**Different types of feature engineering encoding techniques :**

**1)Nominal Encoding**

For ex:

Gender State

Male NewJ

Female NY

We apply nominal encoding in this kind of categorical variable

**2)Ordinal Encoding**

For ex:

Education

BE 3

b.com 4

PHD 1

Masters 2

In ordinal encoding we arrange the categorical variables with rank.

**1)Nominal Encoding**

**a)One hot encoding**

State Germany France Spain

Germany 1 0 0

France 0 1 0

Spain 0 0 1

Dummy Variable Trap

Remove spain column and when Germany and france is 0 and 0 they will specify the spain column

Disadvantage of one hot encoding:

If we have 100 unique categories and we try to convert it into dummy variable so it will increase the dimensionality.so we don’t apply one hot encoding

**b)One hot encoding with many categorical**

ex:KDD orange cup where they applied encoding to top 10 frequent categories

**c)Mean encoding**

pincodes o/p mean

560012 1 0.72

560013 0 0.6

560014 1 0.4

.

.

.

When we have pincodes and its nominal so we use mean encoding

**2)Ordinal encoding**

**a)Label encoding**

Education

BE 3

b.com 4

PHD 1

Masters 2

**b)Target guided Ordinal encoding**

mean

A 1 0.73 4

B 1 0.6 3

C 0 0.4 2

D 1 0.3 1

A 0

B 0

Based on the outputs of categories we find the mean and on the basis of mean we assign them ranks

**Handle missing values in categorical variable**

f1 f2 f3 o/p

male 23 24 yes

24 25 no

female 23 25 yes

male 26 22 yes

ways to handle missing values:-

**1) Delete the rows**

When u have missing values in a dataset then u can delete one row but it is disadvantageous coz u lost some information from dataset which is beneficial for classifying.

When u have millions of records and there are only few missing values then u can delete the rows

**2)Replace with the most frequent value**

In this technique what u can do is if male is the most frequent categorical value then u can replace the missing values with the most frequent value but it sometimes may lead to problem of imbalance dataset.

**3)Apply classifier algorithm to predict the categories**

In above dataset f1 will be the output and f2,f3,o/p will be the independent variable and we will be predicting the f1 feature .

1st,3rd,4th row is training data and 2nd row is the test data ,first we will train our classifier model and then predict it using test data and classify the categories.

**4)Unsupervised Machine learning (clustering)**

In this technique we will select f2 and f3 and on the basis of these two features we will start creating clusters and create 2 clusters if we had 3 categories then we would have divided it in 3 clusters /centroids.

