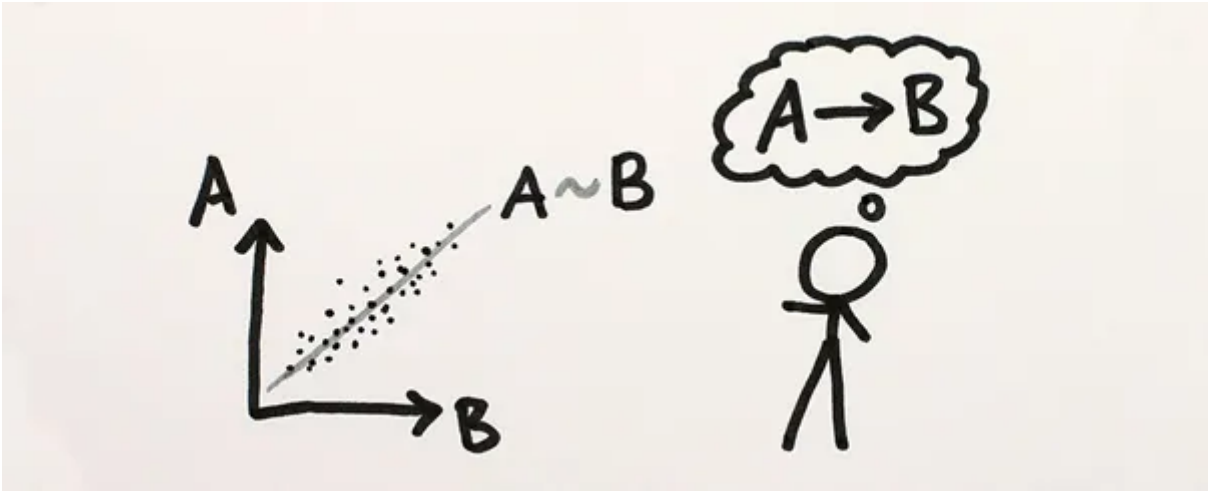
Correlation between Categorical Variables

Ritesh Jain · Follow

12 min read · May 31, 2020



Correlation measures dependency/ association between two variables. It is a very crucial step in any model building process and also one of the techniques for feature selection. While we are well aware of testing correlation between continuous variables and it very easy to understand too; testing association between categorical variables is not so common. So I decided to explore this and write an article covering following topics:

1) What is correlation

2) Chi Square test theory — How to check correlation between categorical variables

3) Chi Square test implementation in Python and understanding the output

4) Post Hoc test

5) Conclusion

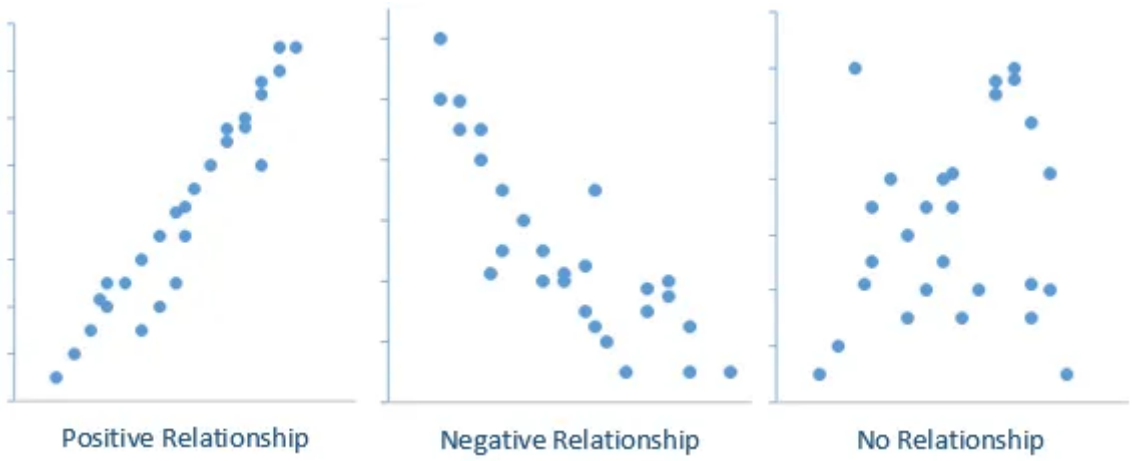
Let us get started:

1) Correlation

Let's understand correlation in general. Correlation tells relationship between two variables. While working on any predictive scorecard, we generally check correlation between two independent variables to avoid multicollinearity.

