Module 5

ch1 - Featurization & Feature Engineering

Video1: Introduction

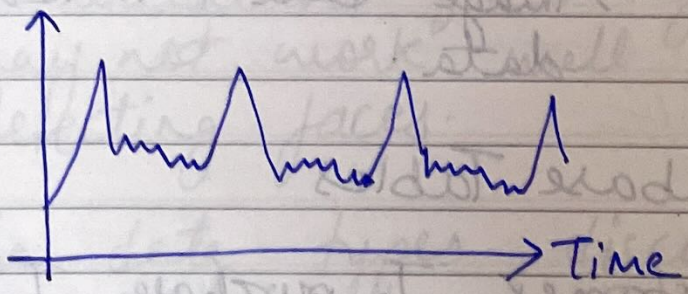
Featurization means converting any sort of data to numerical vectors.

Feature engineering means modifying the numerical vector so it works well with a model.

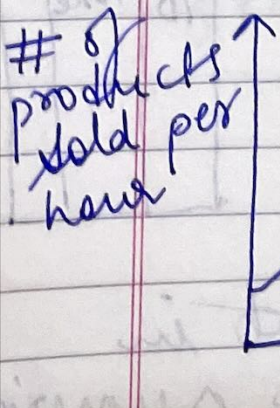
eg - BoW, TFIDF, Avg. W2V, etc are various featurizations of text data.

I Examples of time series data :-

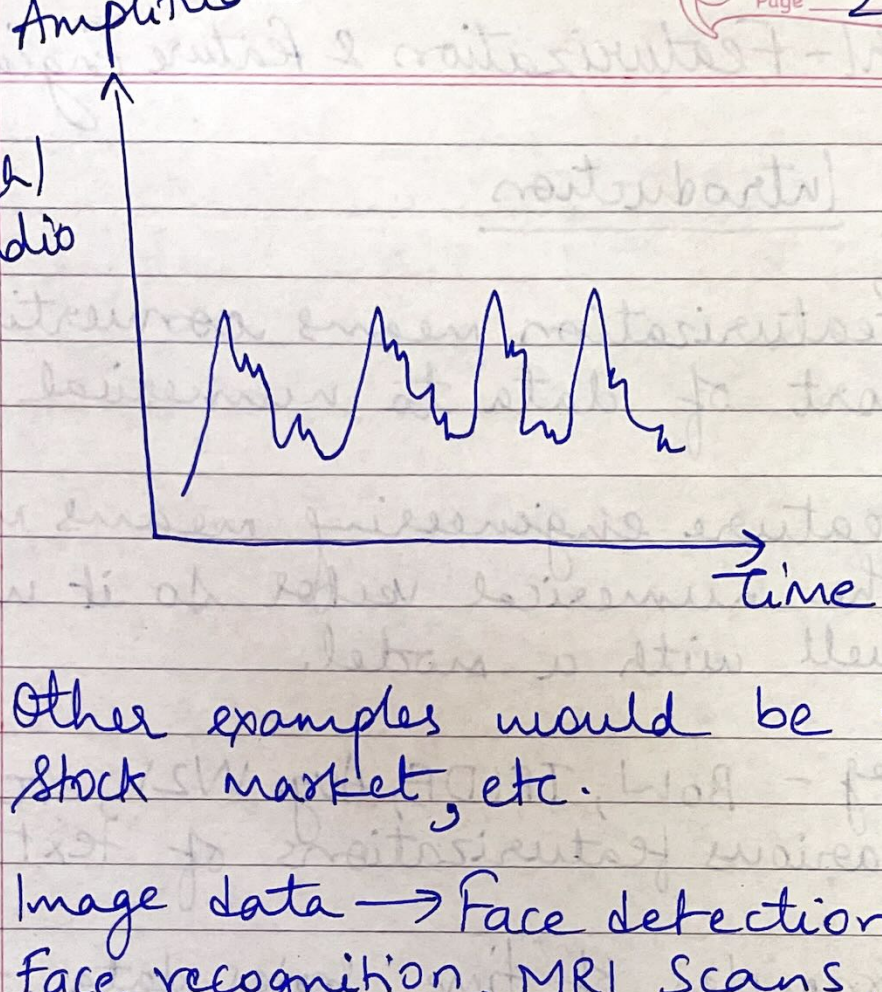
① Heart Rate



② e-commerce



③ Speech/Audio



Other examples would be ~~Stock~~ Stock market, etc.

II Image data → Face detection, Face recognition, MRI Scans, etc

Video → Image data + Time Series data.

III Database Tables

T_1 Customer Table

CustID	

T_2 purchase table

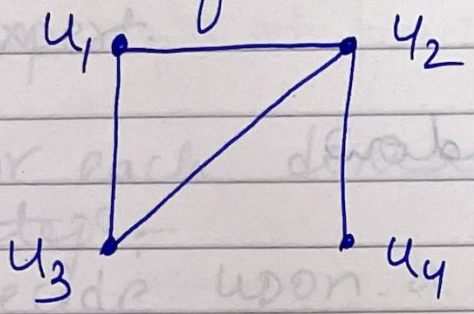
Cust ID	Pr ID	Time

T_3 Produc Table

Pr ID	Price	Di

All data in text format in database is converted to numeric features before supplying to ML mod

IV Graph data → Recommended a friend on facebook.



There are tons of types of data Researches have spent decades for getting proper featurizations.

