ML Model Validation

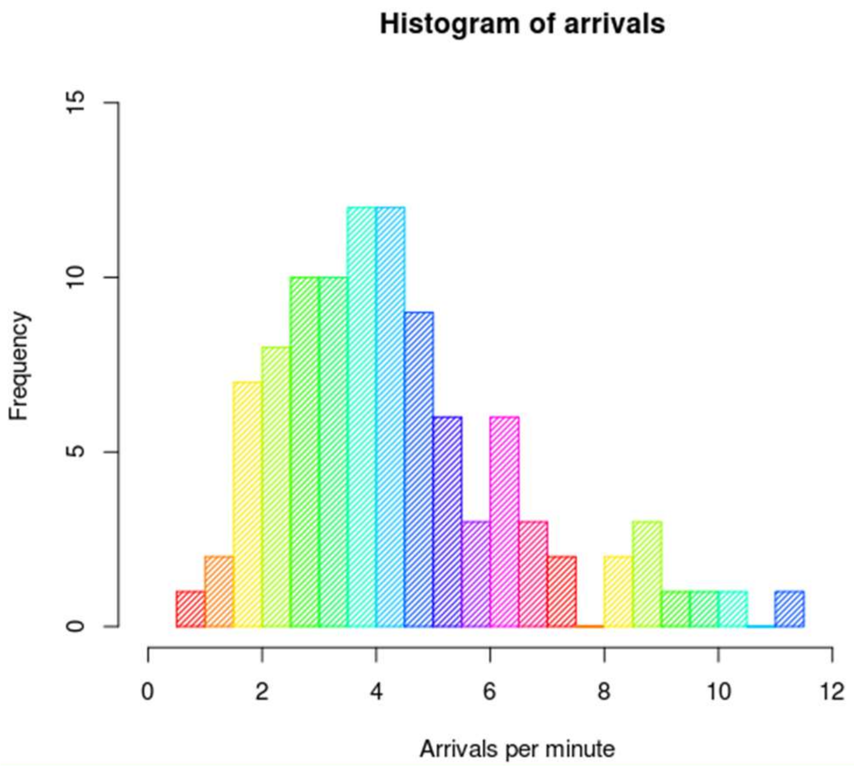
DR PRADIPTA BISWAS, PHD (CANTAB)
ASSISTANT PROFESSOR
INDIAN INSTITUTE OF SCIENCE
[HTTPS://CAMBUM.NET/](https://cambum.net/)
PRADIPTA@IISC.AC.IN

Contents

- Prerequisite
- Statistical Hypothesis Testing
  - Data Screening
  - T-test
  - ANOVA
- Linear Regression
- Classification and Clustering
  - Cross Validation
  - Cluster Validation Metrics
- Case Studies
  - Cognitive Load Estimation
  - Cursor Trajectory Prediction
  - Gaze controlled HUD for Cars

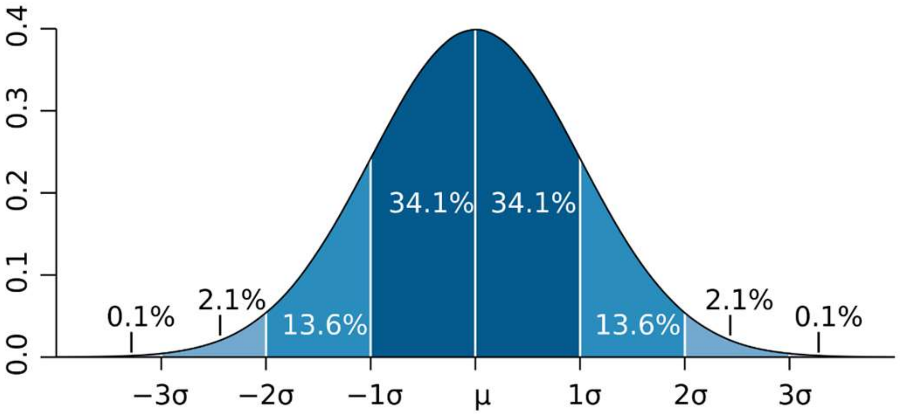
PREREQUISITE TO DATA ANALYSIS

Histogram



Histogram of arrivals
Frequency
0
5
10
15
0
2
4
6
8
10
12
Arrivals per minute

Central Limit Theorem

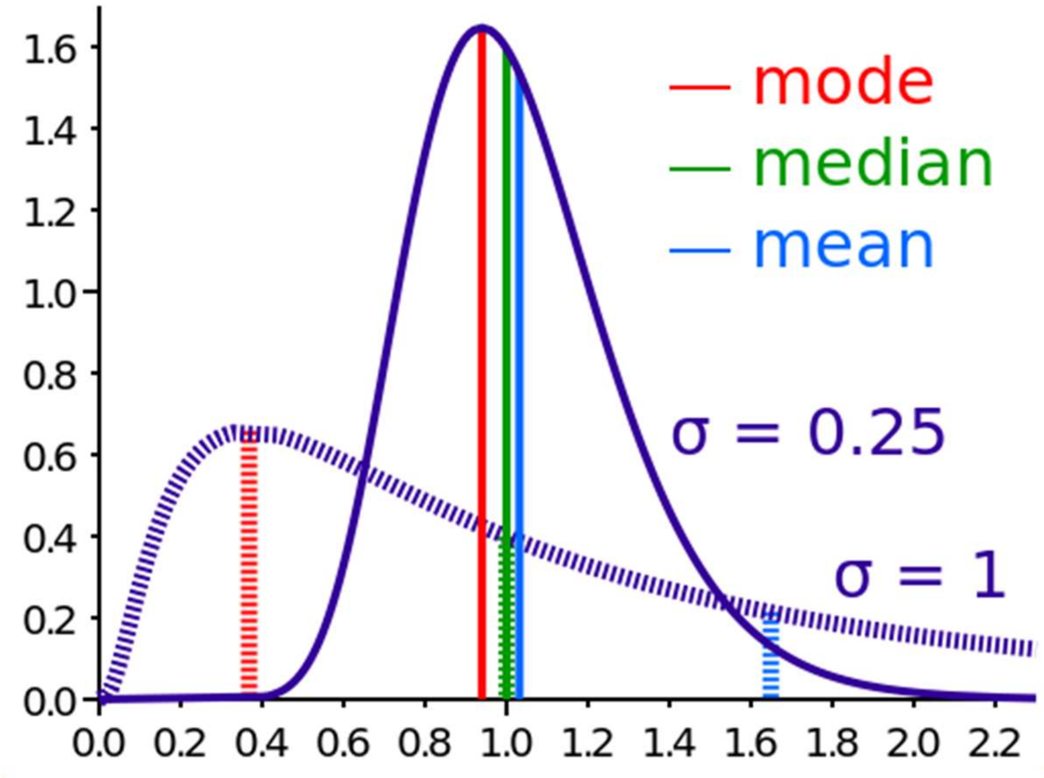


A sample is obtained containing many observations, each observation being randomly generated in a way that does not depend on the values of the other observations, and that the arithmetic mean of the observed values is computed.

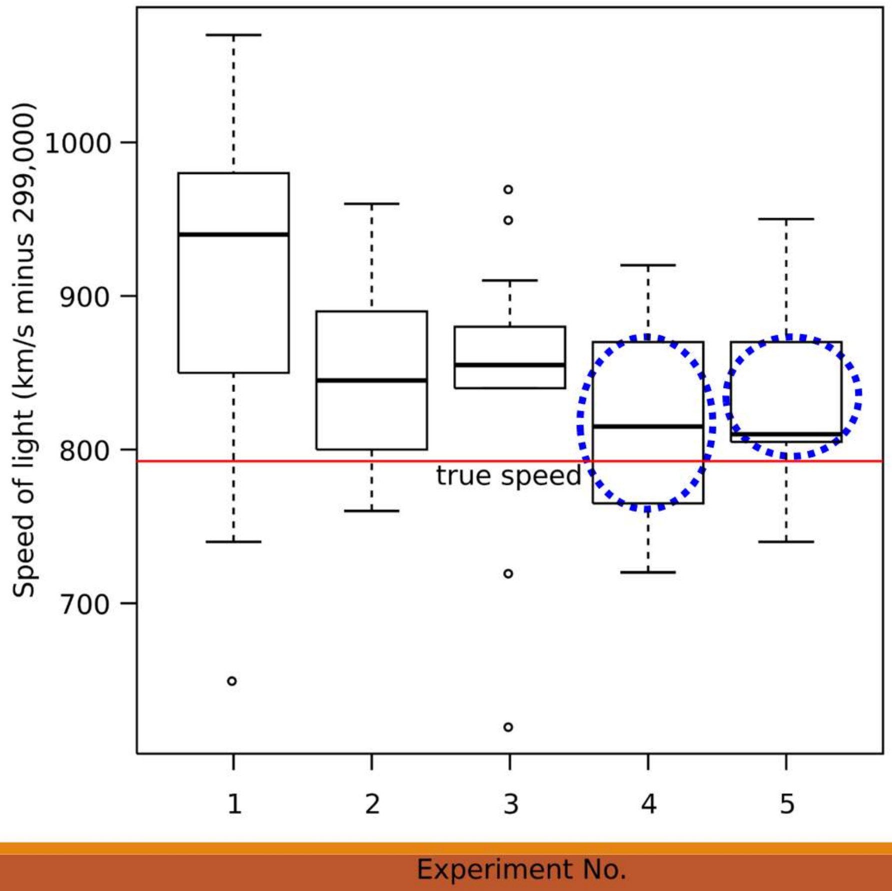
If this procedure is performed many times, the central limit theorem says that the probability distribution of the average will closely approximate a normal distribution.

If $X_1, X_2, \ldots, X_n$ are n random samples drawn from a population with overall mean μ and finite variance σ^2, and if $\bar{X}_n$ is the sample mean, then the limiting form of the distribution, $Z = \lim_{n\to\infty} \sqrt{n}\left(\frac{\bar{X}_n - \mu}{\sigma}\right)$, is a standard normal distribution.

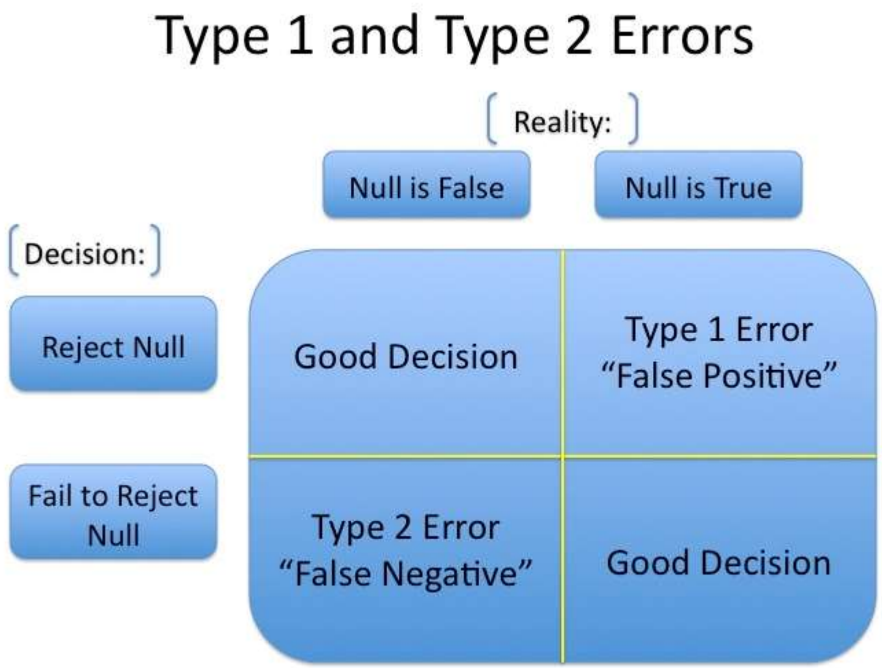
Central Tendency



Box plot


Speed of light (km/s minus 299,000)
1000
900
800
700
true speed
1
2
3
4
5
Experiment No.

