One-Hot Encoding

- One-Hot Encoding is another popular technique for treating categorical variables.

- It simply creates additional features based on the number of unique values in the categorical feature.

- Every unique value in the category will be added as a feature.

- One-Hot Encoding is the process of creating dummy variables.

- In this encoding technique, each category is represented as a one-hot vector. Let's see how to implement one-hot encoding in Python:

#Importing One hot Encoder

```
from sklearn.preprocessing import OneHotEncoder
# creating one hot encoder object
onehotencoder = OneHotEncoder()

#reshape the 1-D country array to 2-D as fit_transform expects 2-D and finally fit the object


X = onehotencoder.fit_transform(df.Country.values.reshape(-1,1)).toarray()


#To add this back into the original dataframe
dfOneHot = pd.DataFrame(X, columns = ["Country_"+str(int(i)) for i in

df = pd.concat([data, dfOneHot], axis=1)

#dropping the country column
df= df.drop(['Country'], axis=1)

#printing to verify
print(df.head())
```

Output:

0	1	2	Ag
1	0	0	44
0	0	1	34
0	1	0	46

As you can see here, 3 new features are added as the country contains 3 unique values – India, Japan, and the US. In this technique, we solved the problem of ranking as each category is represented by a binary vector.

Can you see any drawbacks with this approach?

Challenges of One-Hot Encoding: Dummy Variable Trap

One-Hot Encoding results in a Dummy Variable Trap as the outcome of one variable can easily be predicted with the help of the remaining variables.

Dummy Variable Trap is a scenario in which variables are highly correlated to each other. The Dummy Variable Trap leads to the problem known as **multicollinearity**.