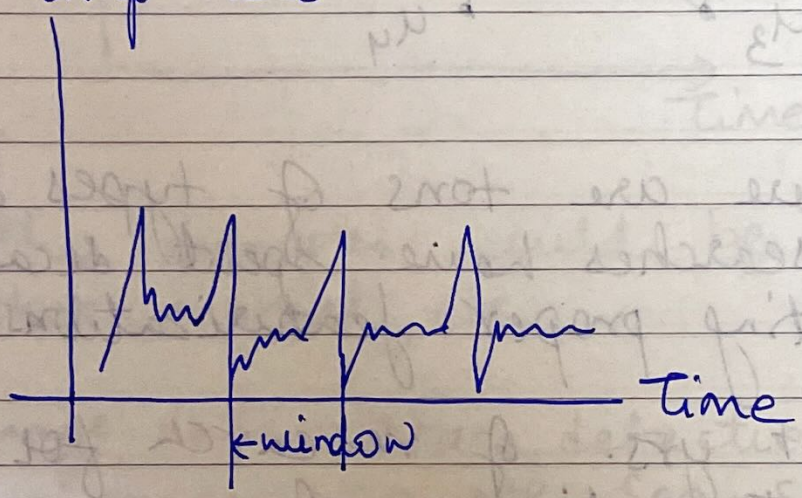
30+ yrs. of research for image featurizations only.

Featurizations for X-ray images may not work well with detecting faces.

The data types discussed here are only some of the many types, refer data and research for more ML featurizations.

Video2: Featurization of Time Series data

1) Sliding Window



Right window width should be taken care of and it depends upon domain and application specific.

We can do various types of calculations upon all the values belonging to the window like

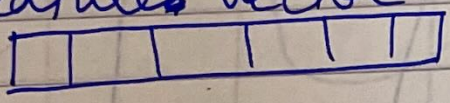
1. mean, std-dev
2. median, quantiles
3. min, max
4. max-min
5. max ÷ min
6. # local minima or maxima
7. # mean crossing & # zero crossing

which all out of these is suitable is decided by domain expert.

For each domain do :-

Steps :-

1. Decide upon a window width
2. Select features to constitute feature vector



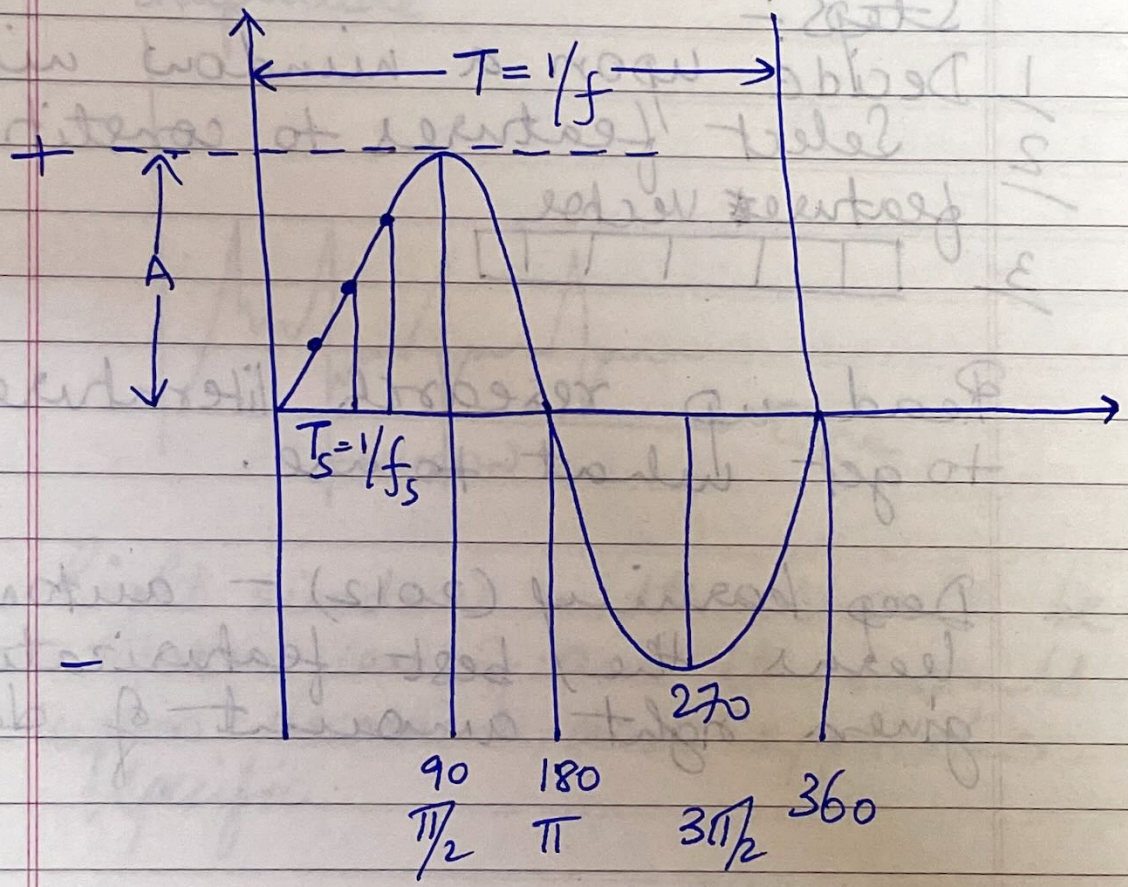
3.

Read up research literatures to get what to use.

Deep learning (2012) - automatically learns the best featurizations given right amount of data.

Video 3: Fourier Decomposition / Transform

It is a method to represent Time Series data.