Confusion Matrix



Statistical Hypothesis Testing

Data Screening

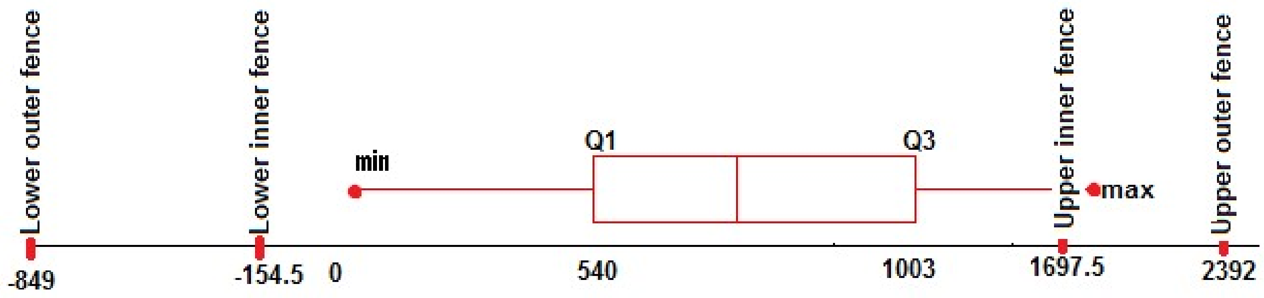
Outliers
- Inner Fence
- Outer Fence

Skewing
- In opposite direction

Unequal Variance

Missing Values

Data Transformation



Test Selection

Data normally distributed - Kolmogorov-Smirnov (K-S) test, Shapiro-Wilk test

- Parametric / Non-parametric

Relationship between two columns of data

- Correlation (Pearson's *r* / Spearman's *ρ*)

Comparing means between two columns of data

- T-test / Mann-Whitney U-Test / Wilcoxon signed rank test

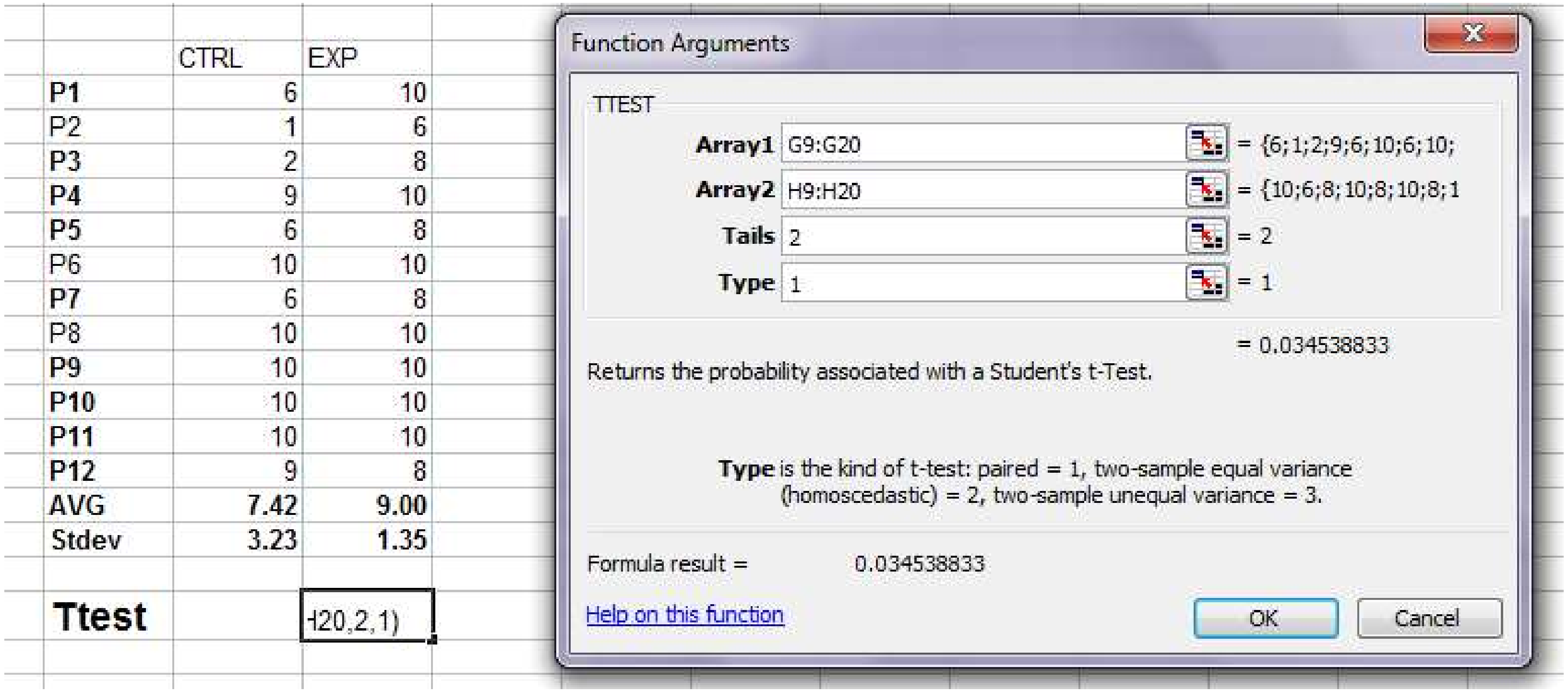
Independent **Dependent**

More than two columns

- ANOVA / Kruskal-Wallis H test /Scheirer Ray Hare Test

Categorical Data – Chi Square

Comparing means – t-test



	CTRL	EXP
P1	6	10
P2	1	6
P3	2	8
P4	9	10
P5	6	8
P6	10	10
P7	6	8
P8	10	10
P9	10	10
P10	10	10
P11	10	10
P12	9	8
AVG	7.42	9.00
Stdev	3.23	1.35
Ttest		H20,2,1)

The basic ANOVA situation

Two variables: 1 Categorical, 1 Quantitative

Main Question: Do the (means of) the quantitative variables depend on which group (given by categorical variable) the individual is in?

If categorical variable has only 2 values:

- 2-sample t-test

ANOVA allows for 3 or more groups

An example ANOVA situation

Subjects: 25 patients with blisters
Treatments: Treatment A, Treatment B, Placebo
Measurement: # of days until blisters heal

Data [and means]:

- A: 5,6,6,7,7,8,9,10 [7.25]
- B: 7,7,8,9,9,10,10,11 [8.875]
- P: 7,9,9,10,10,10,11,12,13 [10.11]

Are these differences significant?

What does ANOVA do?

At its simplest (there are extensions) ANOVA tests the following hypotheses:

H_0: The means of all the groups are equal.

H_a: Not all the means are equal

- doesn't say how or which ones differ.
- Can follow up with "multiple comparisons"

Note: we usually refer to the sub-populations as "groups" when doing ANOVA.

How ANOVA works (outline)

ANOVA measures two sources of variation in the data and compares their relative sizes

- variation BETWEEN groups
    - for each data value look at the difference between its group mean and the overall mean

$$\left(\bar{x}_i - \bar{x}\right)^2$$

- variation WITHIN groups
    - for each data value we look at the difference between that value and the mean of its group

$$\left(x_{ij} - \bar{x}_i\right)^2$$

F- statistics

The ANOVA F-statistic is a ratio of the Between Group Variation divided by the Within Group Variation:

$$F = \frac{Between}{Within} = \frac{MSG}{MSE}$$

A large F is evidence *against* H_0, since it indicates that there is more difference between groups than within groups.

ANOVA Output

```
Analysis of Variance for days
Source       DF        SS        MS        F        P
treatment     2     34.74     17.37     6.45    0.006
Error        22     59.26      2.69
Total        24     94.00
```

MS = SS /df

F = MSG /MSE

Effect size and Power

Effect size

- Percent of variance explained
- Standardized measure of magnitude of effect
- Cohen's *d*, correlation coefficient, η^2

Power

- Power of a test to detect significant effect
- (1 – Type II error)
    - Type II error (β) → probability of not detecting an effect
- Can be used to estimate sample size

Linear Regression

Linear Regression and Correlation

Explanatory and Response Variables are Numeric

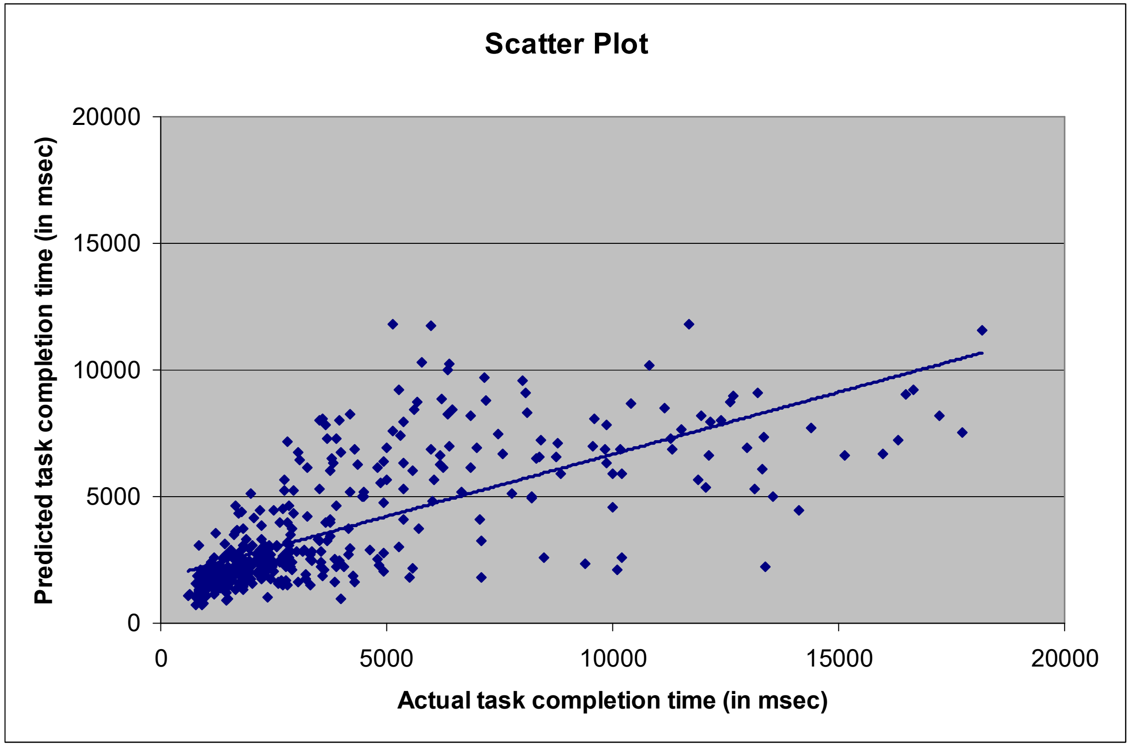
Relationship between the mean of the response variable and the level of the explanatory variable assumed to be approximately linear (straight line)

Model:

$$Y = \beta_0 + \beta_1 x + \varepsilon \qquad \varepsilon \sim N(0,\sigma)$$

- $\beta_1 > 0 \Rightarrow$ Positive Association
- $\beta_1 < 0 \Rightarrow$ Negative Association
- $\beta_1 = 0 \Rightarrow$ No Association

Scatter Plot



Validation

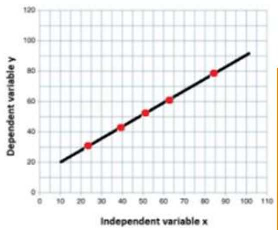
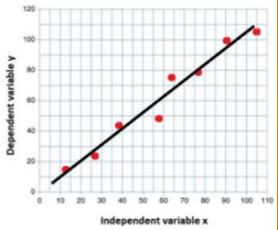
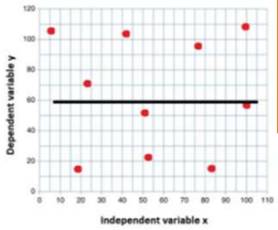
SUMMARY OUTPUT

Regression Statistics	
Multiple R	0.99295403
R Square	0.985957705
Adjusted R Square	0.98127694
Standard Error	10.90901764
Observations	9

ANOVA

	df	*SS*	*MS*	*F*	*Significance F*
Regression	2	50135.19842	25067.59921	210.640295	2.76894E-06
Residual	6	714.0399953	119.0066659		
Total	8	50849.23841			

	Coefficients	*Standard Error*	*t Stat*	*P-value*	*Lower 95%*	*Upper 95%*	*Lower 95.0%*	*Upper 95.0%*
Intercept	145.8116608	9.633328714	15.13616581	5.24478E-06	122.2397546	169.3835669	122.2397546	169.3835669
ScreenX	-0.200503294	0.00977297	-20.5161072	8.72158E-07	-0.22441689	-0.176589699	-0.22441689	-0.176589699
ScreenY	-0.000257379	0.017796814	-0.014462065	0.988930238	-0.043804614	0.043289856	-0.043804614	0.043289856

R^2 Values	Interpretation	Graph
$R^2 = 1$	All the variation in the y values is accounted for by the x values	
$R^2 = 0.83$	83% of the variation in the y values is accounted for by the x values	
$R^2 = 0$	None of the variation in the y values is accounted for by the x values	

$$R^2 = 1 - \frac{\text{sum squared regression (SSR)}}{\text{total sum of squares (SST)}}, \\ = 1 - \frac{\sum(y_i - \hat{y}_i)^2}{\sum(y_i - \bar{y})^2}.$$

$$R^2_{\text{adjusted}} = 1 - \frac{(1 - R^2)(N - 1)}{N - p - 1}$$

where

R^2 = sample R-square

p = Number of predictors

N = Total sample size.

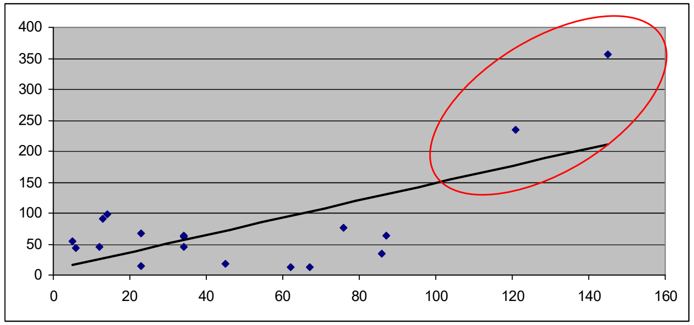
Adjusted R-square

- measures the proportion of variation explained by only those independent variables that really help in explaining the dependent variable.
- penalizes for adding independent variable that do not help in predicting the dependent variable.