So 1st row in f2 and f3 is in 1st cluster

2nd row in 2nd cluster

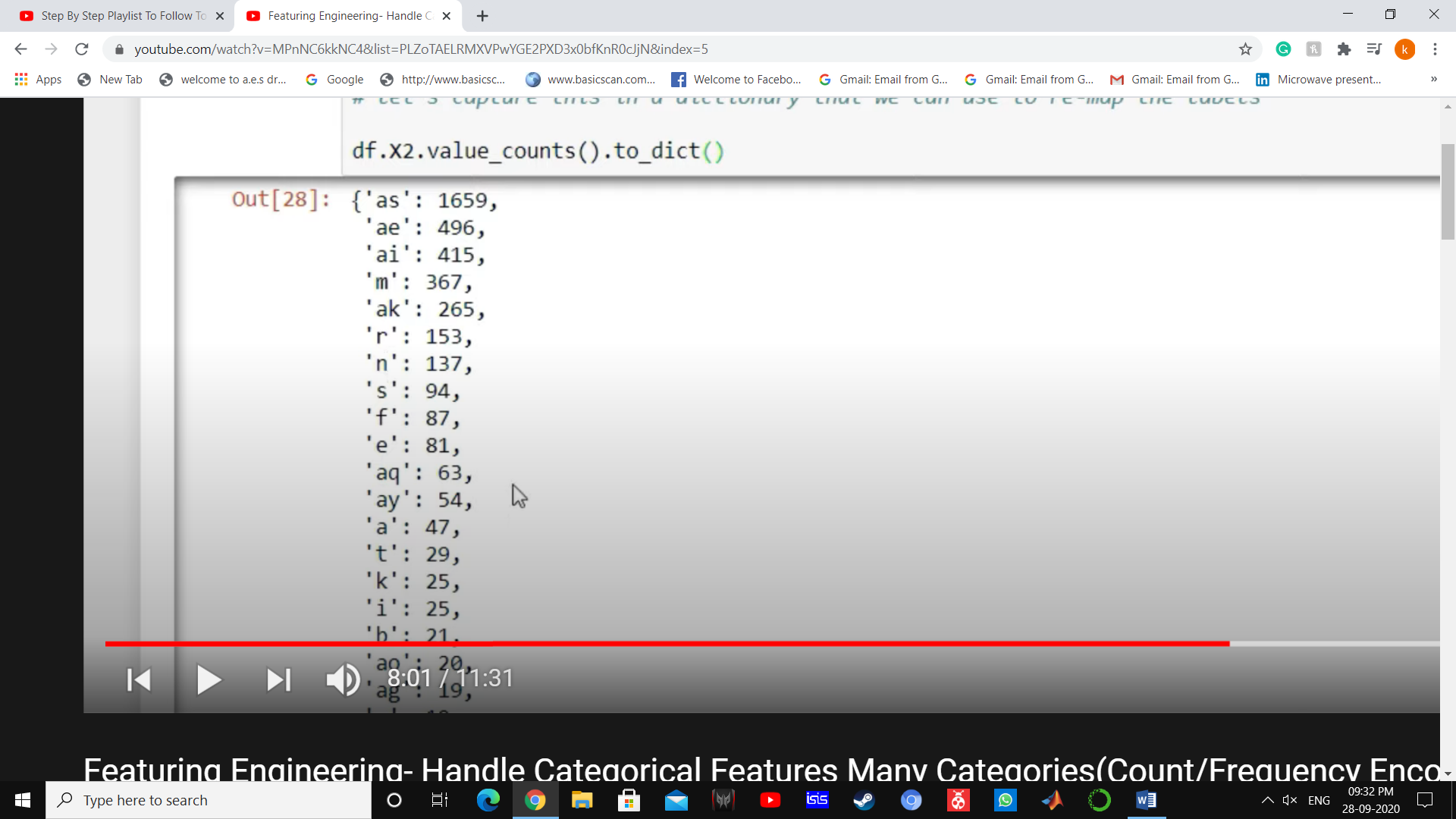
