Team Name – Machine Learners

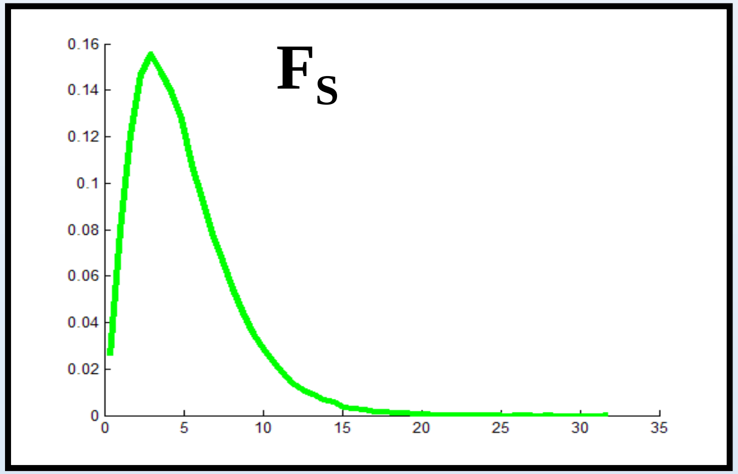
Statistical Change Detection for Multi-Dimensional Data

Course Instructor: Prof P Balamurugan	
Course Code: IE-506	
Machine Learning Principles and Techniques	

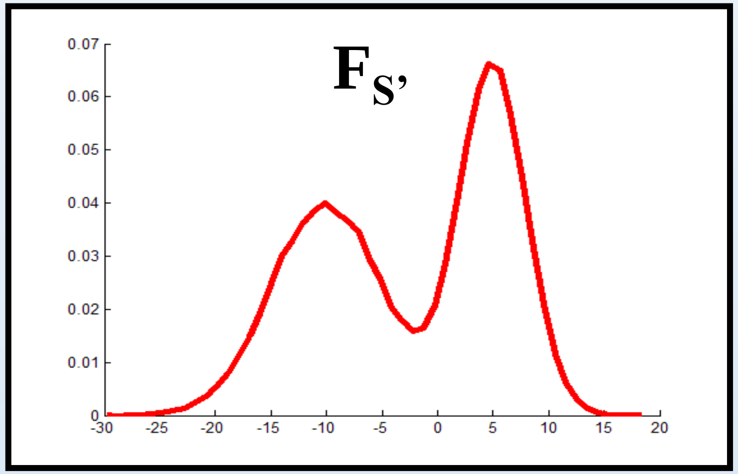
Name	Roll Number
Mayur Dhanawade	23M1512
Shivam Negi	23M1508

Contents

Problem Explanation.



Source: [3]



Source: [3]

Unknown distributions

Data Set (S)

Baseline data

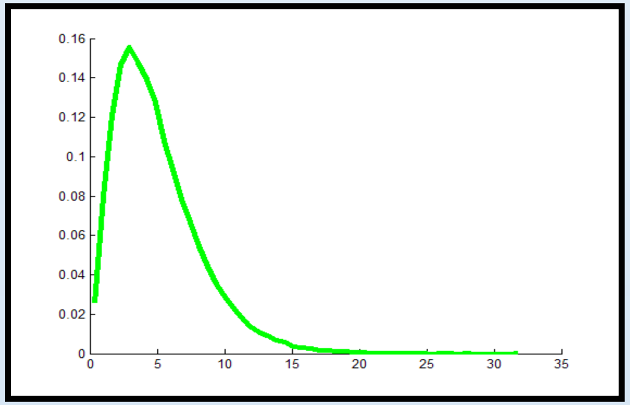
Data Set (S')