R^2 Calculation

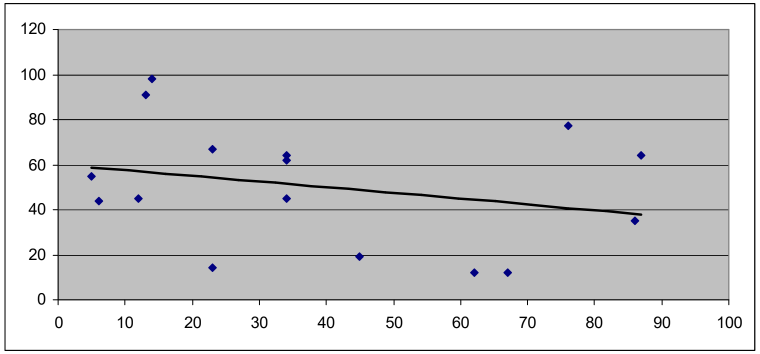
[Numeracy, Maths and Statistics - Academic Skills Kit (ncl.ac.uk)]

Correlation - outliers

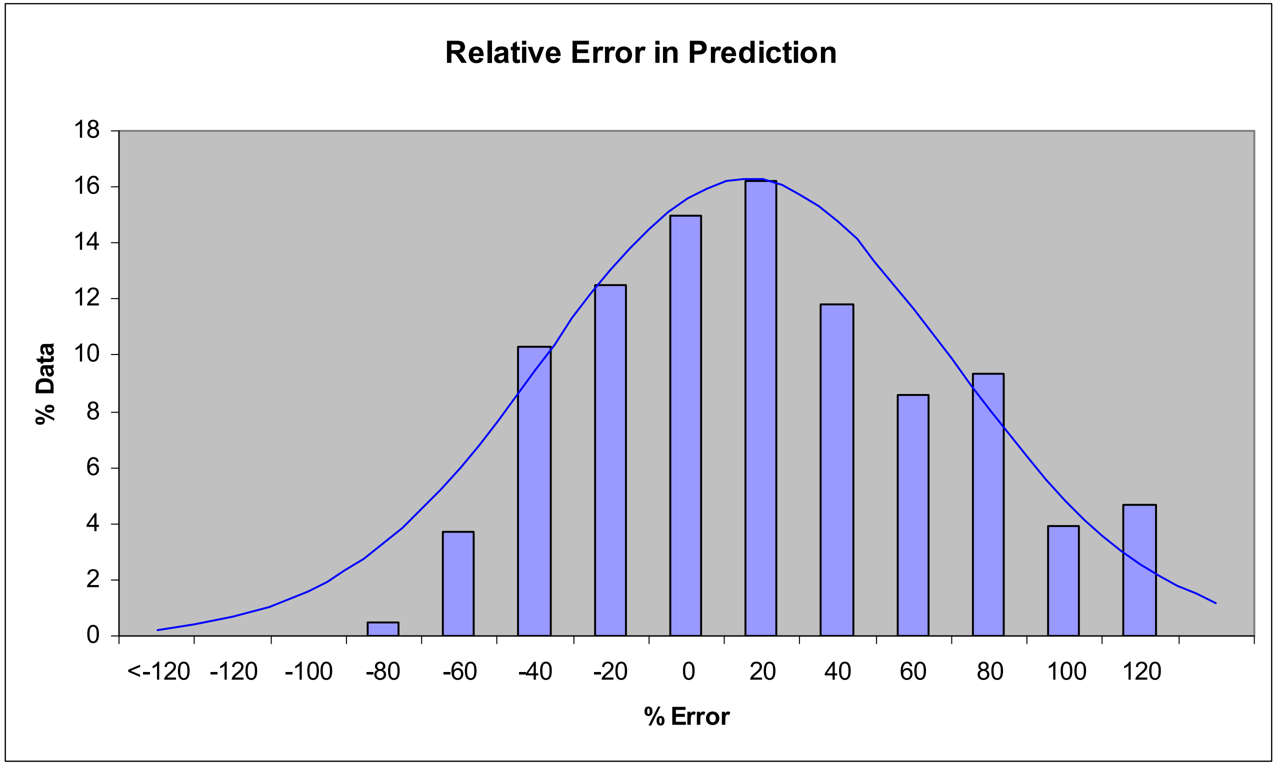


r = 0.66

r = -0.27



Error plot


Relative Error in Prediction
% Data
18
16
14
12
10
8
6
4
2
0
<-120 -120 -100 -80 -60 -40 -20 0 20 40 60 80 100 120
% Error

Classification & Clustering

Supervised vs. Unsupervised Learning

Supervised learning (classification)

- Supervision: The training data (observations, measurements, etc.) are accompanied by labels indicating the class of the observations
- New data is classified based on the training set

Unsupervised learning (clustering)

- The class labels of training data is unknown
- Given a set of measurements, observations, etc. with the aim of establishing the existence of classes or clusters in the data

Classification vs. Regreesion

Classification:

- predicts categorical class labels
- classifies data (constructs a model) based on the training set and the values (class labels) in a classifying attribute and uses it in classifying new data

Regression:

- models continuous-valued functions, i.e., predicts unknown or missing values

Typical Applications

- credit approval
- target marketing
- medical diagnosis
- treatment effectiveness analysis

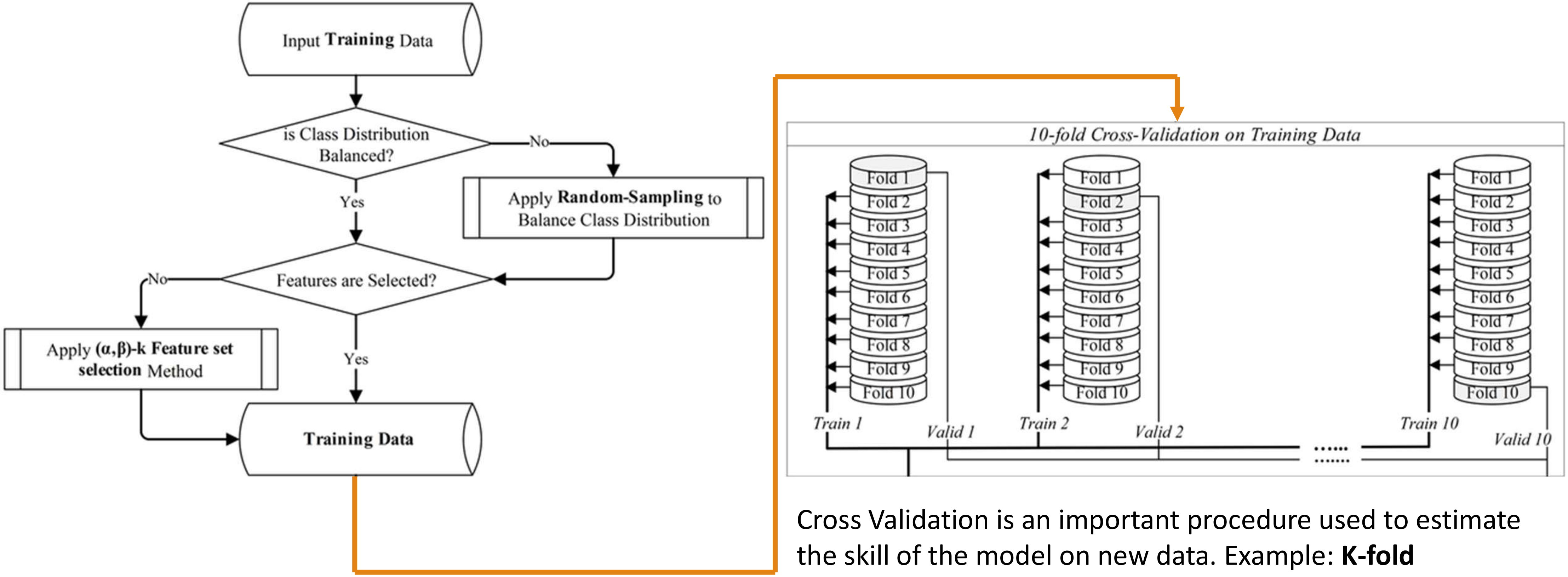
Validation

Cross Validation (10-fold, k-fold)

- Randomly divide training data set in 10 segments
- Train with 9 and test on remaining 1
- Repeat the procedure 10 times
- Training sample should be balanced
  - Nearly equal number of all possible classes

Leave-1-out Validation: same as above, we take one sample as test set and train with the rest

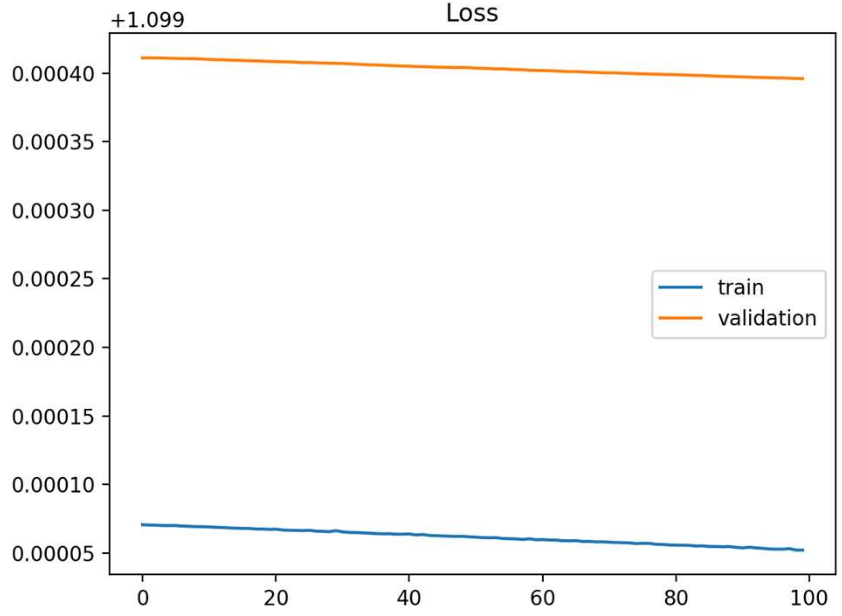
Flow chart of machine learning model



Cross Validation is an important procedure used to estimate the skill of the model on new data. Example: **K-fold**

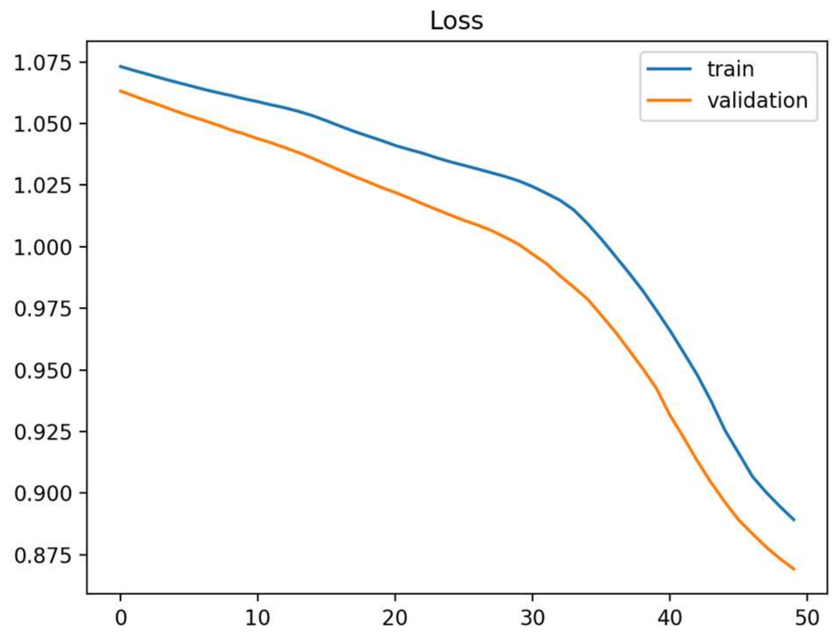
Underfitting

Underfitting
A statistical model is said to be underfited, when it cannot capture the underlying trend of the data
underfit model can be identified from the learning curve of the training loss • It may show a flat line or noisy values of relatively high loss, indicating that the model was unable to learn the training dataset • May also be identified by a training loss that is decreasing and continues to decrease

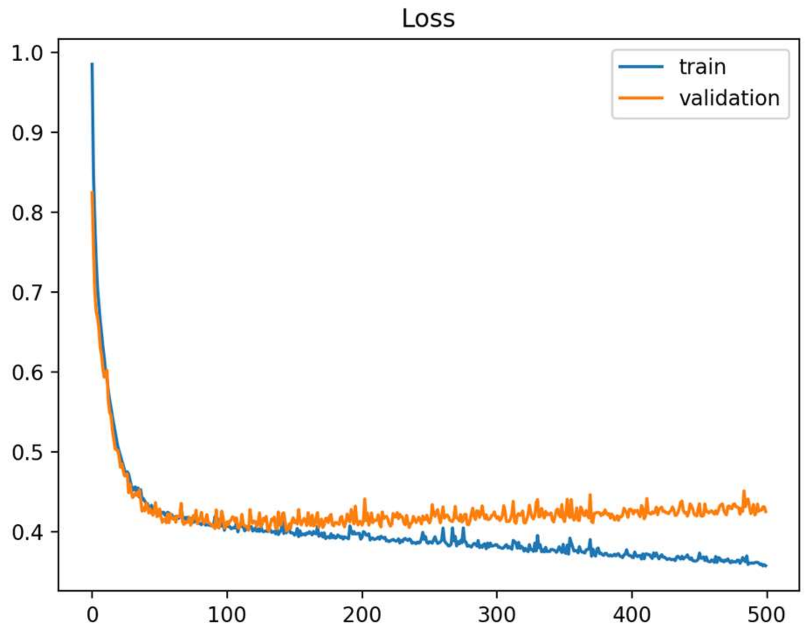


Underfitting

Underfitting
A statistical model is said to be underfited, when it cannot capture the underlying trend of the data
underfit model can be identified from the learning curve of the training loss • It may show a flat line or noisy values of relatively high loss, indicating that the model was unable to learn the training dataset • May also be identified by a training loss that is decreasing and continues to decrease



Overfitting



Overfitting

A statistical model is said to be overfitted, when we train it with a lot of data

A plot of learning curves shows overfitting if

- The plot of training loss continues to decrease with experience.
- The plot of validation loss decreases to a point and begins increasing again.

