליו בדקתי האם ניתן לראות איפה יש חסורים וכמה. לצערי באופן הזה קיבלתי שיש אותם נתונים בכל שדה.(מבחינת מספר ואין חוסרים).

שלב נוסף ניסיתי להחליף את "?" במקום ריק בתקווה שזה כן יעזור לי לראות איפה יש חסורים במקום לעשות את זה בקובץ אקסל:

dataset4=dataset.replace(‘?’,” “)

אבל גם זה החזיר לי תמיד את אותה תשובה הבאה למרות שאני יודעת בודאות שבעמודה B יש לי חסורים:

dataset4.info()

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 205 entries, 0 to 204

Data columns (total 25 columns):

symboling 205 non-null int64

normalized-losses 205 non-null object

fuel 205 non-null object

aspiration 205 non-null object

num-of-doors 205 non-null object

לכן החלטתי לעשות את הreplace בתוך האקסל, הכנתי העתק של האקסל המקורי בשם: autos\_copy.csv

ולאחר מכן הרצתי עליו info() ושם כבר ניתן היה לראות שיש חוסרים בשדות.

dataset.info()

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 205 entries, 0 to 204

Data columns (total 25 columns):

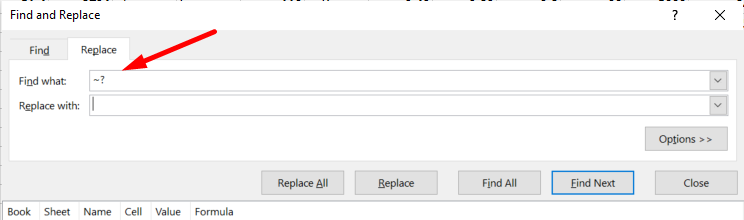
symboling 205 non-null int64

normalized-losses 164 non-null float64

fuel 205 non-null object

aspiration 205 non-null object

כדי להחליף את כל הסימני שאלה בקובץ אקסל בתיבת חיפוש רשמתי ~? והחלפתי את זה במקום ריק:



אחרי הפעולה כבר ניתן היה לראות באיזה עמודות יש חוסרים:

dataset.info()

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 205 entries, 0 to 204

Data columns (total 25 columns):

symboling 205 non-null int64

normalized-losses 164 non-null float64

fuel 205 non-null object

aspiration 205 non-null object

num-of-doors 203 non-null object

body-type 205 non-null object

drive-wheels 205 non-null object

engine-location 205 non-null object

wheel 205 non-null float64

length 205 non-null float64

width 205 non-null float64

height 205 non-null float64

curb-weight 205 non-null int64

engine-type 205 non-null object

cylinders 205 non-null object

engine-size 205 non-null int64

fuel-system 205 non-null object

bore 201 non-null float64

stroke 201 non-null float64

compression 205 non-null float64

horsepower 203 non-null float64

peak 203 non-null float64

city-mpg 205 non-null int64

highway-mpg 205 non-null int64

price 201 non-null float64

dtypes: float64(11), int64(5), object(9)

לאחר פעולה זו ניתן גם לראות איזה סוג נתון חסר.

שלב 2:

מכיוון שחסר לי גם מספרים וגם טקסט חשבתי להשלים בבת אחת את כולם על ידי שימוש ב fillna ולקחת את הנתון החסר משורה לפני:

dataset2=dataset.fillna(method='bfill')

שלב 3:

לאחר השלמת חוסרים גזרתי מהDS הזה את משתנה התלוי שלי:

y = dataset2.iloc [:, -1 ].values #dependent variable

x = dataset2.iloc [:, :-1 ].values #Independent variable

לאחר מכן הוספתי לכל עמודה שיש בה קטגוריה מילולית- קטגוריה מספריתעל ידי LabelEncoder.

כל שורה עשיתי בנפרד (שורות 20-29) עשיתי את המספור גם לעמודה H למרות שהערך בה היה זהה לכל השורות (התלבטתי אם למחוק אותה אבל בסוף השארתי).