While checking correlation between continuous variables we not only get to know if variables are correlated but also the degree to which they are associated. Correlation coefficient for continuous variables vary from -1 to 1. Either of the extremes (-1 & 1) represent very strong relationship and 0 represents no relationship. In general you will not get extreme values.

Let us see how strong relationship (+ve or –ve) and no relationship looks like.



So visually one can easily make out if there is any relationship between two variables. One can get degree of association as well by plotting a contingency table

I have worked with a lot of Data Scientists and have seen people generally ignore checking correlation between categorical variables, which is also true for checking correlation between continuous and categorical. There are two major reasons why people ignore this:

a) Not understanding the importance of checking the correlation

b) Unavailability of the function that would give similar output to that of the correlation between continuous variables

Let us try to address above two problems one by one.

a) Understanding the importance of correlation between categorical variables

i. First and foremost reason would be to avoid multicollinearity. It is a crime to have high two or more highly correlated independent variables in a predictive model. They explain the same variation and also influence each other as well. You will never get to know how much variation each of the individual variable is contributing to the overall variation.

ii. Second reason is missing value treatment: Consider two categorical variables (IDVs) : X1 and X2 and they are trying to predict Y. X1 and X2 are highly correlated and so we have to pick one of them. Variable X1 is making more business sense but data is populated only for 90% of the records. So missing values of X1 can be easily imputed using X2 as they are correlated.

iii. Third reason is to understand the business better. Understanding relationship between categorical variables is not much explored, but importance once understood can do wonders to the business. Many Analytics Professionals think Analytics revolves around Predictive Models. Performing deep dive analysis to identify hidden trends and relationships between various dimensions is something that can really help business heads make big decisions. Sometimes it can also help in validating the data which can further help in improving data quality.

b) Unavailability of the function

Chi-Square test of independence is most commonly used to test association between two categorical variables. The output gives us p-value, degrees of freedom and expected values.

Code for checking correlation between TWO categorical variables is easily available. But it becomes really cumbersome and time consuming when you have more than two variables (as you have to prepare all possible combinations).

This is something which I was really struggling with, and then decided to write my own code and try to get an output that is similar to corr() output in python. Let us first understand Chi-Square test in detail.

2) Chi-Square Test

Theory: Chi-square test of independence tests the association between two categorical variables. The test is performed via contingency table or a frequency count table between the two variables. Depending on the levels that each variable has, the table's dimension can be 2X2, 3X3 etc. Let us consider an example to understand the test better:

Consider a dataset with 1,000 records and having variables — Education and Smoking. We would like to test if there is any relationship between Education Level and Smoking.

Education Categories: < Graduation, Graduation, >= Post Graduation

Smoking Categories: Yes and No

We split this process into various steps — Setting Hypothesis, Prepare Contingency Table, Getting Expected Value Count, Comparing Observed Value with Expected Value and concluding the Hypothesis

Assumptions:

i) Groups should be independent. Participants should belong to single group and not multiple.

ii) It is a Random Sample from the Population

a) We would start by setting hypothesis

Ho — There is no relationship between Education level and Smoking

Ha — There is relationship between Education level and Smoking

b) Preparing contingency table or frequency count table from the existing data

We also call it as observed values

Education	Smoking		Total
	Yes	No	
< Graduation	140	140	280
Graduation	200	340	540
>= Post Graduation	60	120	180
Total	400	600	1,000

Numbers highlighted in the red box are observed values

c) Expected Values Calculation

Now we recreate above table with expected values. This is a very important step as one should understand how we are arriving at the expected values. So let us look at the workings –

Probability of people who are '< Graduation' and 'Smoke' can be represented by:

P(< Graduation and Smoke) = P(< Graduation) * P(Smoke)

*{Applying probability concept: P(A and B) = P(A) * P(B)}*

P(< Graduation) = 280/ 1,000 = 0.28

P(Smoke) = 400/ 1,000 = 0.4

So, P(< Graduation and Smoke) = 0.28 * 0.4 = 0.112

Going back to the basics of probability for calculating Expected value,

E(X) = P(X) * n

Since two events here are people with different levels of education qualification and whether they smoke or not.