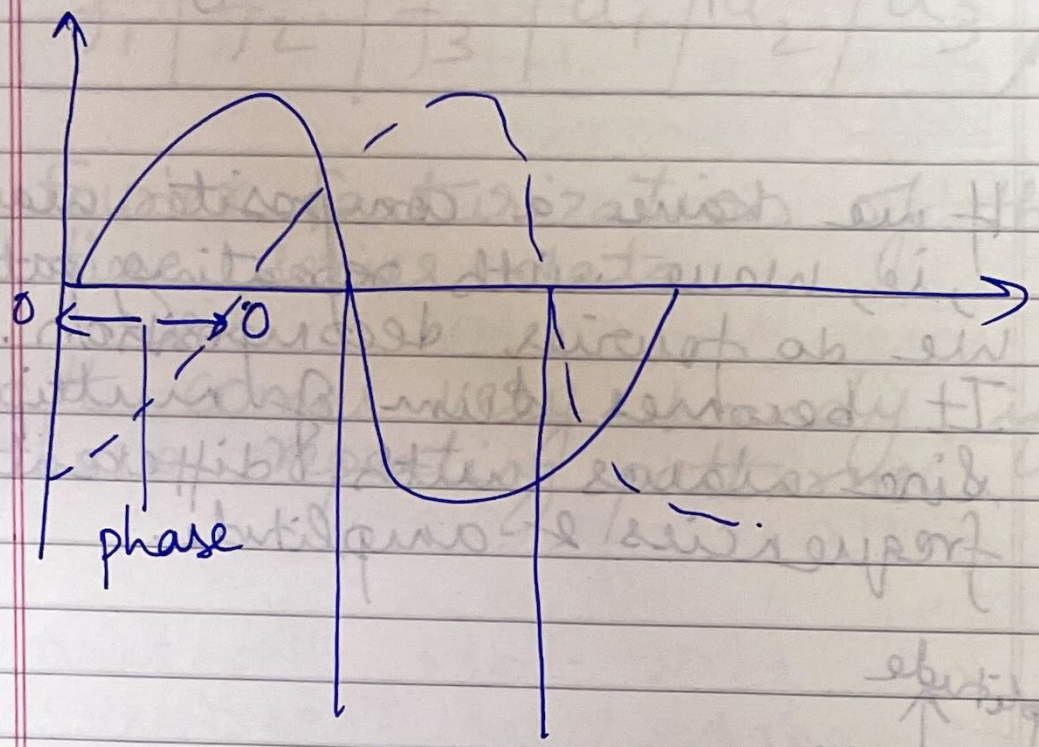


T = period
f = Frequency
A = amplitude.

$$freq = \frac{1}{T}$$

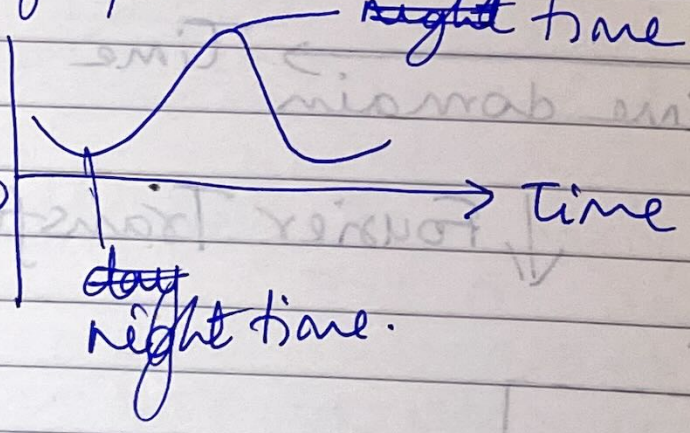
1 oscillation/sec = 1 Hz freq
2 " " = 2 Hz "