3rd row in 2nd cluster

4th row in 1st cluster

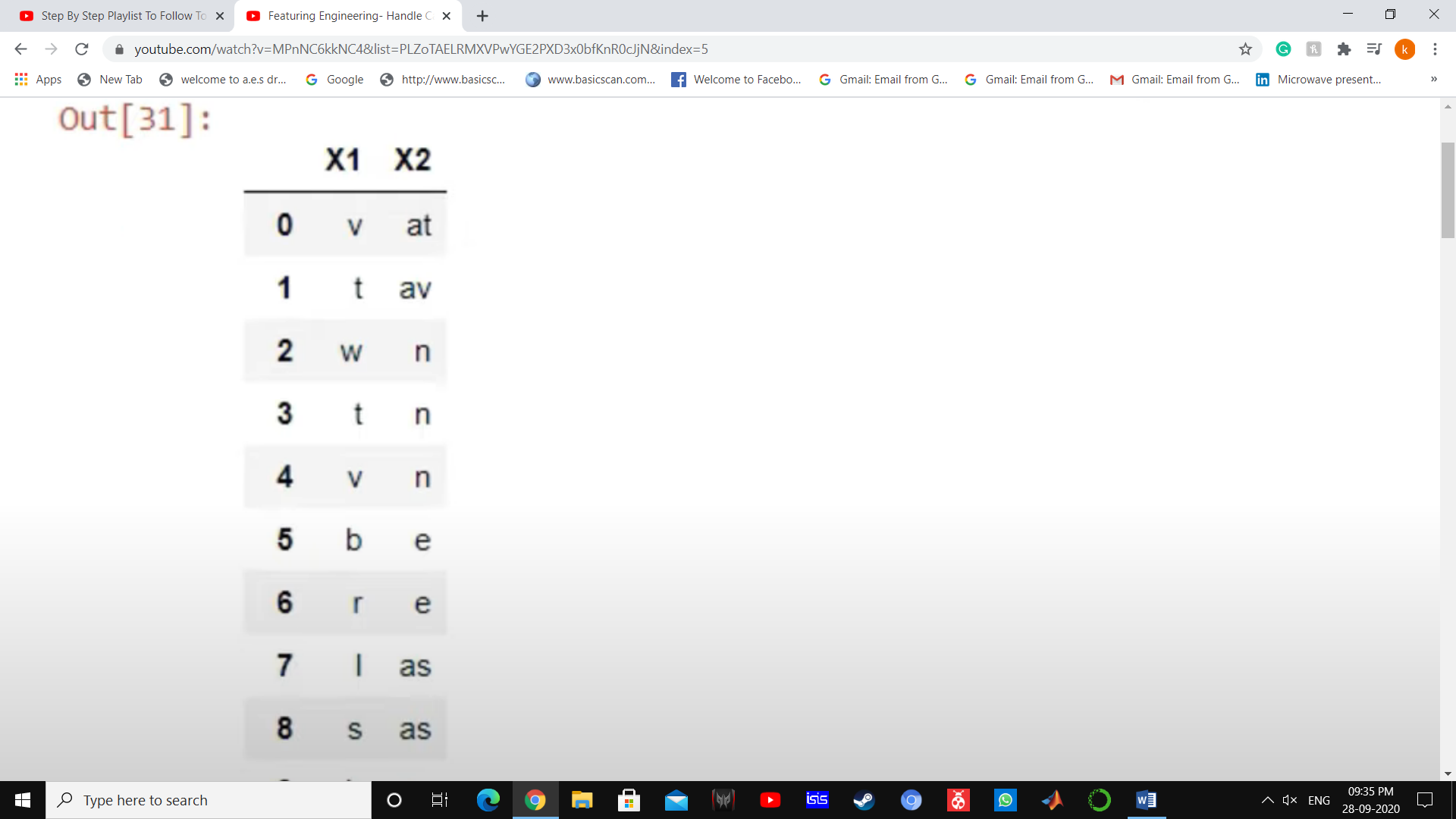
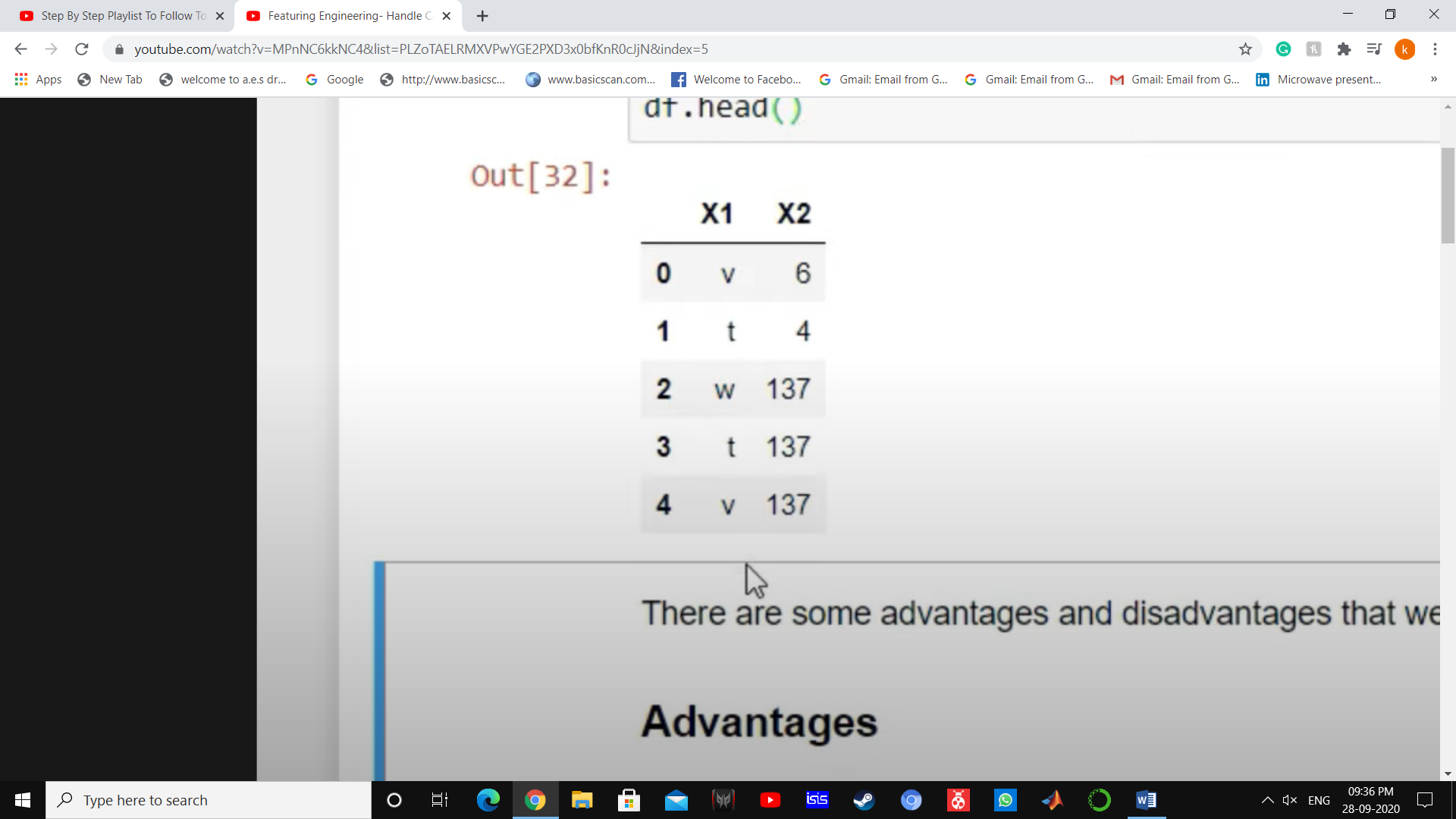
So if 2nd row is in 2nd cluster and 3rd row is in 2nd cluster and o/p is female so 2nd row also belongs to female