So Expected Value(< Graduation and Smoke) = 1,000 * 0.112 = 112

Similarly we calculate Expected Values for other cells,

Expected Value(< Graduation and No Smoke) = 1,000 * 0.168 = 168

Expected Value(Graduation and Smoke) = 1,000 * 0.216 = 216

Expected Value(Graduation and No Smoke) = 1,000 * 0.324 = 324

Expected Value(> Post Graduation and Smoke) = 1,000 * 0.072 = 72

Expected Value(> Post Graduation and No Smoke) = 1,000 * 0.108 = 108

[**Formula**: Expected Count = (Column Total * Row Total)/ (Table Total)]

Let us recreate the Expected Value contingency table:

Education	Smoking		Total
	Yes	No	
< Graduation	112	168	280
Graduation	216	324	540
>= Post Graduation	72	108	180
Total	400	600	1,000

d) Drawing inference from the result

Once the Expected Values contingency table is ready, Chi-Square test will compare Observed Values with Expected Values. The output of Chi-Square test will have following parameters: p-value, Chi-Square Statistics and Degrees of Freedom

With the variables under consideration we get following output (at significance level 0.05):

p-value = 0.000206

Chi — Square Statistics = 16.97

Degrees of Freedom = 2

Here, p-value < 0.05, we can reject Null Hypothesis and we can say that there is some relationship between Education level and smoking habit.

3) Chi-Square implementation in Python

In the above example we got an idea on how Chi-Square test helps us in testing the association between two categorical variables. In the world of Data Science it is equally important to understand the implementation. Since Python is most commonly used, I will show you how one can implement this easily in Python. Now it is very easy to test association between two variables, problem becomes big when you have more than two categorical variables. Consider a dataset where you have 10 categorical variables. So the number of pairs would be 45 (10C2). It would be very difficult to perform Chi-Square test 45 times and then analyse each of them. So the

best way is to have a crosstab where you can analyse all the variables in one go and arrive at some conclusion. Let us get started with this.

We would consider a Dataset from Analytics Vidhya's Hackathon. Here is the Training dataset link: ***https://datahack.analyticsvidhya.com/contest/practice-problem-loan-prediction-iii/#ProblemStatement***

Python Code:

```
## Importing required libraries
import os as os
import pandas as pd
from itertools import product
import numpy as np
import scipy.stats as ss

## Importing Dataset
df = pd.read_csv("train_ctrUa4K.csv")

## Checking column names
df.columns
```

Output:

```
Index(['Loan_ID', 'Gender', 'Married', 'Dependents',
'Education','Self_Employed', 'ApplicantIncome', 'CoapplicantIncome',
'LoanAmount','Loan_Amount_Term', 'Credit_History', 'Property_Area',
'Loan_Status'],dtype='object')
```

```
## Checking Datatype
df.dtypes
```

Output:

```
Loan_ID               object
Gender                object
Married               object
Dependents            object
Education             object
Self_Employed         object
ApplicantIncome        int64
CoapplicantIncome    float64
LoanAmount           float64
Loan_Amount_Term     float64
Credit_History       float64
Property_Area         object
Loan_Status           object
dtype: object
```

```
## Creating a DataFrame with all categorical variables
df_cat = pd.DataFrame(data = df.dtypes, columns =
                                          ['a']).reset_index()
```

```
cat_var = list(df_cat['index'].loc[df_cat['a'] == 'object'])
cat_var
```

Output:['Loan_ID', 'Gender', 'Married', 'Dependents', 'Education', 'Self_Employed', 'Property_Area', 'Loan_Status']

```
df_cat = df[cat_var]
df_cat.head()
```

Output:

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	Property_Area	Loan_Status
0	LP001002	Male	No	0	Graduate	No	Urban	Y
1	LP001003	Male	Yes	1	Graduate	No	Rural	N
2	LP001005	Male	Yes	0	Graduate	Yes	Urban	Y
3	LP001006	Male	Yes	0	Not Graduate	No	Urban	Y
4	LP001008	Male	No	0	Graduate	No	Urban	Y

```
## Removing records with at least one null value in a row
df_cat_v1 = df_cat.dropna()
df_cat_v1.shape'
```

Output: (554, 8)

```
## Let us split this list into two parts
cat_var1 = ('Gender', 'Married', 'Dependents', 'Education',
'Self_Employed', 'Property_Area')

cat_var2 = ('Gender', 'Married', 'Dependents', 'Education',
'Self_Employed', 'Property_Area')

## Let us jump to Chi-Square test
## Creating all possible combinations between the above two
variables list
cat_var_prod = list(product(cat_var1,cat_var2, repeat = 1))
```