Phase of a wave -

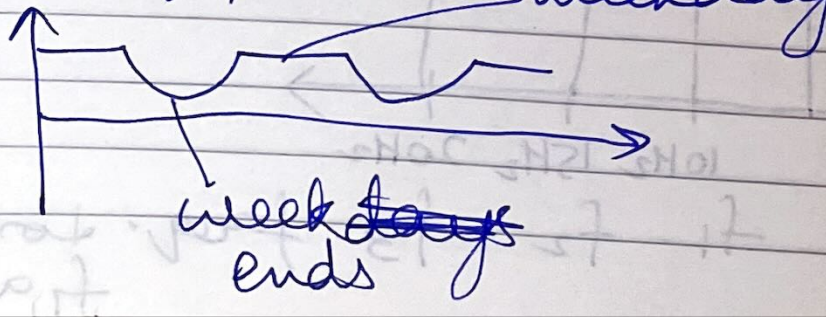


eg- e-commerce sales

daily pattern day ~~night~~ time

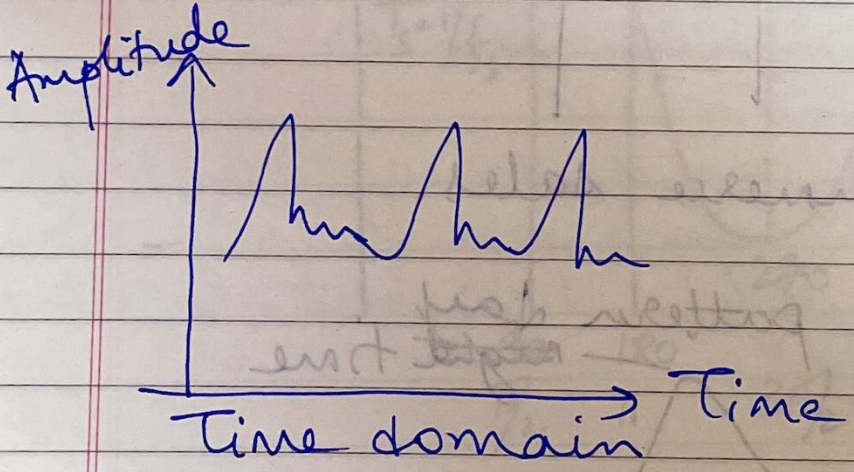


weekly pattern weekdays

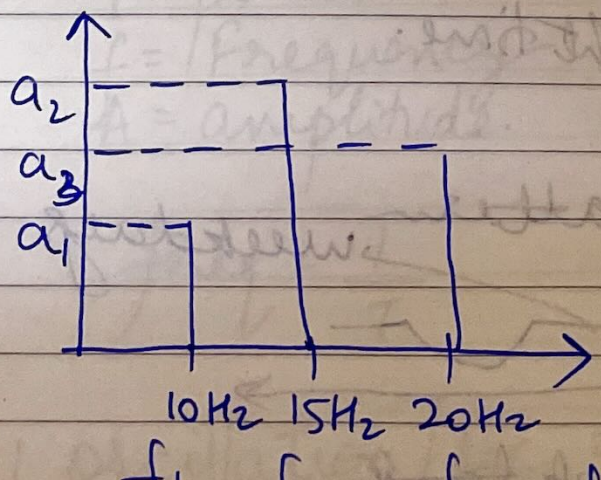


Similarly annual pattern
→ deep ~~[illegible]~~ during vacation

If we have a composite wave
, ie, wave with repeating pattern
we do fourier decomposition.
It becomes sum of multiple
sine waves with different
frequencies & amplitude.



↓ Fourier Transform



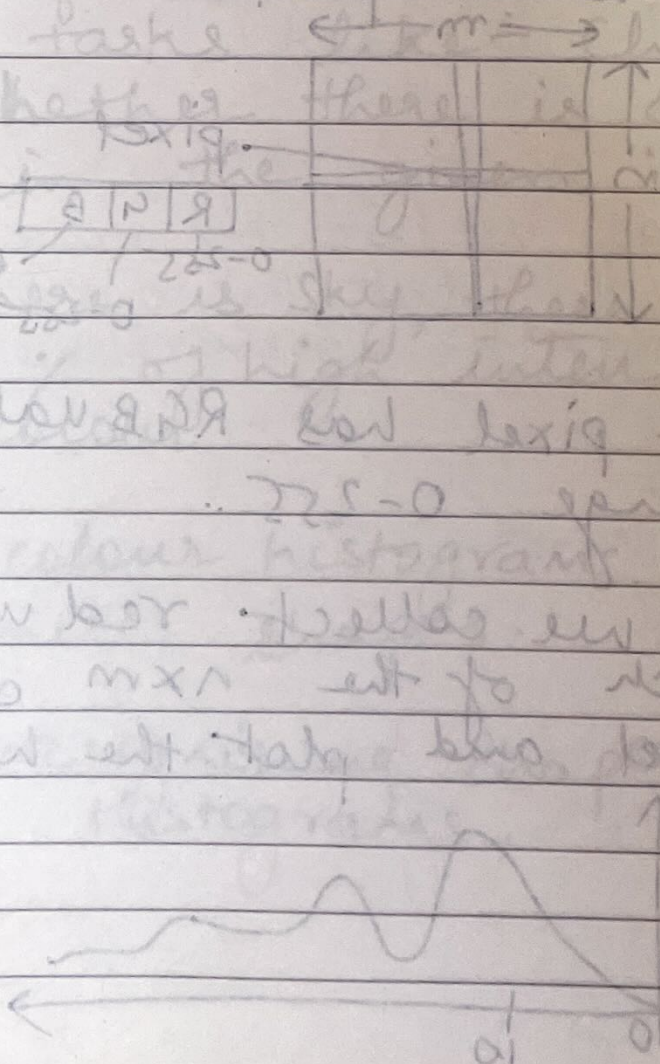
f_2, a_2
f_3, a_3

Feature vector is like :-

f_1	f_2	f_3	a_1	a_2	a_3

Fourier featurization of the time series data

It comes very handy if we have repeating patterns.



Video 5: Image Histogram

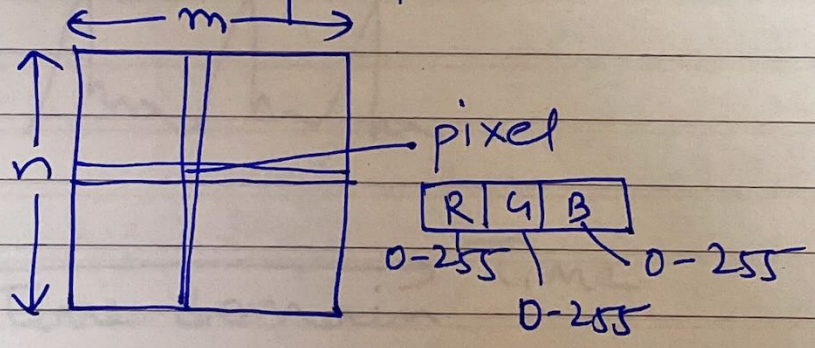
2 types of ~~Featurizations~~ Image histogra of image data:-

1. ~~Image~~ Colour histogram
2. Edge histogram

1. ~~Image~~ Colour histogram ~~(Colour~~

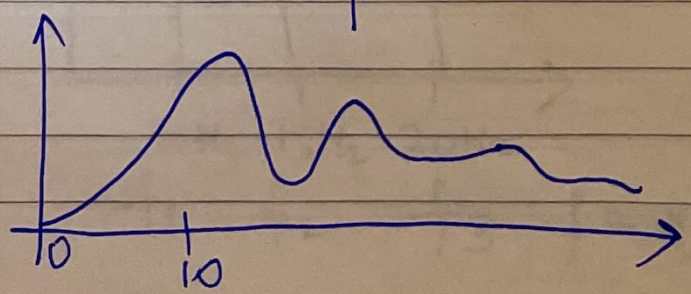
The colour histogram in the following steps:-

We first take the RGB value of each pixel like this:-



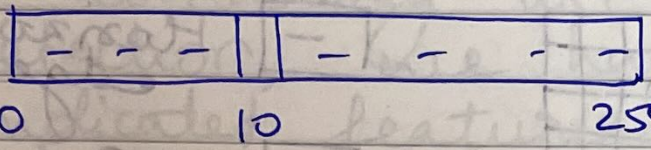
The pixel has RGB values in the range 0-255.