שלב 5: dummy encoding

לאחר מכן ניסיתי לבטל את התלות בין השדות כפי שלמדנו בעזרת oneHotEncoder אבל לא הצלחתי.

onehotencoder = OneHotEncoder(categorical\_features = [0])

x =onehotencoder.fit\_transform(x).toarray()

הוספתי לכל עמודה שעשיתי בה מספור עמודות נוספות שלא תהיה תלות בין המשתנים (במקום onehotencoder בעזתר get\_dummies).

a=pd.DataFrame(x) #mishtane bilti talui as dataframe

x=pd.get\_dummies(a, columns=[2, 3, 4, 5, 6, 7, 13, 14,16])

ראיתי שאין צורך למחוק עמודה מיותרת עבור על יצירה של עמודות דמה.

שאלה 2:

כאשר ניסיתי להפעיל backward elimination לקח לי זמן להבין מה לא בסדר ולמה מתקבלת שגיאה. לאחר שמצאתי פתרון טוב ב stackoverflow הוספתי אותו לקוד:

x=x.astype(np.float64)

לאחר מכן יכלתי להריץ את השורות הבאות:

x\_opt=x[:, [0,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41,42,43,44,45,46,47,48,49,50,51,52,53]]

ols=sm.OLS(endog=y,exog=x\_opt).fit()

ols.summary()

1. כדי להבין איזה משתנים רלוונטיים לחיזוי עשיתי backward elimination.
2. השיטה עובדת באופן הבא:

כדי להשתמש בשיטה אנחנו צריכים לדעת significant level והמספר הזה ניתן לנו בעבודה – 5 אחוז.

עבור כל משתנה בלתי תלוי אנחנו מחשבים p value (כמה הX הזה תורם לרגרסיה), סוג של ציון עבור כל פרמטר. לאחר שעשינו חישוב של P לכל הפרמטרים מחפשים את הX הכי גבוה וזה אומר שההססתברות שהוא תורם לחיזוי היא נמוכה מאוד ולכן יש להוציא אותו מהרשימה של העמודות בתוך x\_opt ולהמשיך לחשב לכל השאר.

הפרמטרים לפי סדר הוצאה והP שלהם:

|  |  |  |
| --- | --- | --- |
| **משתנה** | **P שלו** | **משתנה שהוצאתי ב x\_opt** |
| [X2](https://prnt.sc/uc3267) | 0.956 | 2 |
| X15 | 0.829 | 16 |
| X44 | 0.841 | 46 |
| X48 | 0.902 | 51 |
| X46 | 0.864 | 49 |
| X29 | 0.808 | 31 |
| X1 | 0.775 | 1 |
| X19 | 0.757 | 22 |
| X41 | 0.706 | 47 |
| X10 | 0.739 | 12 |
| X1 | 0.617 | 3 |
| X39 | 0.524 | 48 |
| X10 | 0.379 | 14 |
| X3 | 0.377 | 6 |
| X21 | 0.446 | 29 |
| X12 | 0.407 | 19 |
| X13 | 0.324 | 21 |
| X12 | 0.450 | 20 |
| X16 | 0.321 | 27 |
| X5 | 0.351 | 9 |
| X28 | 0.642 | 43 |
| X24 | 0.532 | 39 |
| X15 | 0.508 | 28 |
| X8 | 0.331 | 15 |
| X26 | 0.269 | 45 |
| X21 | 0.269 | 38 |
| X19 | 0.975 | 36 |
| X17 | 0.874 | 34 |
| X15 | 0.549 | 32 |
| X6 | 0.161 | 11 |
| X21 | 0.147 | 50 |
| X11 | 0.103 | 25 |
| X7 | 0.083 | 17 |
| X8 | 0.074 | 23 |

שאלה 3:

האלגוריתם ההראשון שבחרתי להפעיל הוא רגרסיה לינארית מרובת משתנים מכיוון שזה יחסית פשוט יותר ורצוי להתחיל מזה כדי לראות מה הכיוון של הנתונים. סיבה נוספת היא כי הפיצ'רים היו נראים מתאימים לטלגוריתם הזה.