Output:

```
[('Gender', 'Gender'),
('Gender', 'Married'),
('Gender', 'Dependents'),
('Gender', 'Education'),
('Gender', 'Self_Employed'),
('Gender', 'Property_Area'),
('Married', 'Gender'),
('Married', 'Married'),
('Married', 'Dependents'),
('Married', 'Education'),
('Married', 'Self_Employed'),
('Married', 'Property_Area'),
('Dependents', 'Gender'),
('Dependents', 'Married'),
('Dependents', 'Dependents'),
```

```
('Dependents', 'Education'),
('Dependents', 'Self_Employed'),
('Dependents', 'Property_Area'),
('Education', 'Gender'),
('Education', 'Married'),
('Education', 'Dependents'),
('Education', 'Education'),
('Education', 'Self_Employed'),
('Education', 'Property_Area'),
('Self_Employed', 'Gender'),
('Self_Employed', 'Married'),
('Self_Employed', 'Dependents'),
('Self_Employed', 'Education'),
('Self_Employed', 'Self_Employed'),
('Self_Employed', 'Property_Area'),
('Property_Area', 'Gender'),
('Property_Area', 'Married'),
('Property_Area', 'Dependents'),
('Property_Area', 'Education'),
('Property_Area', 'Self_Employed'),
('Property_Area', 'Property_Area')]
```

So we have in all 15 unique combinations. Now just imagine if you were to really check association between each of these variables, you would have to run chi-square test 15 times. So here I will help you with the code that will reduce the execution times from 15 to 1. Let us see how we can do this.

```
## Creating an empty variable and picking only the p value from the
output of Chi-Square test
result = []

for i in cat_var_prod:
if i[0] != i[1]:
result.append((i[0],i[1],list(ss.chi2_contingency(pd.crosstab(
                            df_cat_v1[i[0]], df_cat_v1[i[1]])))[1]))
result
```

Output:

```
[('Gender', 'Married', 3.195919822839657e-17),
('Gender', 'Dependents', 2.578960792620006e-05),
('Gender', 'Education', 0.2988070774266799),
('Gender', 'Self_Employed', 0.8367073961290672),
('Gender', 'Property_Area', 0.05166338663302401),
('Married', 'Gender', 3.195919822839657e-17),
('Married', 'Dependents', 4.622117038501888e-18),
('Married', 'Education', 0.9780364597922157),
('Married', 'Self_Employed', 0.9667467939322089),
('Married', 'Property_Area', 0.885151312411991),
('Dependents', 'Gender', 2.578960792620006e-05),
('Dependents', 'Married', 4.622117038501888e-18),
('Dependents', 'Education', 0.3262645381335668),
```

```
('Dependents', 'Self_Employed', 0.10758819405474623),
('Dependents', 'Property_Area', 0.25342662767163165),
('Education', 'Gender', 0.2988070774266799),
('Education', 'Married', 0.9780364597922157),
('Education', 'Dependents', 0.3262645381335668),
('Education', 'Self_Employed', 0.9115980403762441),
('Education', 'Property_Area', 0.3918935252051685),
('Self_Employed', 'Gender', 0.8367073961290672),
('Self_Employed', 'Married', 0.9667467939322089),
('Self_Employed', 'Dependents', 0.10758819405474623),
('Self_Employed', 'Education', 0.9115980403762441),
('Self_Employed', 'Property_Area', 0.7041563894088703),
('Property_Area', 'Gender', 0.05166338663302404),
('Property_Area', 'Married', 0.885151312411991),
('Property_Area', 'Dependents', 0.2534266276716316),
('Property_Area', 'Education', 0.3918935252051685),
('Property_Area', 'Self_Employed', 0.7041563894088703)]
```

Now the idea here is to create a crosstab similar to what we get from df.corr() function

```
chi_test_output = pd.DataFrame(result, columns = [‘var1’, ‘var2’,
                                                   ‘coeff’])

## Using pivot function to convert the above DataFrame into a
crosstab
chi_test_output.pivot(index=’var1', columns=’var2', values=’coeff’)
```

var2 / var1	Dependents	Education	Gender	Married	Property_Area	Self_Employed
Dependents	NaN	0.326265	2.578961e-05	4.622117e-18	0.253427	0.107588
Education	3.262645e-01	NaN	2.988071e-01	9.780365e-01	0.391894	0.911598
Gender	2.578961e-05	0.298807	NaN	3.195920e-17	0.051663	0.836707
Married	4.622117e-18	0.978036	3.195920e-17	NaN	0.885151	0.966747
Property_Area	2.534266e-01	0.391894	5.166339e-02	8.851513e-01	NaN	0.704156
Self_Employed	1.075882e-01	0.911598	8.367074e-01	9.667468e-01	0.704156	NaN

There exists a relationship between two variables if p value ≤ 0.05. So from the above table we can say that there is definitely some association between Dependents and Gender, Dependents and Married, Married and Gender. For rest of the pairs there exists no relationship

4) Post Hoc Testing

Before we try to understand Post Hoc Testing it would be good to understand p-value, Type I error and Type II error.

p-value: It is the probability of getting an extreme value when Null hypothesis is true

Type I error: Also called as False Positive, it occurs when we reject Null hypothesis when it is actually true