So we collect red values from each of the nxm datapoints pixel and plot the histogram



This gives the frequency of occurence of a particular intensity of red colour in the entire picture.

The histogram can then be converted to a vector.



Similarly we have vectors for Green & Blue colours as well.

We can concatenate them to get the resultant feature vector.

Such featurization can be used for tasks like: Check whether there is sky or not in the given image.

If there is sky, there will be high % of high intensity blue colour.

But colour histograms cannot recognize shapes.

For recognizing shapes, we have Edge Histograms.

2 Edge Histograms

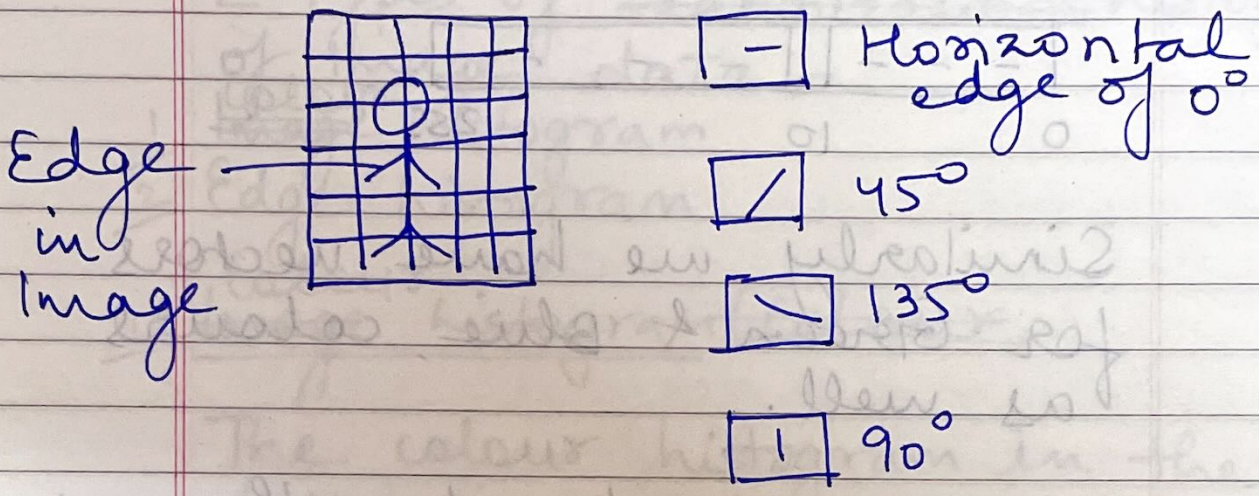
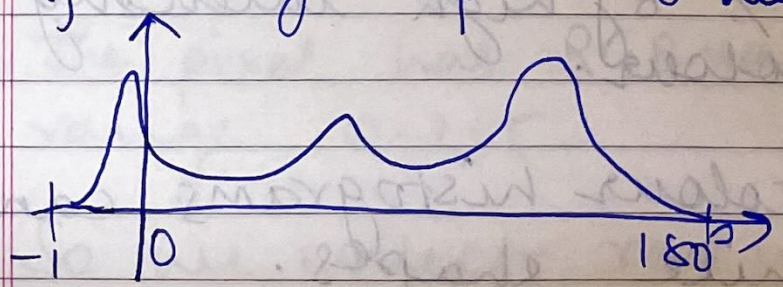
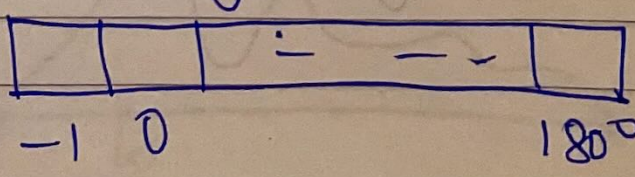


Image is broken down into grid as in above and edge angle for each pixel is assigned.

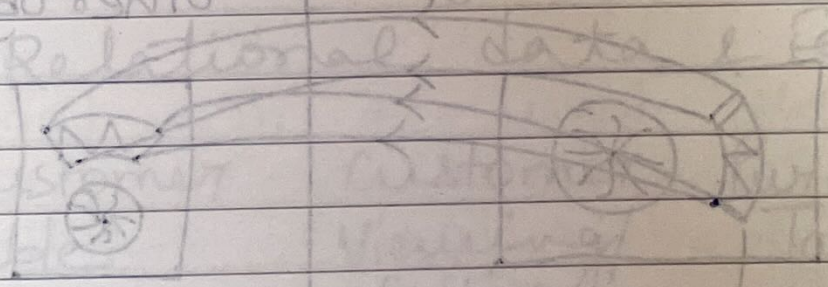
For each region we calculate edge value or edge angle and plot histogram
If no edge is present we give -1°