Classification – Precision & Recall

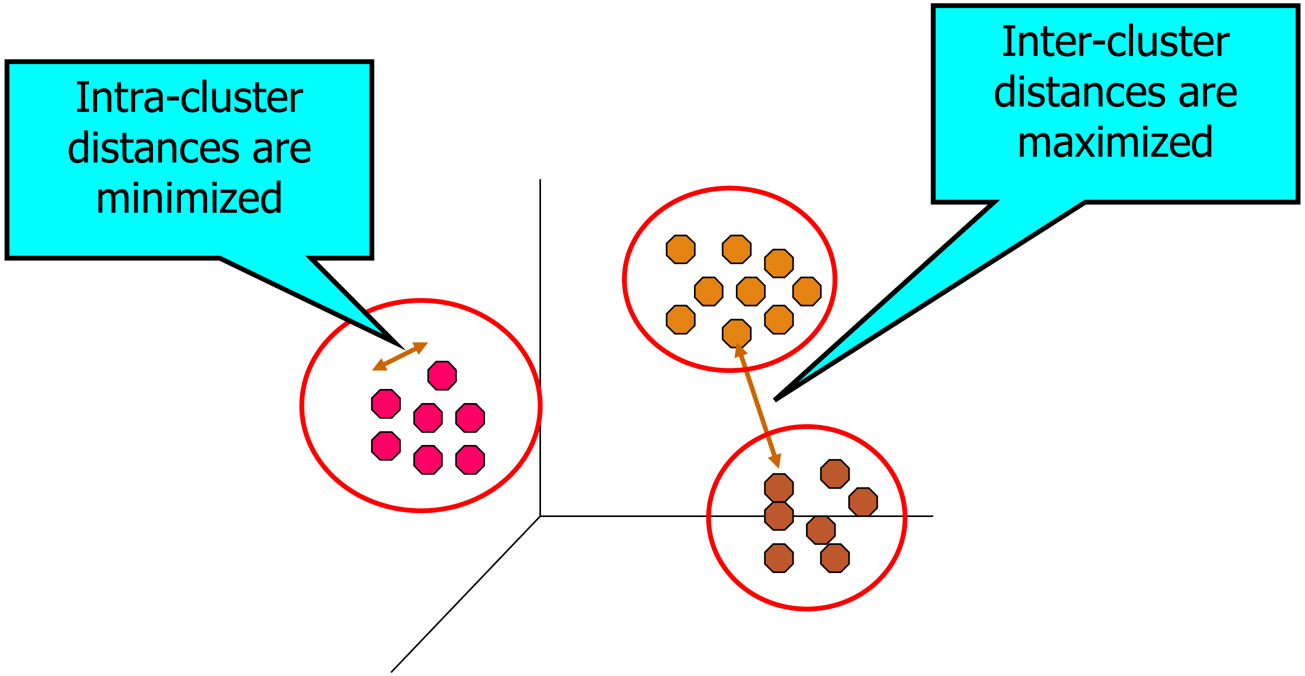
- Precision quantifies the number of positive class predictions that actually belongs to the positive class.
- Recall quantifies the number of positive class predictions made out of all positive examples in the dataset.
- Precision = TruePositives / (TruePositives + FalsePositives)
- Recall = TruePositives / (TruePositives + FalseNegatives)

	Reality	
Prediction	True Positive	False Positive
	False Negative	True Negative

$$F_1 = 2 \cdot \frac{1}{\frac{1}{\mathrm{recall}} + \frac{1}{\mathrm{precision}}} = 2 \cdot \frac{\mathrm{precision} \cdot \mathrm{recall}}{\mathrm{precision} + \mathrm{recall}}$$

What is Cluster Analysis?

Finding groups of objects such that the objects in a group will be similar (or related) to one another and different from (or unrelated to) the objects in other groups



Soft Clustering

What happens when we can not specify the optimum number of clusters beforehand

Can we find the optimum number of clusters?

Two methods can return overlapping clusters

- Fuzzy c-means
- EM Clustering algorithm

Soft Clustering Techniques

FUZZY C-MEANS

Place a set of cluster centres

Assign a fuzzy membership to each data point depending on distance

Compute the new centre of each class

Termination is based on an objective function

Returns cluster centres and membership values of each data point to each cluster

EM ALGORITHM

Assume data came from a set of Gaussian Distribution

Assign data points to distributions and find Expected probability

Update mean and std dev of distributions to Maximize probabilities

Measures of Cluster Validity

Numerical measures that are applied to judge various aspects of cluster validity, are classified into the following three types.

- External Index: Used to measure the extent to which cluster labels match externally supplied class labels.
  - Entropy
- Internal Index: Used to measure the goodness of a clustering structure *without* respect to external information.
  - Sum of Squared Error (SSE)
- Relative Index: Used to compare two different clusterings or clusters.
  - Often an external or internal index is used for this function, e.g., SSE or entropy

Sometimes these are referred to as criteria instead of indices

- However, sometimes criterion is the general strategy and index is the numerical measure that implements the criterion.

Case Studies

Case Study 1 – Cognitive Load Estimation from Ocular Parameters

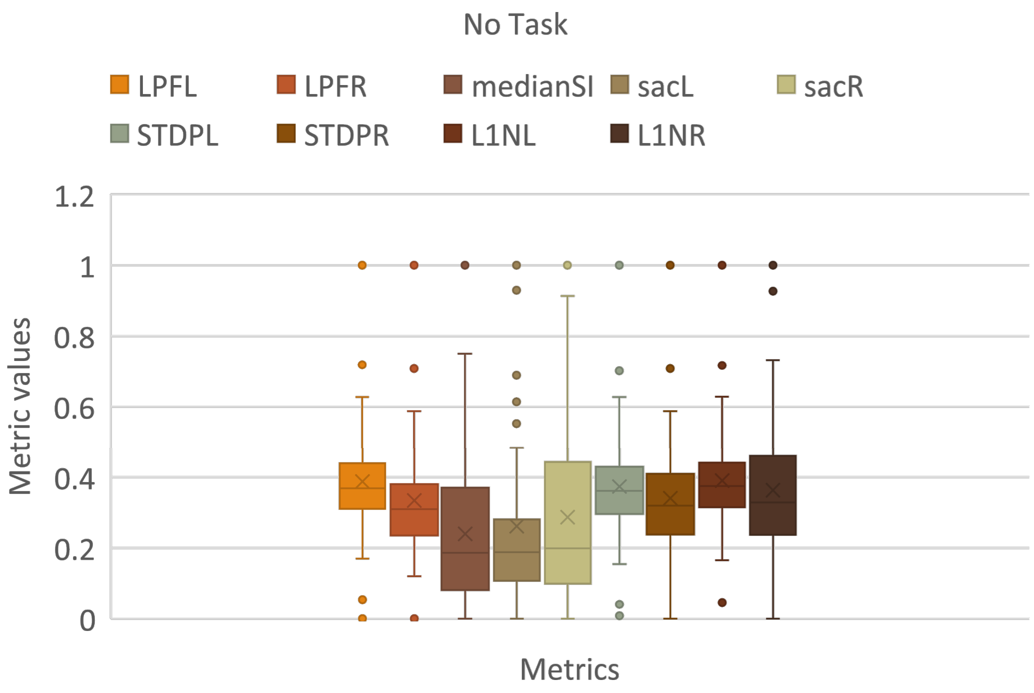
Demonstration

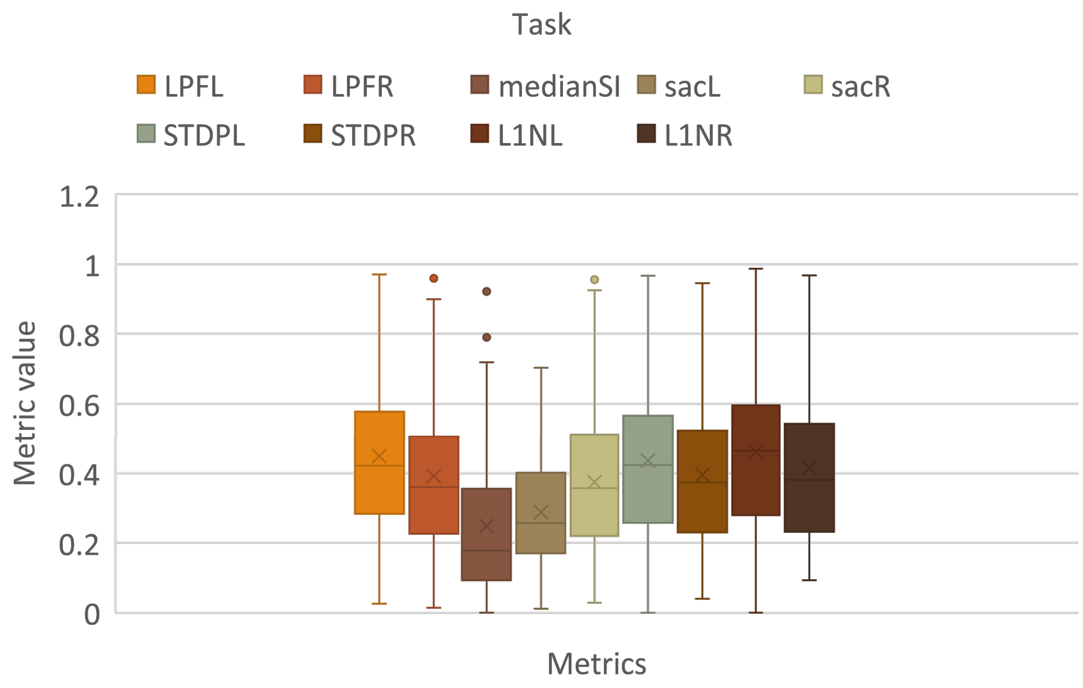
- G. Prabhakar, A. Mukhopadhyay, LRD Murthy, M. Madan, S. Deshmukh and P. Biswas, Cognitive load estimation using Ocular Parameters in Automotive, Transportation Engineering, Elsevier 2020
- P. Biswas, S. Deshmukh, G. Prabhakar, M. Modiksha and A. Mukhopadhyay, System and Method for Monitoring Cognitive Load of a Driver of a Vehicle, Indian Patent Application No.: 201941052358
- M. D. Babu, JeevithaShree DV, G. Prabhakar, KPS Saluja and P. Biswas, Estimating Pilots' Cognitive Load From Ocular Parameters Through Simulation and In-Flight Studies, Journal of Eye Movement Research, Bern Open Publishing, 12 (3), 2019

Dataset preparation

- Analysed and measured ocular parameters and took average of each parameter in tagged time duration
- We have 6 features and 1 prediction vector, i.e., dataset dimension is (26×6)

Average value of parameter corresponding to an event





Training and Testing

STDPL	SMSSL	LPFL	MedianSI	sacl	sacr	Class
0.706459	1.137524	1.828027	49.3185	7.528443	5.505418	0
0.987355	1.530836	2.653351	8.010019	6.559692	7.233281	0
1.05791	1.684927	2.832962	20.94228	41.93676	23.30592	0
1.364831	2.120465	3.673532	5.846828	6.214595	3.726455	0
0.952527	1.495051	2.561194	12.53268	14.98782	5.575253	0
1.004906	1.581594	2.685834	14.80185	12.73494	4.306997	0
1.155556	1.794817	3.104505	7.088327	11.33383	9.864231	0
1.068594	1.672585	2.871536	17.12084	25.50885	29.48941	0
0.997523	1.554821	2.762956	7.757654	8.026166	7.050217	0
0.866554	1.349139	2.350035	18.08792	7.976758	6.305152	0
0.90166	1.403927	2.439939	9.771493	10.49378	9.735114	0
0.90389	1.403503	2.425858	4.601533	2.775808	3.795692	0
1.063928	1.655209	2.866627	10.89931	16.40668	20.90531	0
0.700745	1.091478	1.906794	57.6819	26.58492	12.54121	1
1.146462	1.811556	3.081482	14.24862	16.74549	15.16804	1
1.254093	2.001666	3.385174	30.20105	34.23355	23.86729	1
1.401125	2.194649	3.784753	7.342054	10.08158	7.269943	1
1.040266	1.637844	2.801202	13.30015	19.27478	14.13997	1
1.081412	1.68687	2.910298	17.16687	15.67151	7.225739	1
1.243439	1.93863	3.351217	12.71217	17.34932	14.89335	1
1.072975	1.708304	2.939804	20.14046	28.33814	30.513	1