Type II error: Also called as False Negative, it occurs when we accept Null hypothesis when it is actually false

Consider an example where we would like to check if Education Qualification has any impact on Smoking habit. Researchers would be happy to be surprised to know if Education Qualification is actually associated with Smoking habit else it would be a futile exercise. To test this the first step is to set hypothesis:

Null Hypothesis (Ho) — Education Qualification has no impact on Smoking Habit

Alternate Hypothesis (Ha) — Education Qualification has an impact on Smoking Habit

When we perform Chi-Square test to validate this we get p-value and other statistics in the output. So α at 0.05, we check if p-value <= 0.05. If p-value ≤ 0.05 we reject Null hypothesis and say that Education Qualification does has an impact on Smoking Habit. If p-value > 0.05 we say the variation seen is by chance and we accept Null Hypothesis.

Here Type I error can occur if p-value ≤ 0.05 and we reject Null hypothesis when Null hypothesis is actually true. This can also mean that data is just trying to fool us and ***by chance*** we have got a significant result.

Let us try to understand this in detail. In the above example let us say we have three different education levels : '< Graduate', 'Graduate' and '>= Post Graduate'. In layman's term we are trying to test here does smoking habit vary for these three education qualification levels. So essentially we are comparing the smoking habits of '< Graduate' with 'Graduate', 'Graduate' with '>= Post Graduate' and '< Graduate' with 'Post Graduate'. This pairwise comparison of groups is called 'Family' and Type I error that occurs when each family is compared is called Family Wise Error (FWR). And to address this FWR we use multiple comparison method.

The probability of Null hypothesis being true when comparing '< Graduate' with 'Graduate' is 95%, so it is when we compare 'Graduate' with '>= Post Graduate' and '<

Graduate' with 'Post Graduate'. So the actual probability of accepting Null hypothesis is 0.95 x 0.95 x 0.95 which is 0.857. So α becomes (1–0.857), which is 0.14 or 14%. Initially we were considering α as 5% or 0.05 which has now increased to 14%.

So depending on the number of comparisons we can calculate the inflated α by using formula: 1 — (1- α)N , where N is the number of comparisons or tests and α is 0.05. Based on this formula below table has been constructed:

No. of comparisons	Significance level
1	0.05
2	0.098
3	0.143
4	0.185

There are various methods of addressing this issue. One of the most commonly used technique is Bonferroni corrections. This method adjusts the significance level and p-value so obtained after comparing the groups is compared with new significance level. Here is the formula of new significance level:

Adjusted Significance level (α) = 0.05/ N, where N is number of tests

In the above example we have three comparisons, so new significance level would be 0.05/3 or 0.0167. Now p-value of each group's test would be compared with 0.016. This helps in lowering the Type I error as the p-value has to be <= 0.0167 and not <= 0.05

Please note that this method is very conservative because as the number of tests or comparisons increase the adjusted Significance level (α) decreases. So one might lower the Type I error but at the same time it may lead to increase in Type II error. Type II error increases as we end up accepting Null hypothesis when actually it is false. This is because the probability of accepting null hypothesis increases as we lower the significance level.

There are many more methods that you can explore. For example Benjamini-Hochberg adjustment, Fisher's approach etc.

5) Conclusion

I strongly feel it a lot depends on the business problem at hand and the opportunity at stake. For example let us say you have to send sms and email campaigns to prospects. If the cost associated with campaign is very less then there is no harm in

NOT adjusting the significance level. Whereas in Healthcare industry it would be a good idea to work with adjusted significance level.