Histogram ~~has~~ is converted to vector of size 181



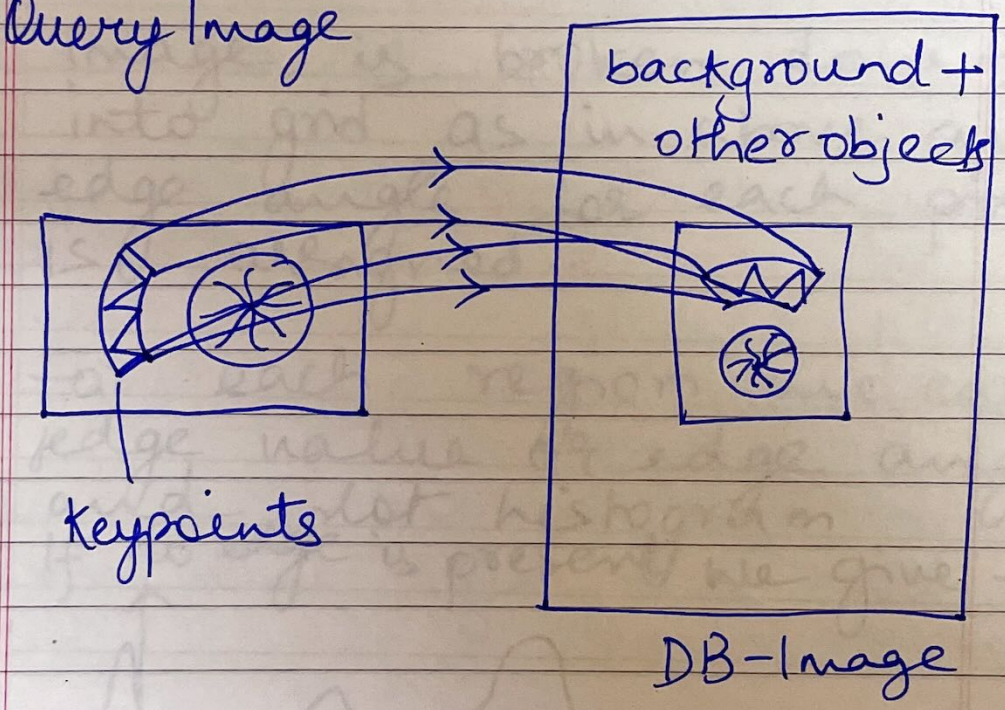
These 2 histogram featurizations are very rudimentary For something like face detection, we have more complicated featurization called "Haar" Features.



Video 6: SIFT (Scale Invariant Feature Transform)

Extremely useful for detecting objects in an image.

Variants of SIFT are used in image search eg - Amazon.



SIFT detects keypoints mostly corners. For every keypoints, it creates a 128-dimension vector. These keypoints help in detecting an object in the image.

Scale Invariance in SIFT means, even if query image is larger or ultra versa than DB image, it will still detect.

Similarly, we have Rotational Invariant.

Another library called openCV can be used to get SIFT features of a given Image.

Video 8: Relational data & Featurization

eg → Customer Table

cust ID	zip code

Customer Viewing/ Visiting

cust ID	prod ID	Time

Purchase Table

cust ID	prod ID	Time

Product data

prod ID	prod Type

This database can be featurized by extracting using pandas, python, sql & domain knowledge.

eg → Task is to predict if a customer would purchase a product in the next 7 days.

customer ID + product ID → 1/0

buy / doesn't buy

We have to use domain knowledge to convert the given (cust ID + prod ID) to a feature vector for the given task.

Some relevant features:-

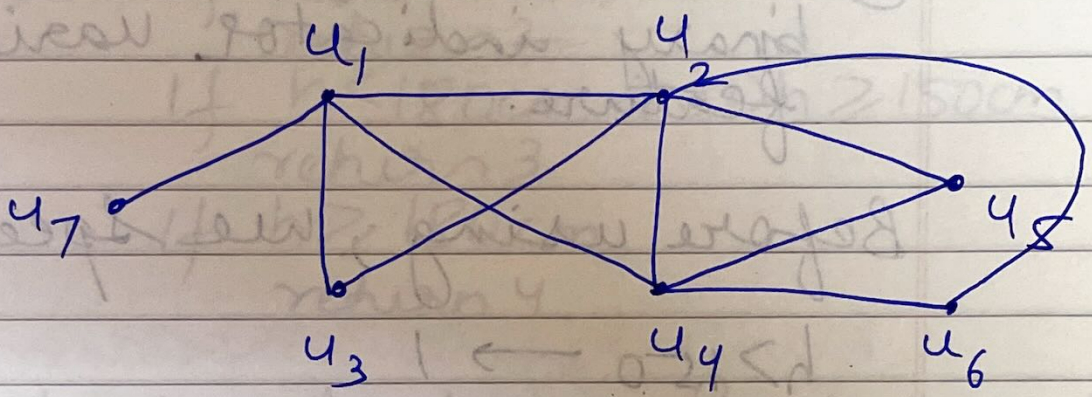
1. f_1: # of times the cust ID visited the page of given prod ID in last 24 hrs.

2. f_2: # of times the cust ID visited any product of same type as prod ID in last 24 hrs.

f_3 : Income level

f_4 : Zip codes → Signifies urban/rural areas.

Video 9: Graph data & Featurization



Facebook/Social networking websites use social graphs.

u_i : vertex (users)

(u_i, u_j) : edge (friendship)

Task : Recommend features would be

1. f_1 : # of mutual friends b/w u_4 & u_i $\forall$ $i \neq 4$
2. f_2 : # of paths b/w u_4 & u_i $\forall$ $i \neq 4$.