Training Data (26 × 6)

STDPL	SMSSL	LPFL	MedianSI	sacl	sacr	Class
0.706459	1.137524	1.828027	48.4645	5.961252	5.826803	0
0.702569	1.107083	1.889691	38.65474	26.72293	13.261	1
0.698922	1.075874	1.923896	27.969	20.62204	18.13298	1
1.00639	1.557407	2.668574	6.824375	6.772009	3.762227	0
1.01435	1.572846	2.678924	6.879444	6.090487	8.12065	0
0.981585	1.518262	2.643635	5.218889	6.337558	9.006004	0
0.947095	1.474827	2.622272	13.11737	7.038713	8.044243	0
1.022034	1.681122	2.742683	28.75488	38.64873	25.47953	1
1.111368	1.762064	3.064896	30.21846	30.45515	22.7577	1
1.263466	1.991384	3.372473	18.605	18.54446	17.14362	1
0.943755	1.482271	2.522211	4.28744	5.055612	13.14459	1
1.286281	2.018722	3.426783	4.154595	7.432819	6.003431	1
1.25368	1.971348	3.553604	4.87398	11.33995	9.338781	1
1.140585	1.790685	3.063587	12.22103	12.06637	15.58572	1
1.214173	1.902103	3.288128	4.102436	12.65823	15.30422	1
1.082814	1.704302	2.876858	9.116888	10.02004	9.733753	1
1.072113	1.687796	2.878466	26.00818	54.86076	32.34951	0
1.156814	1.836831	3.117295	18.97778	51.02551	23.67851	0
1.050196	1.649141	2.80376	16.90274	38.69246	22.84857	0
1.043542	1.655777	2.795555	21.90934	27.01621	20.11207	0
0.966884	1.59509	2.569736	20.91336	38.08887	17.54093	0

Test Data (128 × 6)

Sample Prediction

IPython console

Console 1/A

```
[0.361334214380169, 0.36801489488426997, 0.3561004541979096, 0.11060374493385947, 0.09445010278782445, 0.20094260571898273] => 0 (expected 0)
[0.7019738603303165, 0.7166484199694302, 0.7184630006357013, 0.08036529680365295, 0.11775416305363998, 0.10984961972201632] => 0 (expected 0)
[0.5864920140647172, 0.6039534474640601, 0.598568646743004, 0.15580462609476753, 0.153888900109501, 0.13346826990047744] => 0 (expected 0)
[0.352215120513522, 0.3565815839745744, 0.3517689286445165, 0.10942596216568817, 0.10710436746321106, 0.08208151932064538] => 0 (expected 0)
[0.30695937411626373, 0.3844861880204743, 0.31040512042961543, 0.4107526226300086, 0.6885852598827314, 0.4578485480922916] => 1 (expected 0)
[0.040487219162712695, 0.04964323865859199, 0.053615692405943105, 0.23777784906911337, 0.17876877065674457, 0.22848846193076078] => 0 (expected 0)
[0.39475180157067435, 0.40602872799064615, 0.4066277937481979, 0.08104432538791398, 0.16744377810805178, 0.19812603238554605] => 0 (expected 0)
[0.38376671106404325, 0.3979733153764139, 0.3948705437610669, 0.21004566210045664, 0.17644148162218906, 0.08590265467936653] => 0 (expected 0)
[0.3176415075937593, 0.3403111699386895, 0.33740715057248694, 0.37068642862489704, 0.37891069921460324, 0.582483145904518] => 1 (expected 0)
[0.23710345349064776, 0.27296114836915997, 0.24581259492100194, 0.3592085164629516, 0.40404244049274407, 0.06134554075204082] => 0 (expected 0)
[0.3528791864488703, 0.36087990724197405, 0.3709515891904833, 0.10410522601135672, 0.1692080364696452, 0.19777357292374864] => 0 (expected 0)
[0.2964732174132261, 0.3200188418923765, 0.3080828646483533, 0.19074304107450302, 0.14872398558724934, 0.06141728102846576] => 0 (expected 0)
[0.3870269838410225, 0.4001656707929947, 0.41226296690461184, 0.4956673403697882, 0.5520086359008187, 0.913282005850125] => 1 (expected 0)
[0.30946449956662403, 0.31638447192573016, 0.32531836331396646, 0.05866596088244414, 0.10096674042965886, 0.13953329557477695] => 0 (expected 0)
[0.427502924078085, 0.4531363980302542, 0.4396085854411429, 0.5197177290857956, 1.0, 0.8617019730401331] => 0 (expected 0)
[0.32380018818920653, 0.327594719431323, 0.3413317214113323, 0.07508878979393238, 0.09903763550725467, 0.22508762197382517] => 0 (expected 0)
[0.4720022825356347, 0.4792792100953035, 0.4924732086845418, 0.12229501780501106, 0.11143517959759786, 0.1568779700778709] => 0 (expected 0)
[0.3751316636439159, 0.3950364084279526, 0.3861607114832599, 0.18132272518259493, 0.2879347739283329, 0.37518140096880714] => 1 (expected 0)
[0.43134743741411663, 0.4307880749469458, 0.4389197346543421, 0.0366334520334071, 0.19886142269971183, 0.43782558308999375] => 1 (expected 0)
[0.19246272699971945, 0.19799158562197994, 0.22445989452966253, 0.3919766021836536, 0.21220317038667738, 0.28571758857591995] => 1 (expected 0)
[0.40209660908665534, 0.4098843063750487, 0.41336696956292296, 0.219747585897148, 0.25593580506266717, 0.4797922164493394] => 1 (expected 0)
[0.19480382804949536, 0.2144069988680778, 0.21847743888123272, 0.0667289253900527, 0.21545697635647199, 0.3096639067638501] => 0 (expected 0)
[1.0, 1.0, 1.0, 0.0293864168618267, 0.018620411034722957, 0.0] => 0 (expected 0)
[0.5978668861882079, 0.6158348623530373, 0.6101617175569485, 0.13307240704500978, 0.27706291583080783, 0.34004143844003226] => 1 (expected 0)
[0.4520424491657124б, 0.494663622670З568, 0.4589428609571846, 0.4566210045662101, 0.614140917652284l, 1.0] => 1 (expected 0)
[0.3947736743927712, 0.429426229870684, 0.40491034268024895, 0.4320542170607295, 0.48299179699367656, 0.5279675672542585] => 1 (expected 0)
```

[Input vector => Prediction:0/1 (Actual (0/1))]

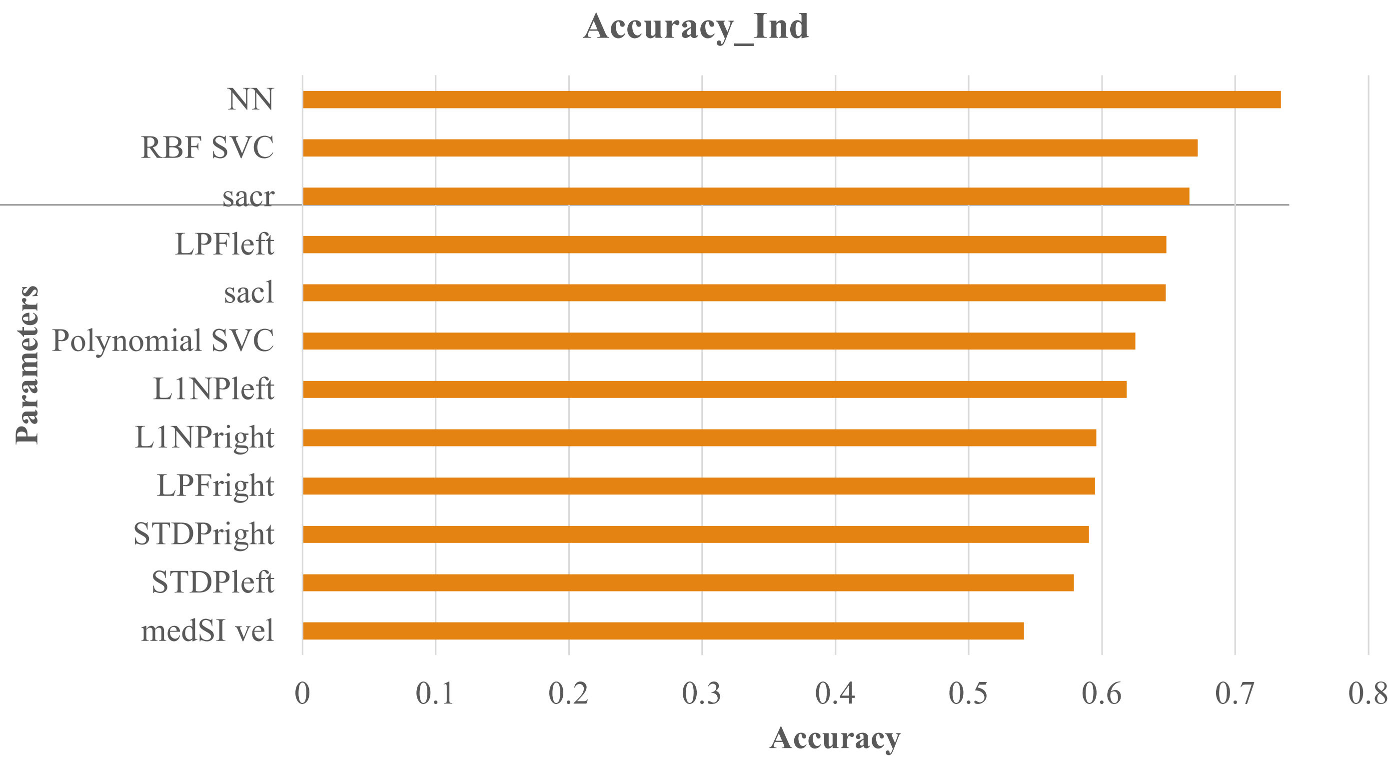
Calculation of Accuracy

We took Task region as positive and No_task region as negative

We counted True positive (TP), False Positive (FP), True Negative (TN), False Negative (FN) as follows:

- TP= If parameter > threshold and lies in Task region
- FP= If parameter>threshold and lies in No_task region
- FN=If parameter<threshold and lies in Task region
- TN=If parameter<threshold and lies in No_task region
- Accuracy=(TP+FP)/(TP+FP+TN+FN)

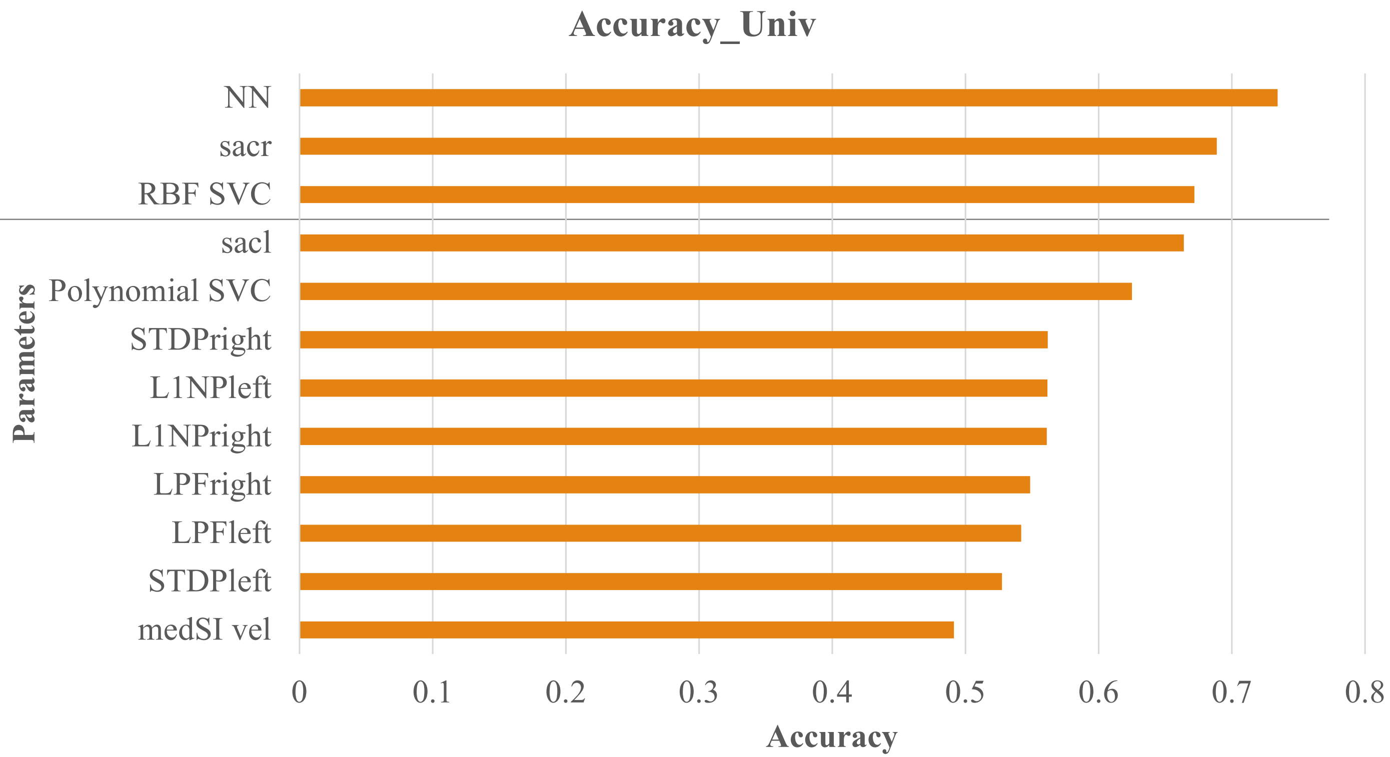
Results - Individual threshold



Calculated accuracy of each parameter by choosing individual threshold corresponding to No_task of each driver