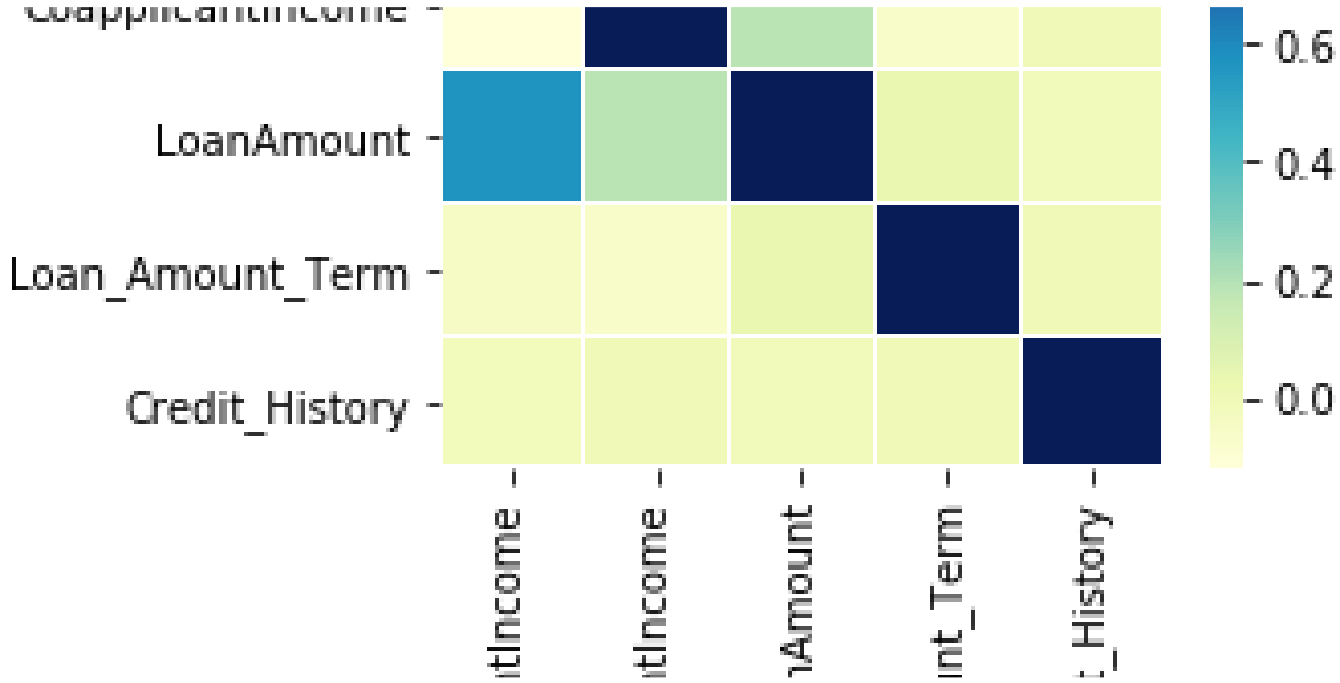
Follow

Written by Ritesh Jain

36 Followers

AI Enthusiast | Python | Machine Learning | Data Scientist | Predictive Analytics

More from Ritesh Jain



 Ritesh Jain in Analytics Vidhya

Missing Value Treatment

This article will talk about following aspects of Missing Values:

7 min read · Jan 25, 2020

1	201	Tom	35000	New York
2	301	Starc	20000	Dallas
3	401	Phil	100000	Los Angeles
4	501	Rick	42000	San Francisco
5	601	Harry	135000	Vegas

Ritesh Jain

Learning Pandas for Data Analysis

Similarity between Kung Fu Panda' Panda and Python's Pandas ?

