A Caveat

Assumptions of a Linear Regression:

1. Linearity
2. Homoscedasticity
3. Multivariate normality
4. Independence of errors
5. Lack of multicollinearity

And before building a linear regression model you need to check that these assumptions are true.

Dummy Variable Trap

Profit	R&D Spend	Admin	Marketing	State	New York	California
192,261.83	165,349.20	136,897.80	471,784.10	New York	1	0
191,792.06	162,597.70	151,377.59	443,898.53	California	0	1
191,050.39	153,441.51	101,145.55	407,934.54	California	0	1
182,901.99	144,372.41	118,671.85	383,199.62	New York	1	0
166,187.94	142,107.34	91,391.77	366,168.42	California	0	1

Dummy Variables

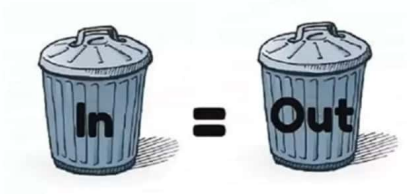
$$y = b_0 + b_1 * x_1 + b_2 * x_2 + b_3 * x_3 + b_4 * D_1 + \cancel{b_5 * D_2}$$

Always omit one

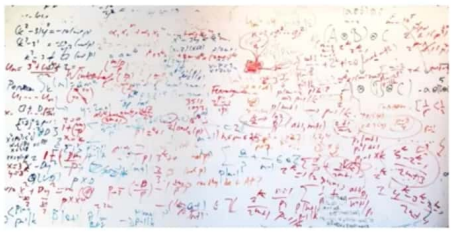
Next time we're going to cover the different ways you can build a model.

Building A Model

1)



2)



This is the process of building the model selecting the right variables.

Building A Model

5 methods of building models:

1. All-in
2. Backward Elimination
3. Forward Selection
4. Bidirectional Elimination
5. Score Comparison

And number five is score comparison.

Building A Model

Backward Elimination

STEP 1: Select a significance level to stay in the model (e.g. SL = 0.05)

STEP 2: Fit the full model with all possible predictors

STEP 3: Consider the predictor with the highest P-value. If ~~P > SL,~~ go to STEP 4, otherwise go to FIN

STEP 4: Remove the predictor

STEP 5: Fit model without this variable*

FIN: Your Model Is Ready

And your model is ready.

Building A Model

Forward Selection

STEP 1: Select a significance level to enter the model (e.g. SL = 0.05)

STEP 2: Fit all simple regression models $\mathbf{y} \sim \mathbf{x_n}$ Select the one with the lowest P-value

STEP 3: Keep this variable and fit all possible models with one extra predictor added to the one(s) you already have

STEP 4: Consider the predictor with the lowest P-value. If P < SL, go to STEP 3, otherwise go to FIN

FIN: Keep the previous model

Building A Model

Bidirectional Elimination

STEP 1: Select a significance level to enter and to stay in the model
e.g.: SLENTER = 0.05, SLSTAY = 0.05

STEP 2: Perform the next step of Forward Selection (new variables must have: P < SLENTER to enter)

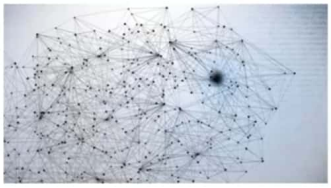
STEP 3: Perform ALL steps of Backward Elimination (old variables must have P < SLSTAY to stay)

STEP 4: No new variables can enter and no old variables can exit

FIN: Your Model Is Ready

Building A Model

All Possible Models



STEP 1: Select a criterion of goodness of fit (e.g. Akaike criterion)

STEP 2: Construct All Possible Regression Models: 2^N-1 total combinations

STEP 3: Select the one with the best criterion

FIN: Your Model Is Ready

There you go your model is ready.