Compared accuracy individual parameters against that of Neural network model to classify

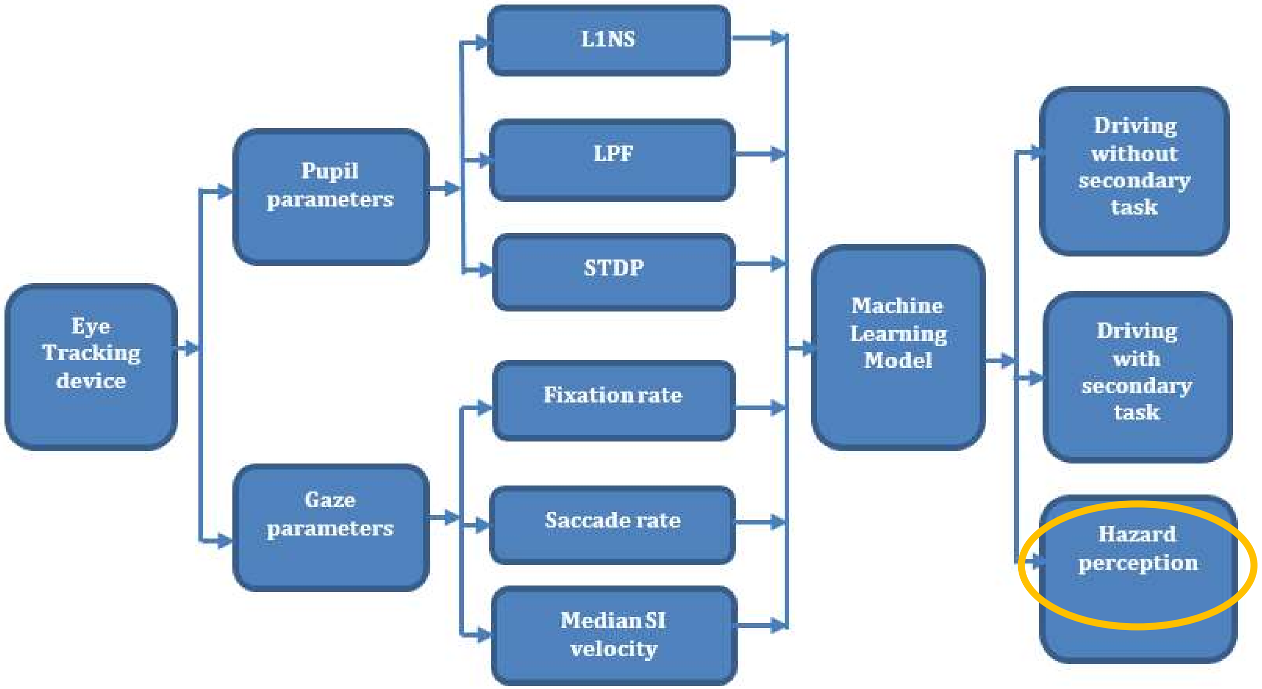
Results - Universal threshold



Calculated accuracy of each parameter by choosing universal threshold which is the average of thresholds corresponding to No_task of each driver

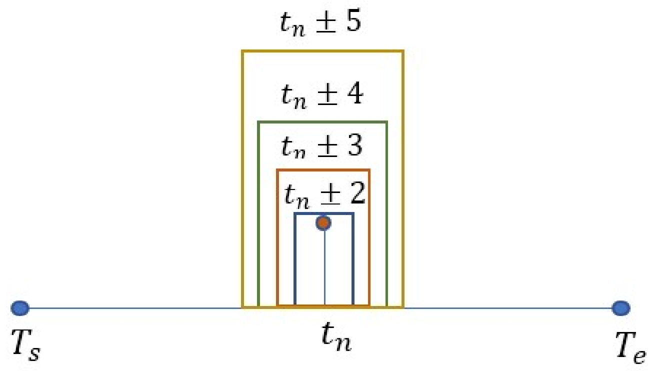
Compared accuracy individual parameters against that of Neural network model to classify

Architecture of proposed model



- **Model structure**: 8 – 160 – 80 – 3
- **Activation function in hidden layers**: ReLU
- **Activation function in output layer**: 'Softmax'
- **Optimizer**: 'Adam'
- **Loss function**: 'categorical cross entropy'

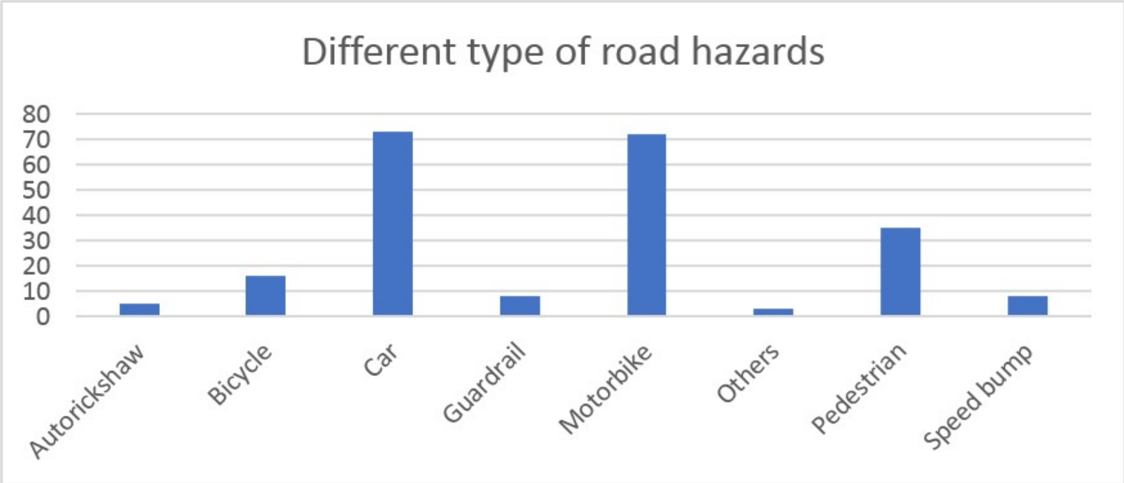
Dataset preparation



T_s : Starting time from when driver stared driving

T_e : End time when driver stopped driving

t_n : event timestamp



- **Followed the guideline of Driver and Vehicle Standards Agency (DVSA), UK to identify developing road hazard**
- **Calculated L1NS, STDP, LPF, saccade rate, fixation rate and median SI velocity in time window duration of ± 2 secs, ± 3 secs, ± 4 secs, and ± 5 secs around the instances of each developing hazard and secondary tasks**
- **Comparative chart between different type of road hazards for the set of driving samples used in our system**

Results and analysis

True \ Predicted	A	B	C
A	T_A	E_{AB}	E_{AC}
B	E_{BA}	T_B	E_{BC}
C	E_{CA}	E_{CB}	T_C

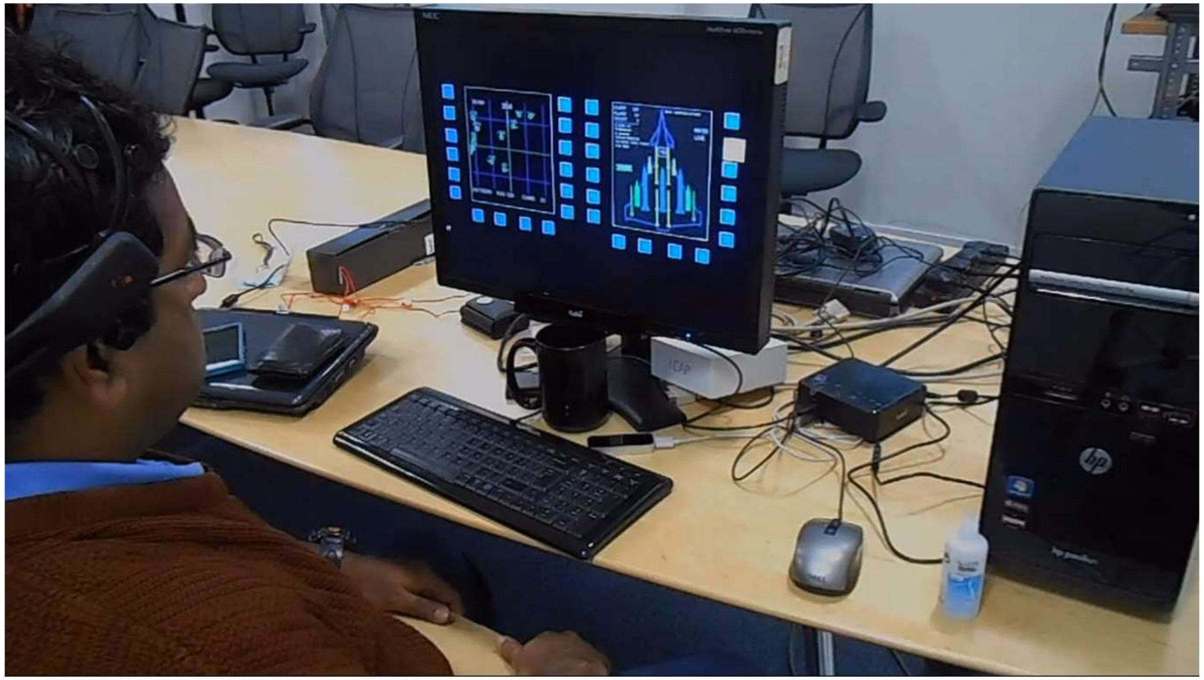
- **Accuracy : ($T_A + T_B + T_C$)/# of test samples.**

	± 2 secs	± 3 secs	± 4 secs	± 5 secs
Training	91.95%	94.62	92.47	84.52%
Test	71.15%	72.44%	70.50%	70.51%

- We found that our model was abled to classify 28 events out of 39 test events correctly
- Accuracy is 72.44 % with ± 3 secs of time window corresponding to road hazards