27 min read · Apr 25, 2020

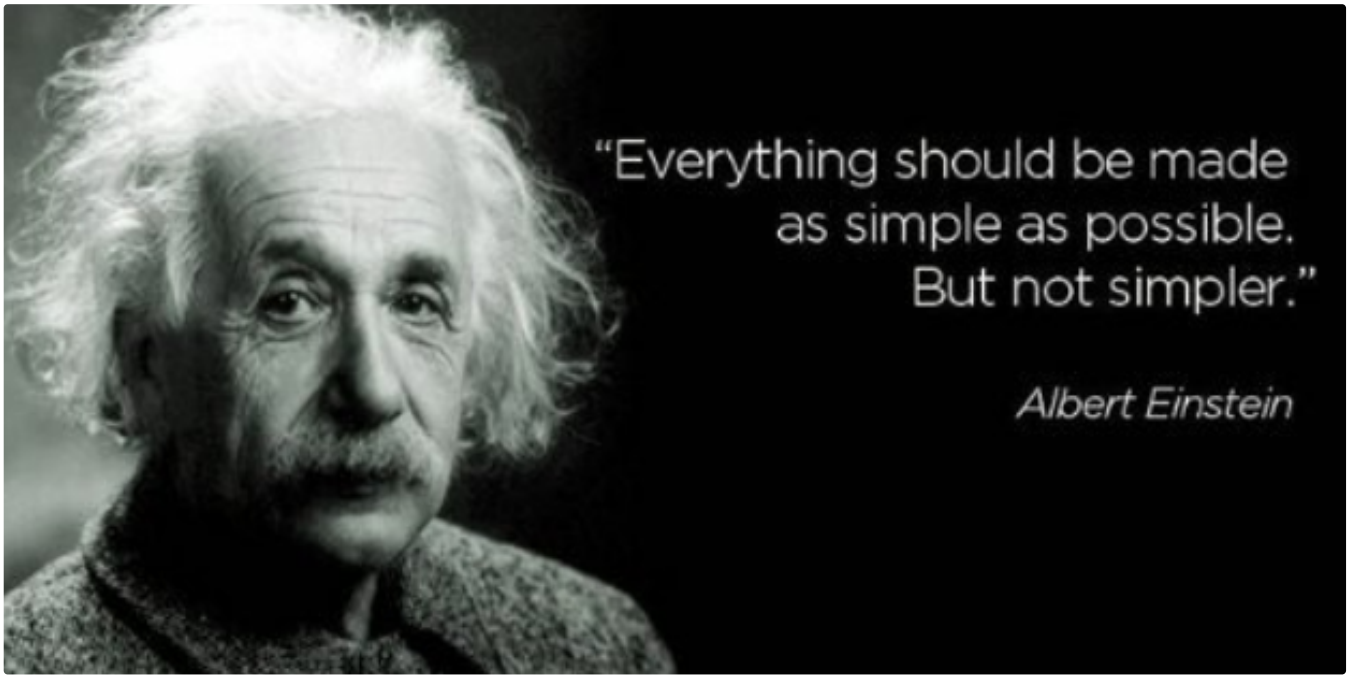
Ritesh Jain

Understanding Feature Selection and its implementation in Python

I have been asked by a lot of analytics folks on how do we select variables/features for building a predictive model. Generally, people...

· 8 min read · Jul 24, 2020

12



Ritesh Jain

Correlation with an Add-On for Easier and Faster Decision Making

You may easily find detailed explanation of this topic on many platforms like Stackoverflow, Pandas documentation or here on Medium. So...

4 min read · Aug 10, 2020

3

See all from Ritesh Jain

Recommended from Medium

...on ...rital Status	Middle School	High School	Bachelor's	Master's	Ph.D
...ried	18	36	21	9	6
...d	12	36	45	36	21
...d	6	9	9	3	3
...d	3	9	9	6	3
	39	90	84	54	33

Maninder Singh

Understanding Categorical Correlations with Chi-Square Test and Cramer's V

In this, I explained about correlation(code) between categorical features which I Learned when wanted to find the same in one of my...

9 min read · Jun 18

Aicha Bokbot in Towards Data Science

4 ways to encode categorical features with high cardinality

We explore 4 methods to encode categorical variables with high cardinality: target encoding, count encoding, feature hashing and embedding.