**Handle Categorical feature with many Categories(Count/frequency encoding)**

Approach used in frequency encoding is to replace each label of the categorical variable by the count ,this is the amount of times each label appears in the dataset .Or the frequency , this is the percentage of observations within that category .

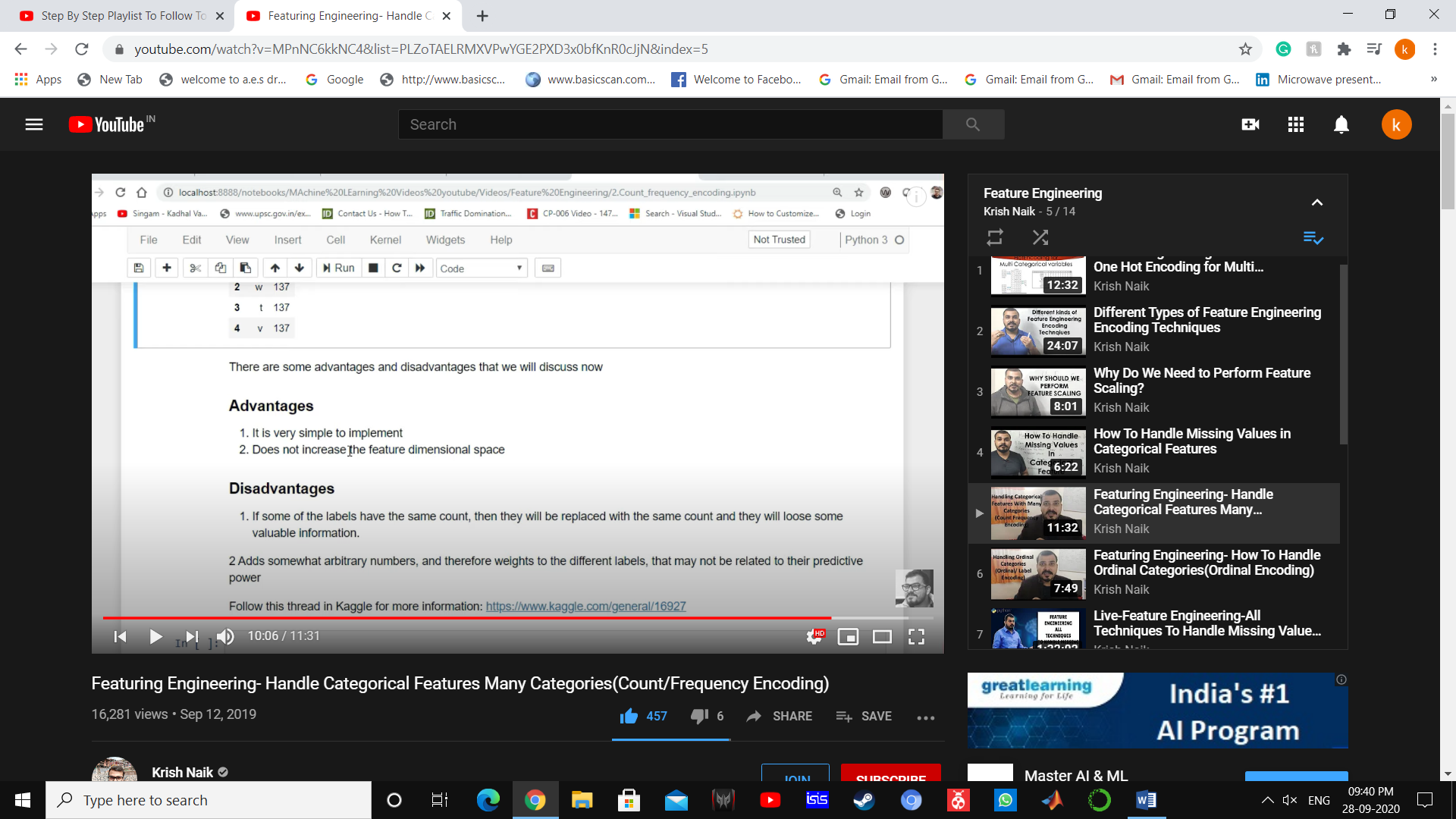


Categories:frequency

After frequency encoding

🡺



**What are the different types of Missing Data?**

Missing Completely at Random, MCAR:

A variable is missing completely at random (MCAR) if the probability of being missing is the same for all the observations. When data is MCAR, there is absolutely no relationship between the data missing and any other values, observed or missing, within the dataset. In other words, those missing data points are a random subset of the data. There is nothing systematic going on that makes some data more likely to be missing than other.

Missing Data Not At Random(MNAR): Systematic missing Values

There is absolutely some relationship between the data missing and any other values, observed or missing, within the dataset.

Missing at Random

Missing data are missing at random (MAR) when the probability of missing data on a variable is related to some other measured variable in the model, but not to the value of the variable with missing values itself. For example, only younger people have missing values for IQ. In that case the probability of missing data on IQ is related to age.

**All the techniques of handling, missing values(Continuous data)**

1. Mean/ Median/Mode replacement

2. Random Sample Imputation

3. Capturing NAN values with a new feature

4. End of Distribution imputation

5. Arbitrary imputation

6. Frequent categories imputation

**1. Mean/ MEdian /Mode imputation**

When should we apply? Mean/median imputation has the assumption that the data are missing completely at random(MCAR). We solve this by replacing the NAN with the most frequent occurance of the variables

**Advantages And Disadvantages of Mean/Median Imputation**

**Advantages**

1.Easy to implement(Robust to outliers)

2.Faster way to obtain the complete dataset

**Disadvantages**

1.Change or Distortion in the original variance

2.Impacts Correlation

**2.Random Sample Imputation**

Aim: Random sample imputation consists of taking random observation from the dataset and we use this observation to replace the nan values.

When should it be used? It assumes that the data are missing completely at random(MCAR)

**Advantages**

1.Easy To implement

2.There is less distortion in variance

**Disadvantages**

1.Every situation randomness wont work

**3.Capturing NAN values with a new feature**

It works well if the data are not missing completely at random.

**Advantages**

1.Easy to implement

2.Captures the importance of missing values

**Disadvantages**

1.Creating Additional Features(Curse of Dimensionality)

**4.End of Distribution imputation**

If there is suspicion that the missing value is not at random then capturing that information is important. In this scenario, one would want to replace missing data with values that are at the tails of the distribution of the variable.

**Advantages**

1.Easy to implement

2.Captures the importance of missingness if there is one

**Disadvantages**

1.Distorts the original distribution of the variable .