Video 10: Indicator Variable

eg:- Height can be used

1. Directly as a real valued feature
2. Can be converted to binary indicator variable feature.

Before using, we specify:-

$h > 150 \rightarrow 1$

$h \leq 150 \rightarrow 0$

why we may used to do so & decide threshold are problem specific

It may refer male as 1 & female or anything else.

eg 2 - Categorical feature like country :-

```
if (country = India) OR
   (country = USA) :
    return 1
else return 0
```

VII: Feature Binning (Bucketing)

This is a logical extension to indicator variable.

eg- If $h < 120cm$
return 1
If $h \leq 150cm$ AND $h \geq 120cm$
return 2
If $h < 180cm$ AND $h \geq 150cm$
return 3
If $h \geq 180cm$
return 4.

Interview Question: Predict gender given weight (w), height (h), hair length (hL) and eye colour (ec)

D_{Train}

h	w	hL	ec	g

Challenge - Perform data driven binning on feature 'height' using

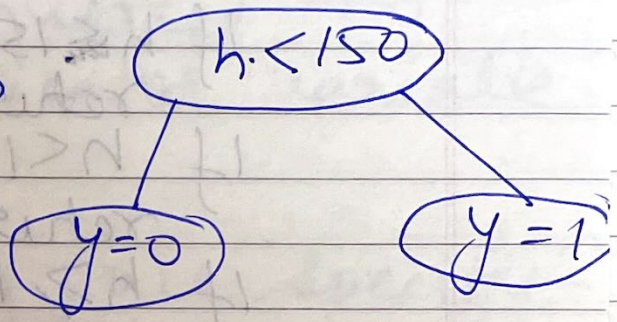
D_{train}

Soln: Take h & g & make a new table.

h	g

Simple DT →

Now train a decision tree.



Many Kaggle competitions, this trick comes handy.

V12: Interaction Variables

eg h, w, hL, ec → gender

x_i y_i

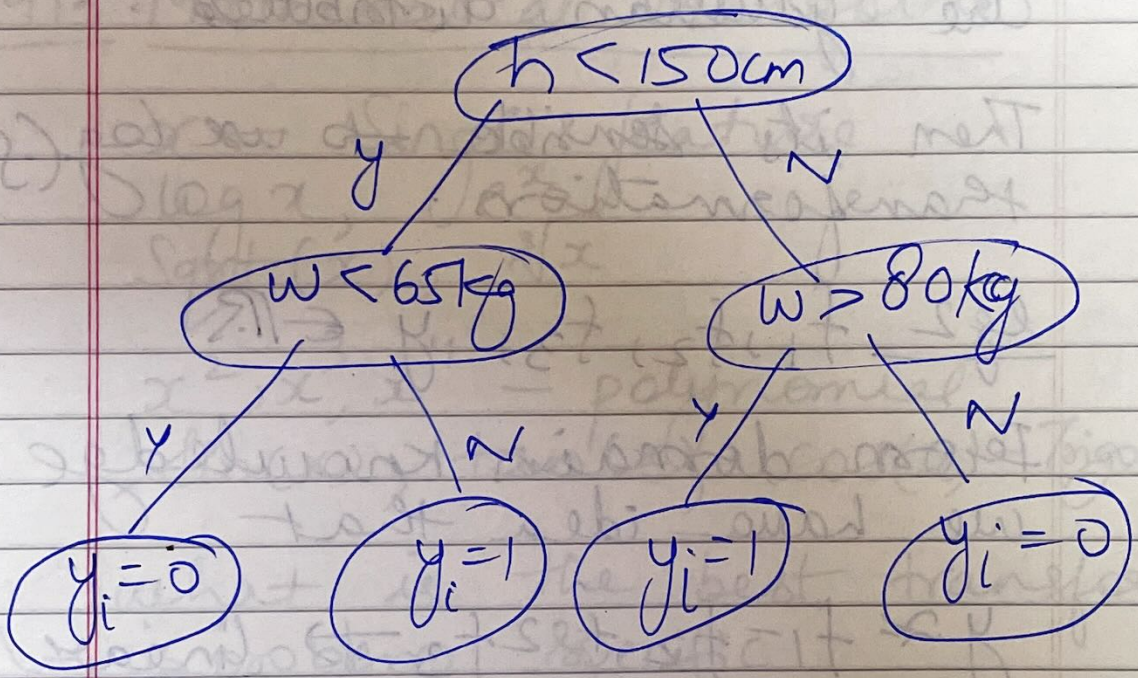
Say from domain knowledge we know if (h<150) & (w<60) female is high probability.

Then we can create a new 5th feature by 2 way logical interaction.

$f_5 = (h < 150) \text{ AND } (W < 60)$

Similarly we can also have $f_6 = h * W$. — 2 way interaction feature.

Again DT can be used to create nice interaction feature.



(h < 150cm) AND (W < 65kg)

Similarly for each leaf node we can have an interaction variable.

And these interaction features are guaranteed to be helpful, because DT was constructed to predict gender using information gain.

Model Specific featurization

VI: ~~Mathematical Transforms~~

eg 1 → f_1 ⟶ has powerlaw distribution.

We want to apply Logistic Regression same as Gaussian Naive Bayes assuming features are gaussian distributed.

Then its sensible to use $\log(f_1)$ transformation.

eg 2 $f_1, f_2, f_3, y \in \mathbb{R}$

From domain knowledge we have idea that

$y \approx f_1 - f_2 + 2f_3$ ⟶ linear combination of f_i's

Then linear models are powerful if used as compared to other.

eg Linear Regression or Linear SVM.

where $w_1 = 1 \quad w_2 = -1$

$w_3 = 2$

fits decently.

eg 3 from domain knowledge we know

$y \xrightarrow{\text{depends on}}$ Interaction of f_1 & f_2

Then better use DT / RF / GBDT

M13 Mathematical Transforms

x — Single feature

$\log x$, e^x

Sqrt(x), $\sqrt[3]{x}$

x^2, x^3, x^4 — polynomial

$\sin(x)$, $\tan(x)$, $\cos(x)$ — Trigonometric

what is the best transform — is problem specific

eg - elf's

V15: Feature Orthogonality

The more different/orthogonal (or I) the features are, the better the model would be

So whenever adding a new feature, keep in mind that the new feature should be correlated to the o/p - 'y' & very less correlated to other features.

eg $f_1, f_2, f_3 \xrightarrow{\text{Model}} \hat{y}_i$ & we also have y_i = actual labels.