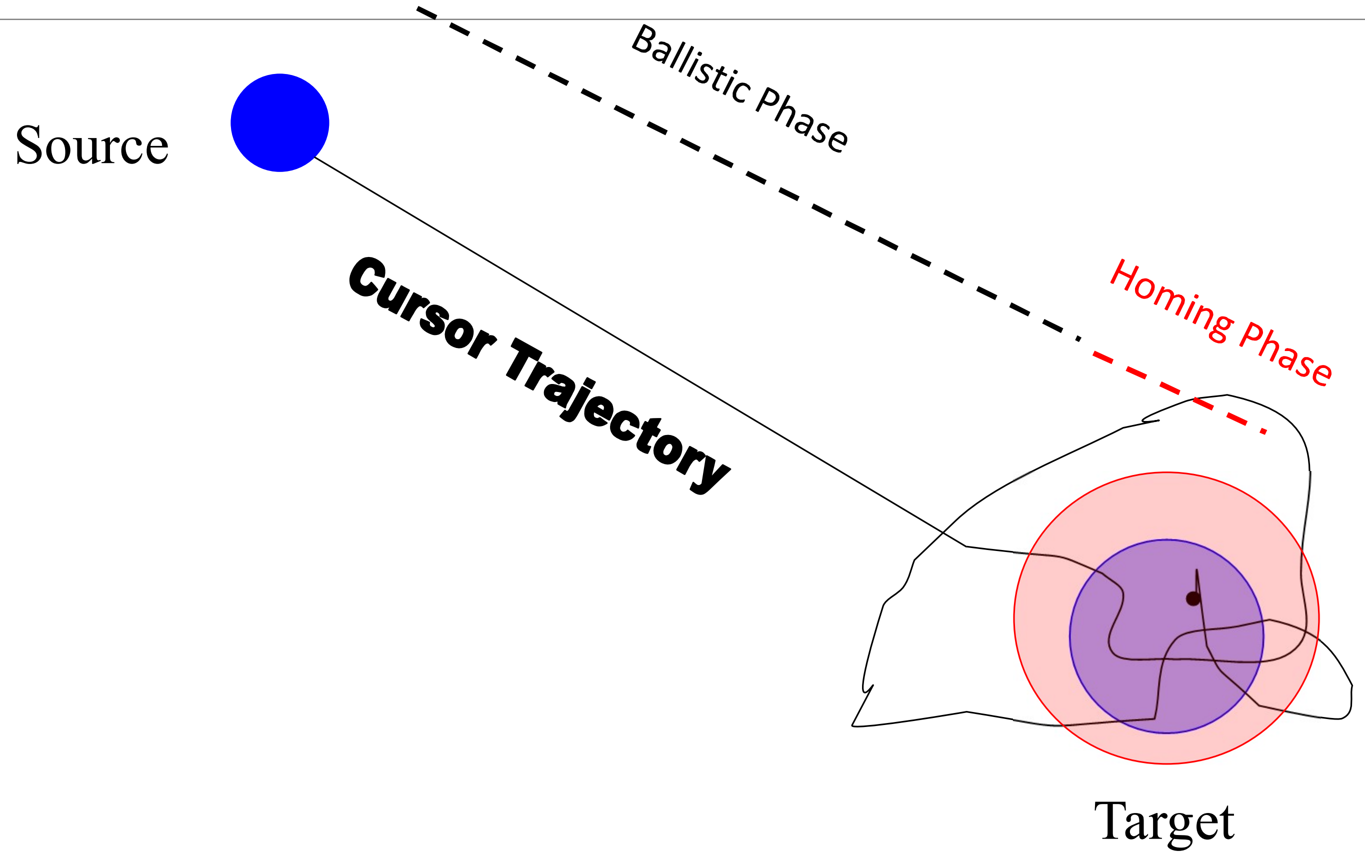
Case Study 2 – Classification for Target Prediction

Demonstration



- P. Biswas and P. Langdon, Mutimodal Target Prediction Model, ACM CHI 2014 Extended Abstract
- P. Biswas, and P. Langdon, Multimodal Intelligent Eye-Gaze Tracking System, International Journal of Human-Computer Interaction 31(4), Taylor & Francis, Print ISSN: 1044-7318

Analysing Trajectory



Algorithm – Neural Network

For every change in position of pointer in screen

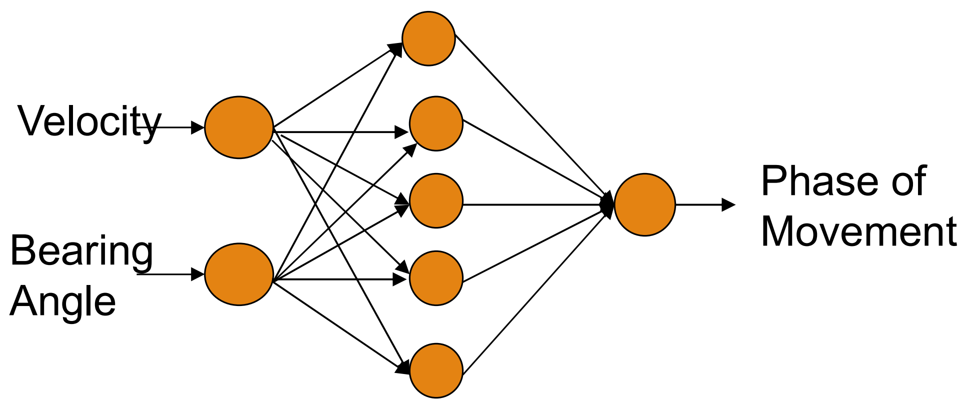
- Calculate angle of movement
- Calculate velocity of movement
- Calculate acceleration of movement

Run Neural Network with Angle, Velocity and Acceleration

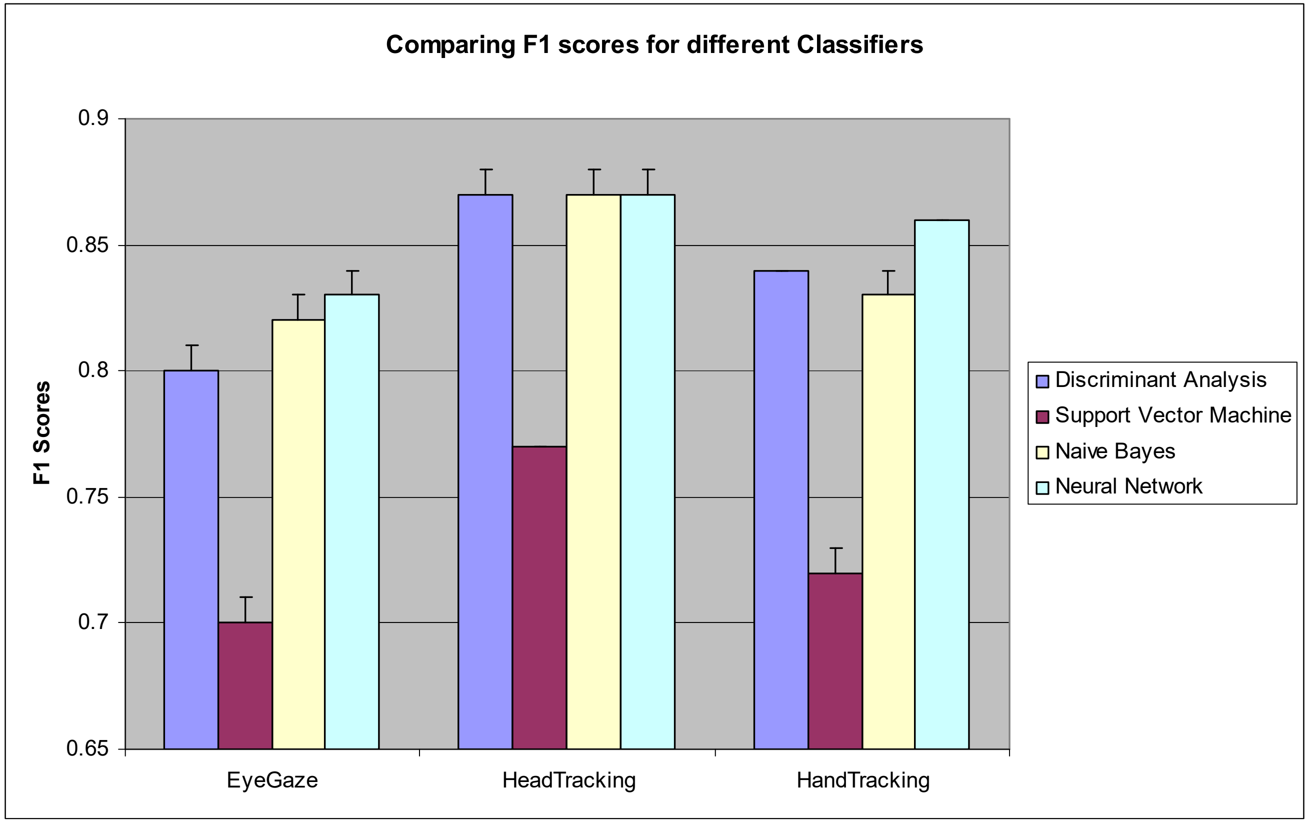
Check output

If output predicts homing phase

- Find direction of movement
- Find nearest target from current location towards direction of movement



Classifier Result

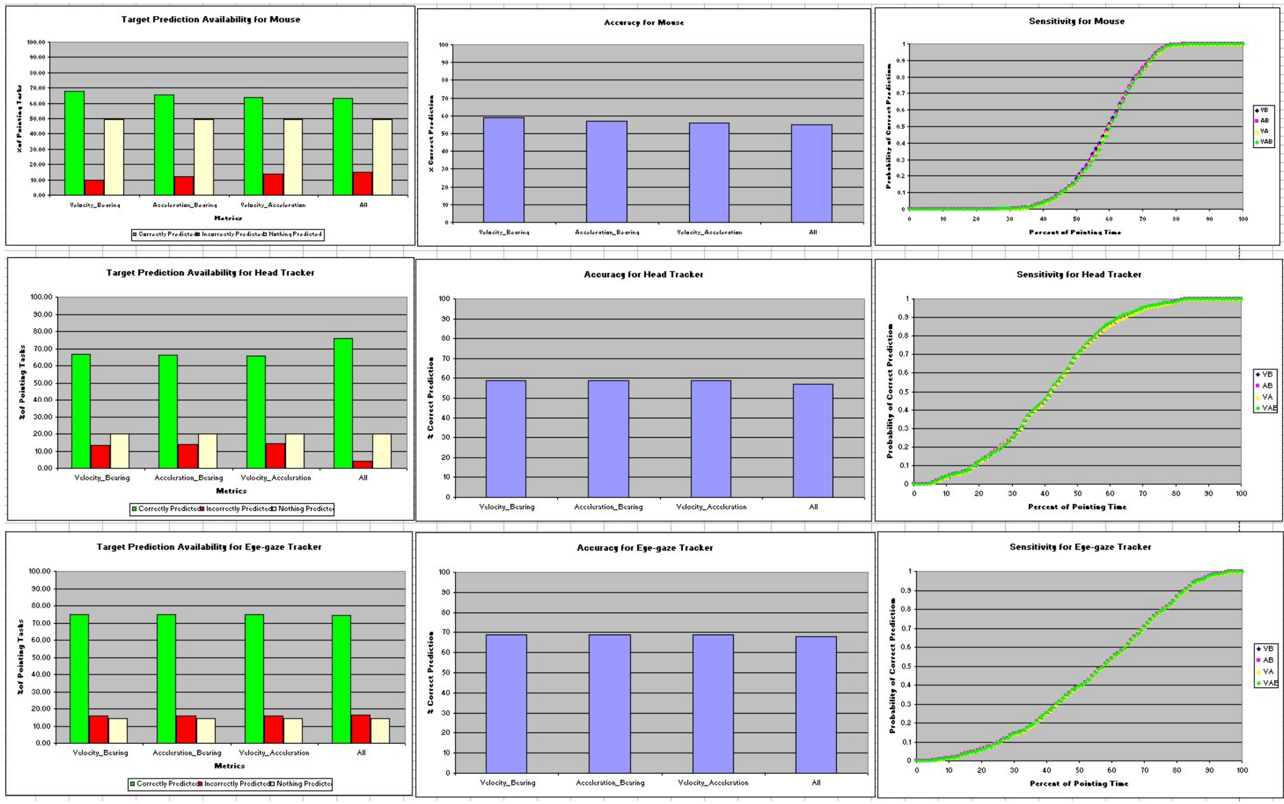


Evaluation Criteria

Availability: In how many pointing tasks the algorithm makes a successful prediction.

Accuracy: Percentage of correct prediction among all predictions

Sensitivity: How quickly an algorithm can detect intended target



Case Study 3 – Gaze Controlled HUD / HMD

Challenge

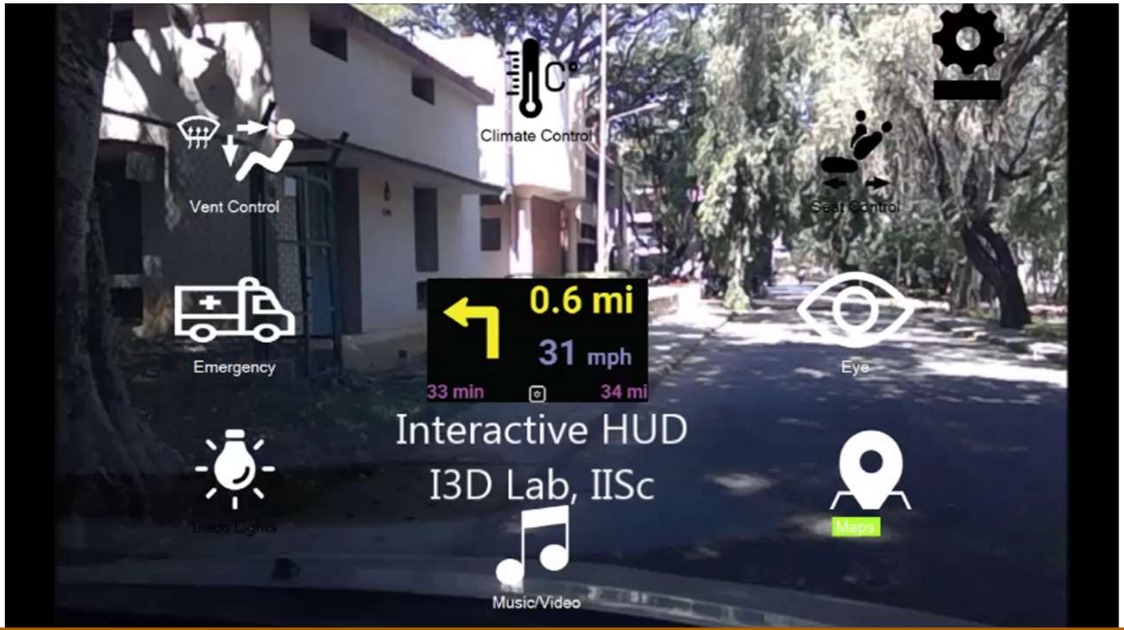
Existing eye trackers are developed for desktop computing environment where