Destination City - LeetCo | Code, Compile & Run | C | CodeChef User | CodeCh | Part 1 - Data Preprocessi | multiple_linear_regressio | Copy of multiple_linear_r

https://colab.research.google.com/drive/1-DDUqRf7MkaGjgMzFuH2vS1FrDN942wF#scrollTo=WemVnqge 90%

Copy of multiple_linear_regression.ipynb

File Edit View Insert Runtime Tools Help

Comment Share

+ Code + Text

Reconnect Editing

Multiple Linear Regression

Importing the libraries

```
[ ] import numpy as np
    import matplotlib.pyplot as plt
    import pandas as pd
```

Importing the dataset

```
[ ] dataset = pd.read_csv('50_Startups.csv')
    X = dataset.iloc[:, :-1].values
    y = dataset.iloc[:, -1].values
```

Encoding categorical data

```
[ ] from sklearn.compose import ColumnTransformer
    from sklearn.preprocessing import OneHotEncoder
    ct = ColumnTransformer(transformers=[('encoder', OneHotEncoder(), [3])], remainder='passthrough')
    X = np.array(ct.fit_transform(X))
```

Inspector Console Debugger Network Style Editor Performance Memory Storage Accessibility What's New

Type here to search

ENG 01:34 03-06-2020

Destination City - LeetCe | Code, Compile & Run | C | CodeChef User | CodeCh | Part 1 - Data Preprocessi | multiple_linear_regressio | Copy of multiple_linear_r

https://colab.research.google.com/drive/1-DDUqRf7MkaGjgMzFuH2vS1FrDN942wF#scrollTo=WemVnqge 90%

Copy of multiple_linear_regression.ipynb

File Edit View Insert Runtime Tools Help

Comment Share

+ Code + Text

Reconnect Editing

Encoding categorical data

```
[ ] from sklearn.compose import ColumnTransformer
    from sklearn.preprocessing import OneHotEncoder
    ct = ColumnTransformer(transformers=[('encoder', OneHotEncoder(), [3])], remainder='passthrough')
    X = np.array(ct.fit_transform(X))
```

Splitting the dataset into the Training set and Test set

```
[ ] from sklearn.model_selection import train_test_split
    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2, random_state = 0)
```

Training the Multiple Linear Regression model on the Training set

```
[ ] from sklearn.linear_model import LinearRegression#here we do not need to do something to avoid dummy trap becoz the
    #linearregression clas will mange this automatically
    #also we do not have to do backward elemeination etc things beco the class
    #(sklearn)that built the mlrm already do it automaticaly
    regressor=LinearRegression()#this builts the mlrm
    regressor.fit(X_train,y_train)#this line train the mlrm on the training set
```

```
LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None, normalize=False)
```

Inspector Console Debugger Network Style Editor Performance Memory Storage Accessibility What's New

Type here to search

ENG 01:34 03-06-2020

Destination City - LeetCo | Code, Compile & Run | C | CodeChef User | CodeCh | Part 1 - Data Preprocessi | multiple_linear_regressio | Copy of multiple_linear_r

https://colab.research.google.com/drive/1-DDUqRf7MkaGjgMzFuH2vS1FrDN942wF#scrollTo=2ORBNuLG 90%

Copy of multiple_linear_regression.ipynb

File Edit View Insert Runtime Tools Help Unsaved changes since 1:33 AM

Comment Share

+ Code + Text Reconnect Editing

```
[ ] #linearregression clas will mange this automatically
    #also we do not have to do backward elemeination etc things beco the class
    #(sklearn)that built the mlrm already do it automaticaly
    regressor=LinearRegression()#this builts the mlrm
    regressor.fit(X_train,y_train)#this line train the mlrm on the training set
```

```
LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None, normalize=False)
```

Predicting the Test set results

```
y_pred=regressor.predict(X_test)
np.set_printoptions(precision=2)
print(np.concatenate((y_pred.reshape(len(y_pred),1),y_test.reshape(len(y_test),1)),1))
```

```
[[103015.2  103282.38]
 [132582.28 144259.4 ]
 [132447.74 146121.95]
 [ 71976.1   77798.83]
 [178537.48 191050.39]
 [116161.24 105008.31]
 [ 67851.69  81229.06]
 [ 98791.73  97483.56]
 [113969.44 110352.25]
 [167921.07 166187.94]]
```

Inspector Console Debugger Network Style Editor Performance Memory Storage Accessibility What's New

Type here to search ENG 01:34 03-06-2020