How to design a new feature f_4 such that

$f_4 \xrightarrow{\text{correlated}} y_i$ &

$f_4 \xrightarrow{\text{less correlated}} f_1, f_2, f_3$

Soln $f_4 = y_i - \hat{y}_i$

Try to design f_4 correlated to eq. So f_4 related to $\hat{y}_i$ but

less correlated to other features. It is similar to boosting.

But do remember to avoid overfitting.

V16: Domain Specific featurizations

eg- Predicting heart attack using ECG data.

Always important to research and study existing featurizations by Doctors/Spl.

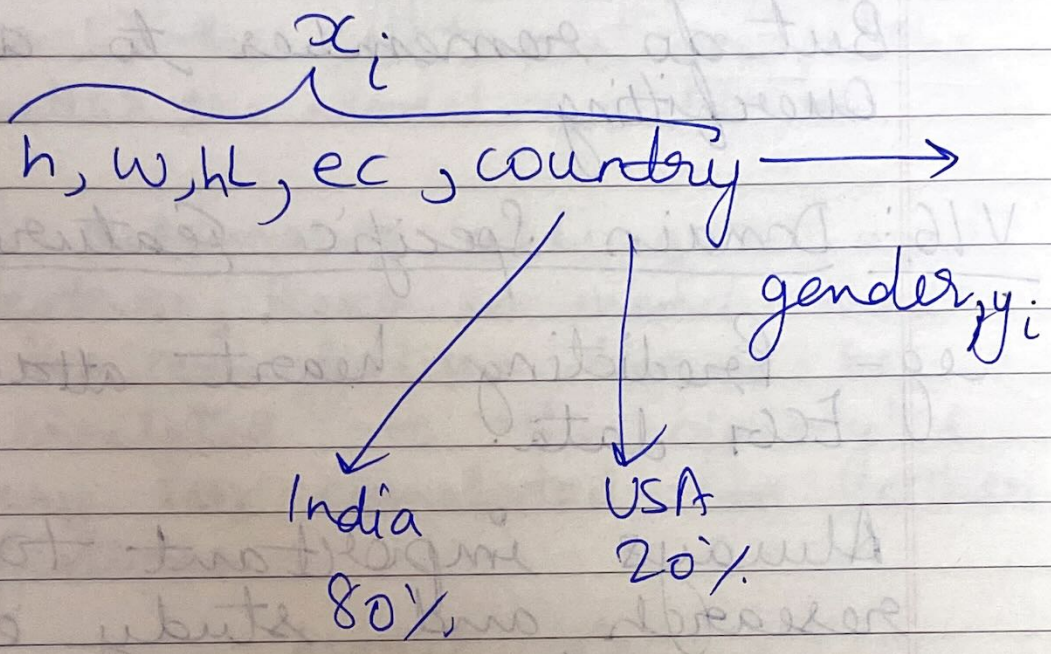
Can create new feature out of these existing using ML hacks from experience.

blog.kaggle.com/winner-solution

Has details of the feature engineering they have done.

V17: Feature Slicing

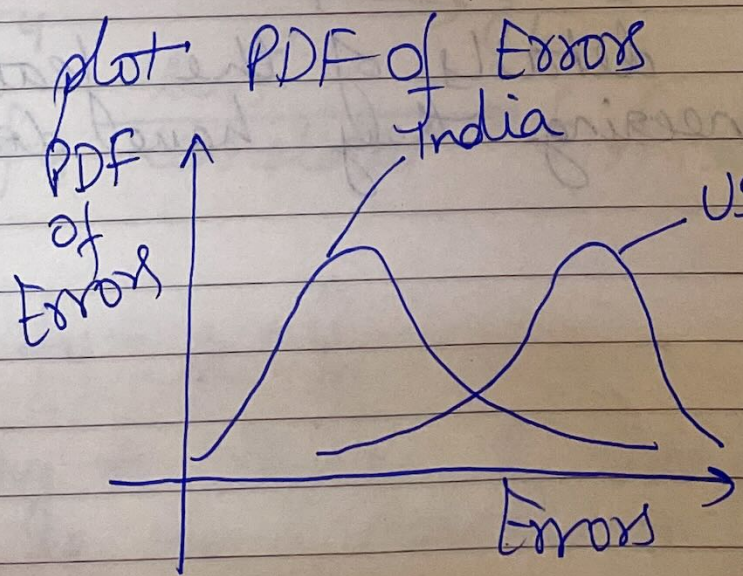
eg



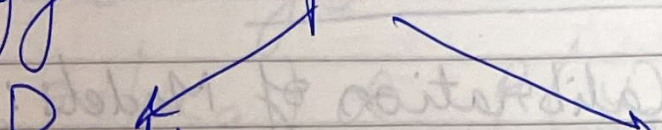
Due to majority of datapoints belonging to India, model performs

1) better on Indian test data
2) worse on USA test data

So plot PDF of Errors



Strategy → Split D to



and train M_1 and train M_2

This idea of slicing the dataset and training separate models helps if:-

1) Both category have different behaviour

2) Sufficient data points for each category.

Real World Example:- Mobile & Desktop users in Amazon have different types of traffics.

So feature slicing is done.