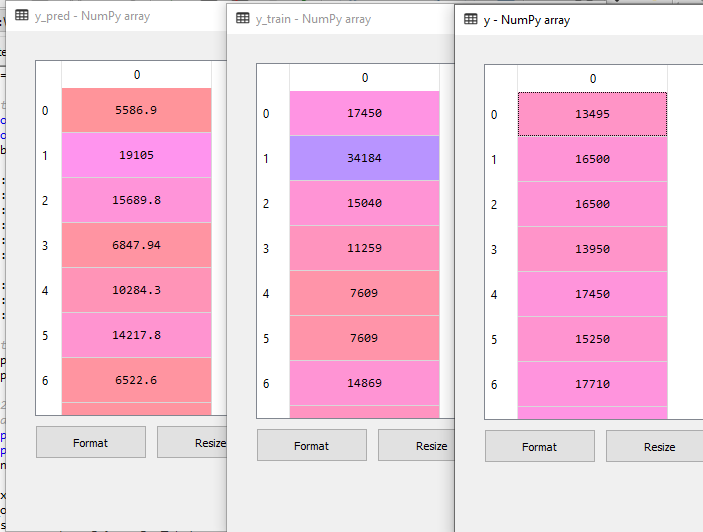
לאחר הפעלתו האלגוריתם רגרסיה לינארית פשוטה ניסיתי היה רגרסיה פולינומית וזה נראה מתאים כי אני יודעת שיש קבוצות שונות של מכוניות למשל: מיני, משפחתי וכו' ואולי כאן זה יהיה יותר מתאים.

ואז נראה מה היחס בינהם מי חוזה יותר טוב.

לינארית מרובית משתנים:

b\_0 = -19961.5428792249

תוצאות חיזוי ברגרסיה לינארית מורבת משתנים לא נראית מוצלחת:



ב y\_pred המספרים יותר נמוכים ממה שיש ב train .

אחרי הפעלת האלגוריתם של רגרסיה לינארית רציתי להפעיל cross validation כדי לראות האם הרגרסיה חוזה טוב אבל אני מקבלת שגיאה שלא הצלחתי להתגבר עליה, מצאתי ב stackoverflow שככל הנראה מדובר בבעיה שנובעת מגודל שונה. הפתרונות שהיו לא עזרו ולכן לא קיבלתי פלט לחלק הזה.

#cross validation to regression

from sklearn.model\_selection import cross\_val\_score

accuracies=cross\_val\_score(regressor,x\_train,y,cv=10,scoring='neg\_mean\_squared\_error')

print(accuracies.mean())

print(accuracies.std())

אלגוריתם רגרסיה פולינומית:

כאן רציתי להריץ רק על פיצ'רים שהיה להם ערך p נמוך בסעיף הקודם:

columns=[2, 3, 4, 5, 6, 7, 13, 14,16]

התחלתי עם עמודה FUEL מספר 2:

poly\_reg=PolynomialFeatures(degree=2)

t=pd.DataFrame(x[:,4])

x\_poly=poly\_reg.fit\_transform(t)

lin\_reg\_2=LinearRegression()

lin\_reg\_2.fit(x\_poly,y)

plt.scatter(t,y,color='red')

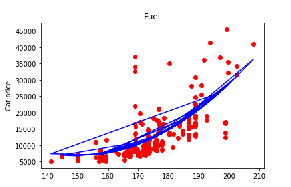
plt.plot(t,lin\_reg\_2.predict(x\_poly),color='blue')

plt.title('poly regression')

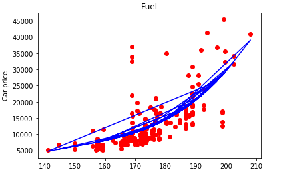
plt.title('Fuel')

plt.ylabel('Car price')

plt.show()



עדכון לחזקה 3:



\*\* גם במקרה הזה לא הצלחתי להריץ cross validation.

cross validation to poly regression for Fuel column:

accuracies=cross\_val\_score(estimator=lin\_reg\_2,x=x\_poly,y=y,cv=10,scoring='neg\_mean\_squared\_error')

print(accuracies.mean())

print(accuracies.std())

לא יצא לי לסיים את העבודה ולכן אני לא יודעת איזה אלגוריתם חוזה טוב יותר את המחיר של המכונית.