- Tracker is attached below display
- Display is a flat screen

We used eye tracker to track eyes on windshield

Display was away from eye tracker

Display surface was not flat like a computer screen



- G Prabhakar, A Ramakrishnan, LRD Murthy, VK Sharma, M Madan, S Deshmukh and P Biswas, Interactive Gaze & Finger controlled HUD for Cars, Journal on Multimodal User Interface, Springer, 2019
- P. Biswas, S. Deshmukh, G. Prabhakar, M. Modiksha and A. Mukhopadhyay, System and Method for Monitoring Cognitive Load of a Driver of a Vehicle, Indian Patent Application No.: 201941052358, PCT Application No.: PCT/IB2020/062016, US Patent Application No.: 17/437,003

Implementation

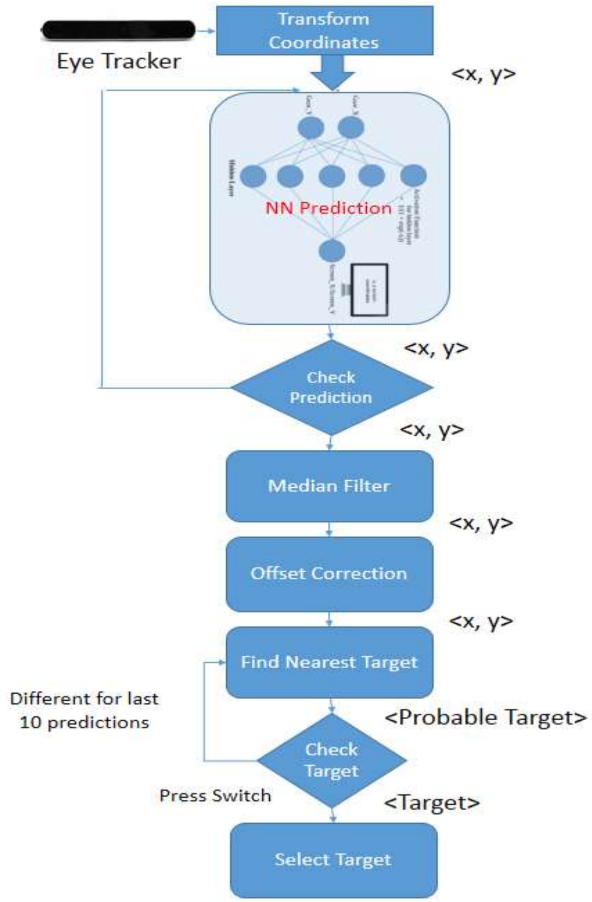
Transform raw gaze coordinates geometrically for inverted image

Run calibration program to train neural net

Filter predicted gaze coordinates

Correct offset based on initial calibration

Activate target nearest to predicted gaze location



Exploration

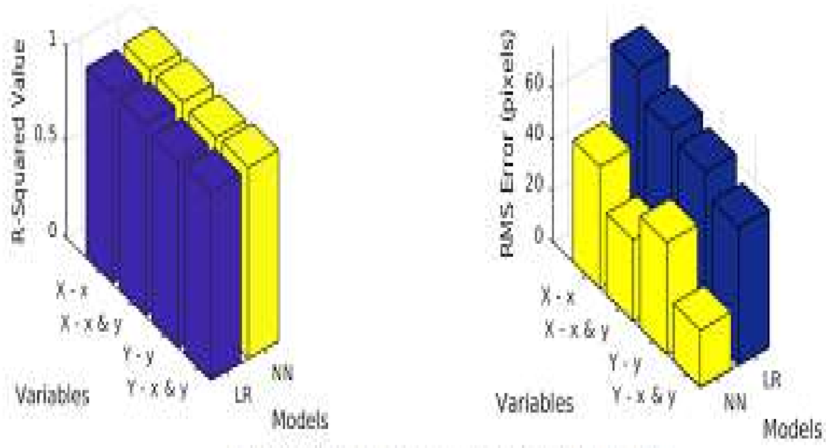
Compared ML systems to convert eye gaze coordinates to screen coordinates on windshield

Set up Linear Regression and Backpropagation Neural Network Models for

- Predicting x-coordinate in screen from x coordinate recorded by gaze tracker
- Predicting x-coordinate in screen from x and y coordinates recorded by gaze tracker
- Predicting y-coordinate in screen from y coordinate recorded by gaze tracker
- Predicting y-coordinate in screen from x and y coordinates recorded by gaze tracker

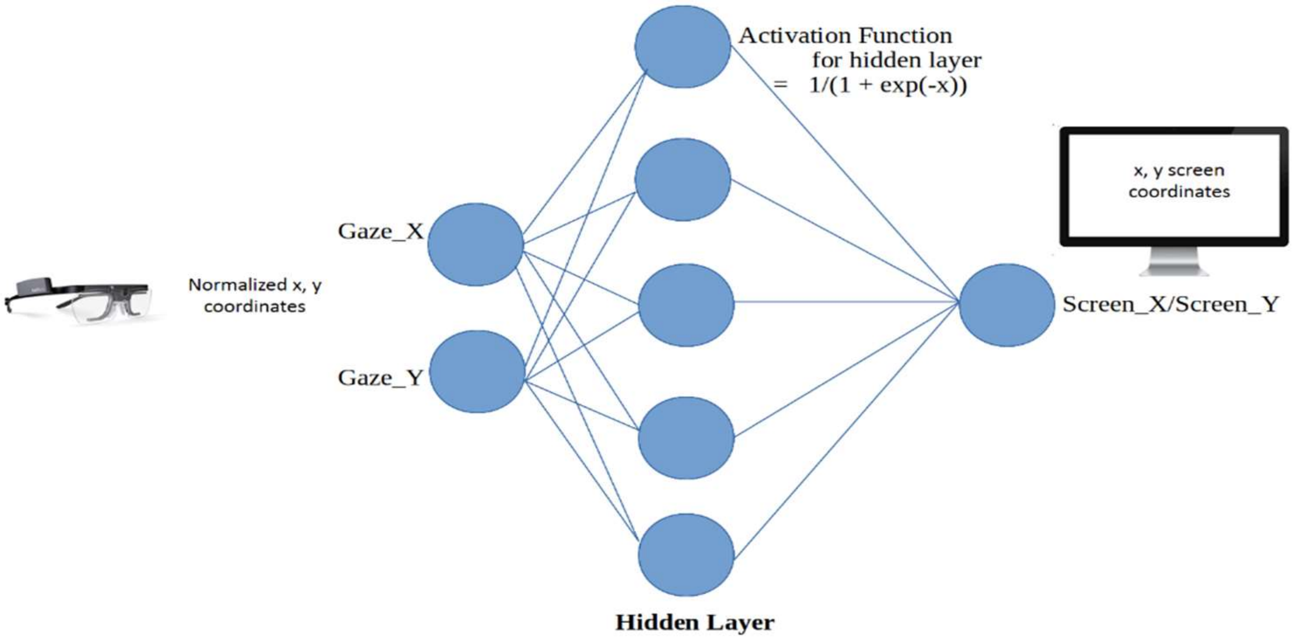
Compared R^2 and RMS error

Neural Network model worked better than Linear Regression



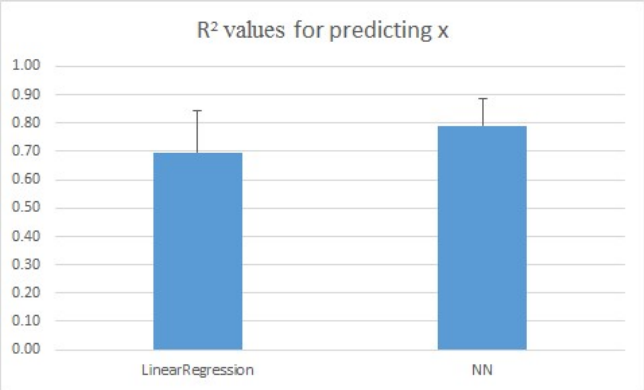
R^2 and RMS error for screen mounted tracker

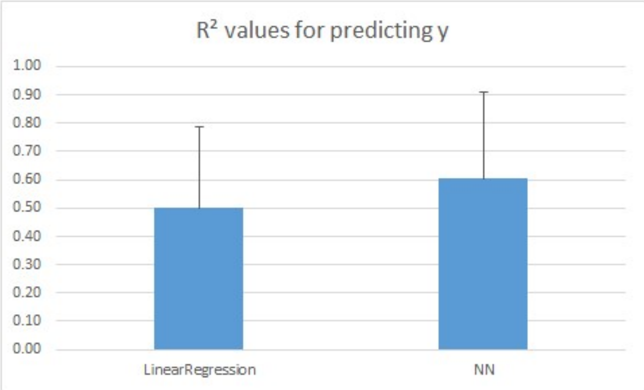
HMD

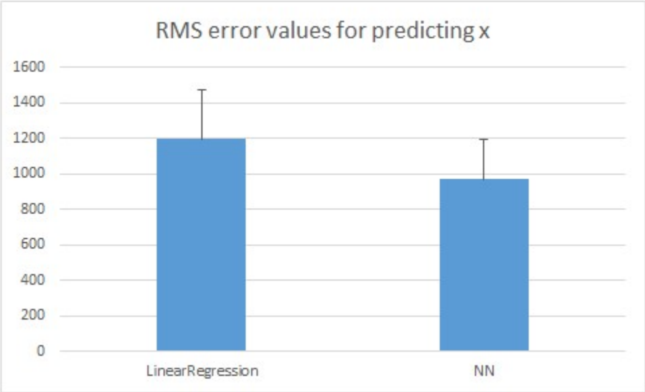


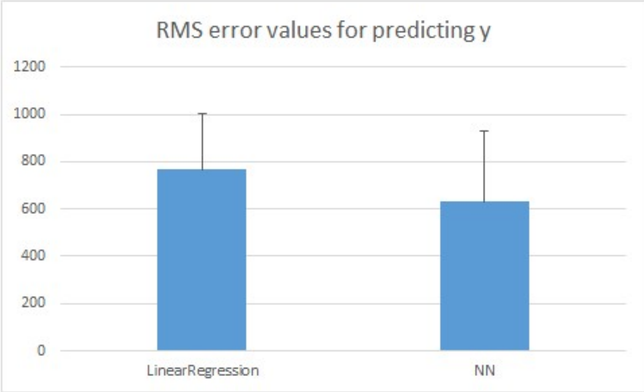
P. Biswas, [Interactive Gaze Controlled Projected Display](), Indian Patent Application No.: 201641037828

Comparison


R² values for predicting x
1.00
0.90
0.80
0.70
0.60
0.50
0.40
0.30
0.20
0.10
0.00
LinearRegression
NN


R² values for predicting y
1.00
0.90
0.80
0.70
0.60
0.50
0.40
0.30
0.20
0.10
0.00
LinearRegression
NN


RMS error values for predicting x
1600
1400
1200
1000
800
600
400
200
0
LinearRegression
NN


RMS error values for predicting y
1200
1000
800
600
400
200
0
LinearRegression
NN

Video Demonstration

- LRD Murthy and P. Biswas, Deep Learning Based Eye Gaze Estimation Algorithms for Military Aviation, IEEE Aerospace 2022
- LRD Murthy, A. Mukhopadhyay, V Yelleti, S Arjun, P Thomas, MD Babu, KPS Saluja, JeevithaShree DV and P. Biswas, Evaluating Accuracy of Eye Gaze Controlled Interface in Military Aviation Environment, IEEE Aerospace 2020
- JeevithaShree DV, KPS Saluja, LRD Murthy and P. Biswas, Operating different displays in military fast jets using eye gaze tracker, Journal of Aviation Technology and Engineering 8(1), Purdue University Press, 2018
- LRD Murthy, A. Mukhopadhyay, S. Arjun, V. Yelleti, P. Thomas, MD Babu, and P Biswas, Eye-Gaze-Controlled HMDS and MFD for Military Aircraft, Journal of Aviation Technology and Engineering 10(2), Purdue University Press, 2021

Summary

- We aim to develop accurate model that generalizes well across unknown dataset
- Statistical hypothesis testing help us
  - To understand contribution of input parameters
  - To test accuracy of prediction
- R^2, Adjusted R^2, Correlation and RMS error are goodness of fit criteria for regression models
- Precision, Recall and F_1 scores indicate accuracy of classification
- Cluster validation indices help to find best number of cluster centres within a pre-set limit.

Take Away Points

TOPICS

Data Screening

Statistical Hypothesis Testing
- Parametric vs Non-Parametric Tests
- Comparing Means – T-test, ANOVA

Load Balancing

Cross Validation

Underfitting and Overfitting

Precision & Recall

Cluster Validation Indices

CASE STUDIES

Cognitive Load Estimation
- Data Preparation
- Local vs Global Threshold

Cursor Trajectory Classification
- Precision
- Recall
- Latency

Gaze Controlled Interface
- R^2
- RMS Error