· 9 min read · Jun 26

Chandradip Banerjee

P value and Feature Selection

P value and Feature Selection

8 min read · Sep 21

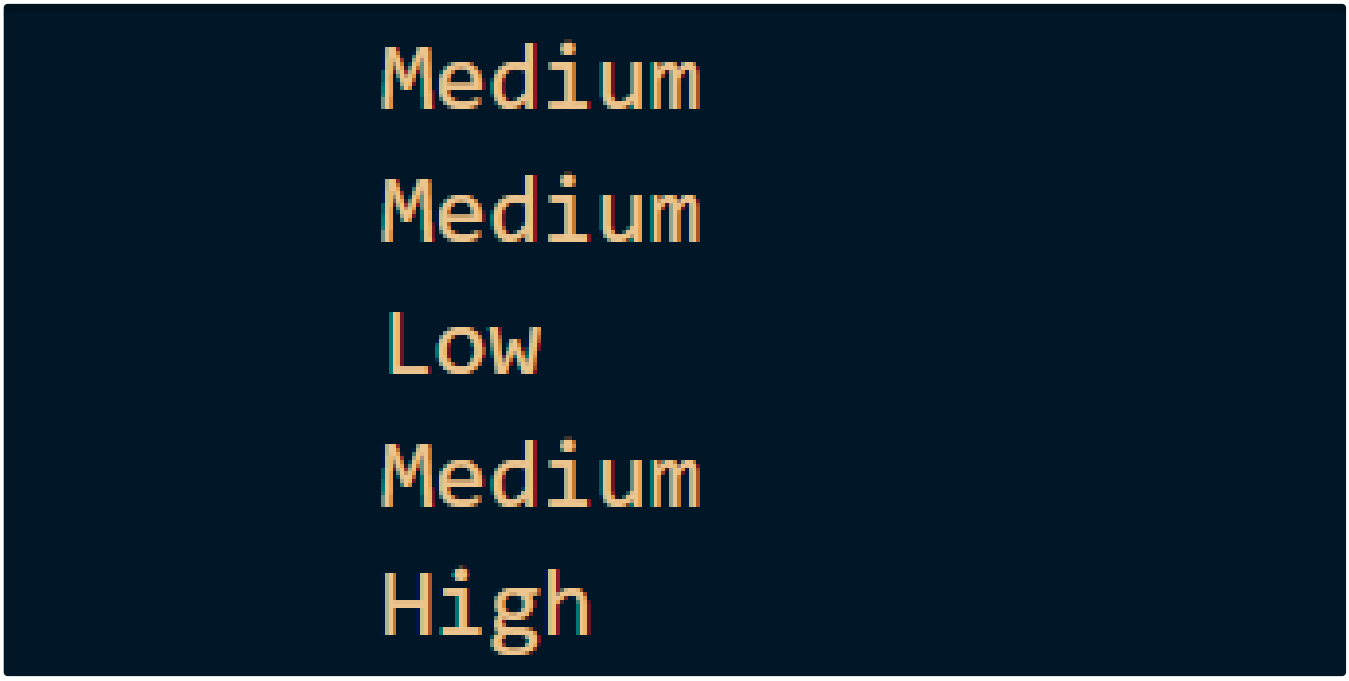
MLblog

Correlation among features and between feature | output-label, Intuition and Implementation

As a part of learning machine learning understanding about the correlation among the features helps in various aspects of data...

5 min read · Jun 6

18



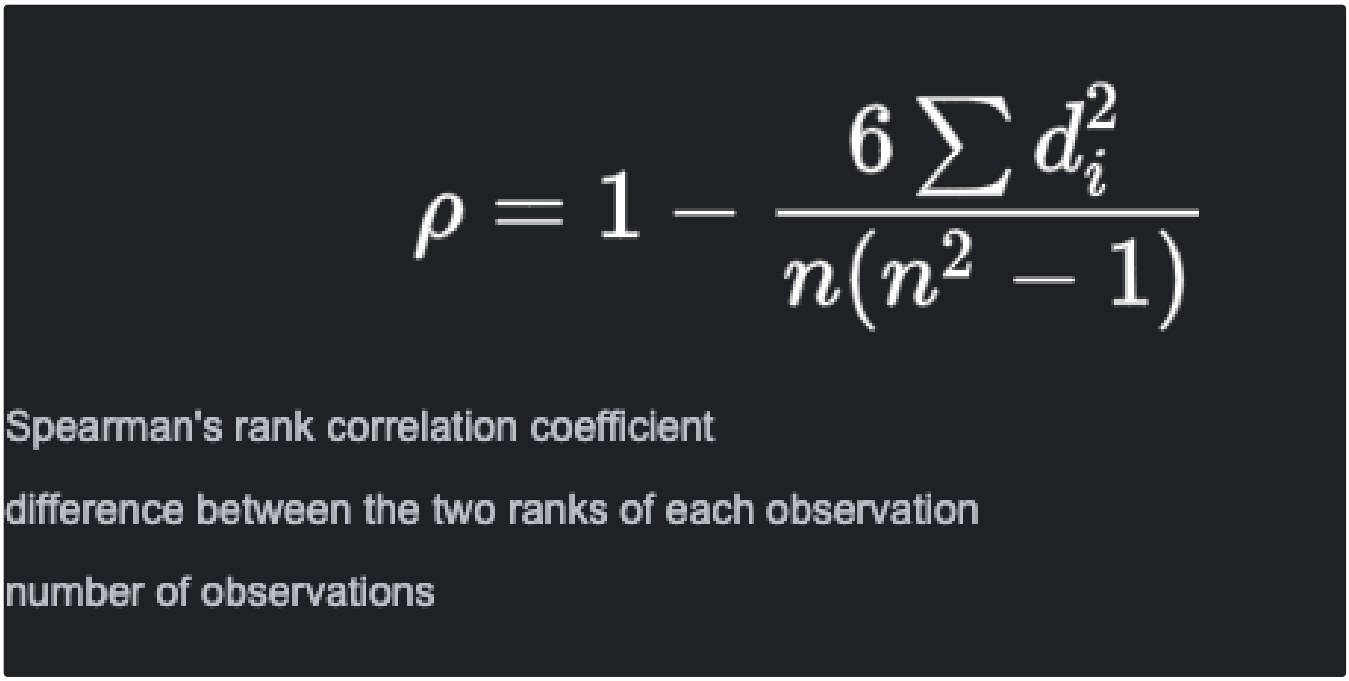
Deniz Gunay

Feature Encoding

Although some machine learning models are able to deal with categorical(non numerical) values, most of the machine learning models can only...

19 min read · Aug 18

2



Ishan | Virginia Tech & IIT Delhi | ✍️ ML/AI

Understanding Spearman Correlation

Spearman Correlation, also known as Spearman's rank correlation coefficient, is a statistical measure used to assess the strength and...

2 min read · Oct 3