2.If missingness is not important ,it may mask the predictive power of the original variable by distorting its distribution.

3.If the number of NA is big ,it will mask true outliers in the distribution

4.If the number of Na is small the replaced Na may be considered an outlier and preprocessed in a subsequent step of feature engineering

**5.Arbitrary Value Imputation**

This technique was derived from kaggle competition It consists of replacing NAN by an arbitrary value.

**Advantages**

1.Easy to implement

2.Captures the importance of missingness if there is one

**Disadvantages**

1.Distorts the original distribution of the variable

2.If missingness is not important, it may mask the predictive power of the original variable by distorting its distribution

3.Hard to decide which value to use

**How To Handle Categorical Missing Values**

**1. Frequent Category Imputation**

**Advantages**

1.Easy To implement

2.Faster way to implement

**Disadvantages**

1. Since we are using the more frequent labels, it may use them in an over represented way, if there are many nan's

2. It distorts the relation of the most frequent label

**2. Adding a variable to capture NAN**

**3.One hot encoding**

**4.Ordinal encoding**

**5.Frequency encoding**

**Advantages**

1.Easy To Use

2.Not increasing feature space

**Disadvantages**

1.It will provide same weight if the frequencies are same

**6. Target Guided ordinal encoding**

1.Ordering the labels according to the target

2.Replace the labels by the joint probability of being 1 or 0

**7.Mean encoding**

**8.Probability Ratio Encoding**

1.Probability of survived based on cabin –Categorical Feature

2.Probability of Not survived =1- pr (survived)

3.pr(survived)/pr(Not survived)

4.Dictionary to map cabin with probability

5.Replace with the categorical feature

**Transformation of Features**

***Why Transformation of Features Are Required?***

Linear Regression---Gradient Descent ----Global Minima

Algorithms like KNN,K Means, Hierarchal Clustering--- Euclidean Distance

Every Point has some vectors and Direction

Deep Learning Techniques(Standardization, Scaling)

1.ANN--->Global Minima, Gradient

2.CNN

3.RNN

0-255 pixels

**Types of Transformation**

**1. Normalization and Standardization**

Standardization

We try to bring all the variables or features to a similar scale. Standardisation means centering the variable at zero. z=(x-x\_mean)/std

**2. Scaling to Minimum and Maximum values**

Min Max Scaling (### CNN) ---Deep Learning Techniques

Min Max Scaling scales the values between 0 to 1. X\_scaled = (X - X.min / (X.max - X.min)

**3. Scaling To Median and Quantiles**

Robust Scaler

Scale features using statistics that are robust to outliers.

It is used to scale the feature to median and quantiles Scaling using median and quantiles consists of substracting the median to all the observations, and then dividing by the interquantile difference. The interquantile difference is the difference between the 75th and 25th quantile:

IQR = 75th quantile - 25th quantile

X\_scaled = (X - X.median) / IQR

0,1,2,3,4,5,6,7,8,9,10

9-90 percentile---90% of all values in this group is less than 9 1-10 precentile---10% of all values in this group is less than 1 4-40%

**4. Gaussian Transformation**

Logarithmic Transformation

df['Age\_log']=np.log(df['Age'])

Reciprocal Transformation

df['Age\_reciprocal']=1/df.Age

Square Root Transformation

df['Age\_sqaure']=df.Age\*\*(1/2)

Exponential Transformation

df['Age\_exponential']=df.Age\*\*(1/1.2)

BoxCOx Transformation

The Box-Cox transformation is defined as:

T(Y)=(Y exp(λ)−1)/λ

where Y is the response variable and λ is the transformation parameter. λ varies from -5 to 5. In the transformation, all values of λ are considered and the optimal value for a given variable is selected.

df['Age\_Boxcox'],parameters=stat.boxcox(df['Age'])

**Outliers and impact on Machine Learning**

1.Naive Bayes Classifier : Not sensitive to outlier

2.SVM :Not sensitive

3.Linear Regression : sensitive to outliers

4.Logistic regression :Sensitive to outliers

5.Decision tree :not sensitive

6.Ensemble(RF,Xgboost,GB) :Not sensitive

7.KNN :Not sensitive

8.Kmeans :Sensitive

9.Hierarichal :Sensitive

10.PCA :Sensitive

11.Neural networks :Sensitive