Recently observed data

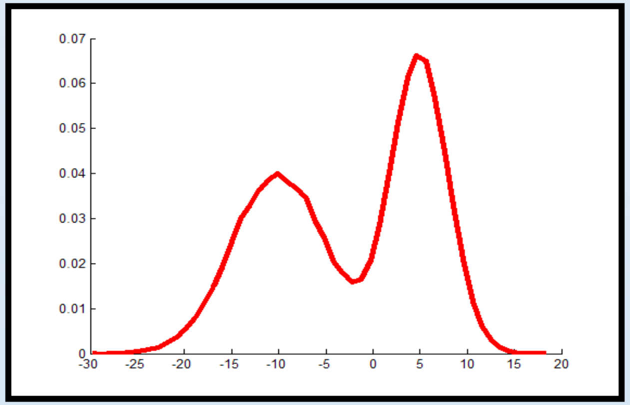
Multi-dimensional space

Motivation for the Problem

- Analyzing the changes in the distribution of financial market.
- **Question: Does financial market shows different pattern recently?**
- We need a distributional change detection method to answer this question.



Source: [3]



Source: [3]

- If Distribution change Observed, In this case, we will take further steps accordingly.

Introduction to Prior Work

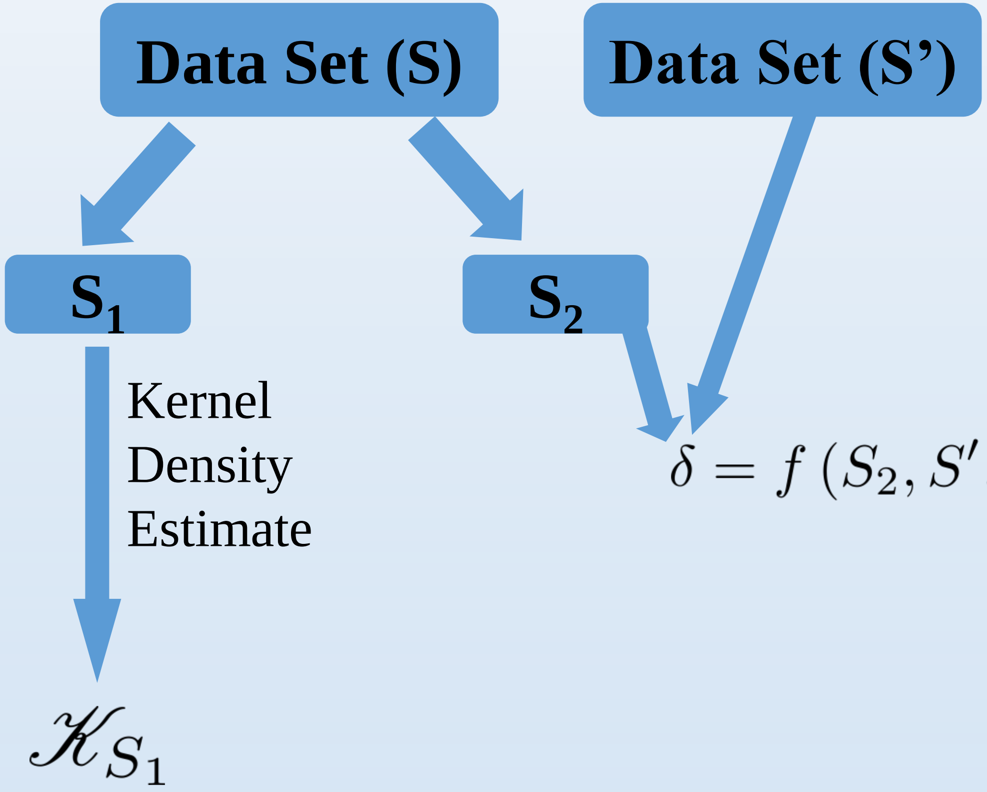
- For uni-dimensional data, many existed tests, such as K-S test, chi-square test and many more.

- Only two tests to detect a generic distributional change in multi-dimensional space.

  - Kdq-tree test: suffer from curse of dimensionality. [1]

  - Cross-match test: computationally expensive due to maximum matching algorithm. [2]

Density test high-level overview



Step 1: Gaussian kernel density estimate of S_1.

Step 2: Define and calculate test statistic δ.

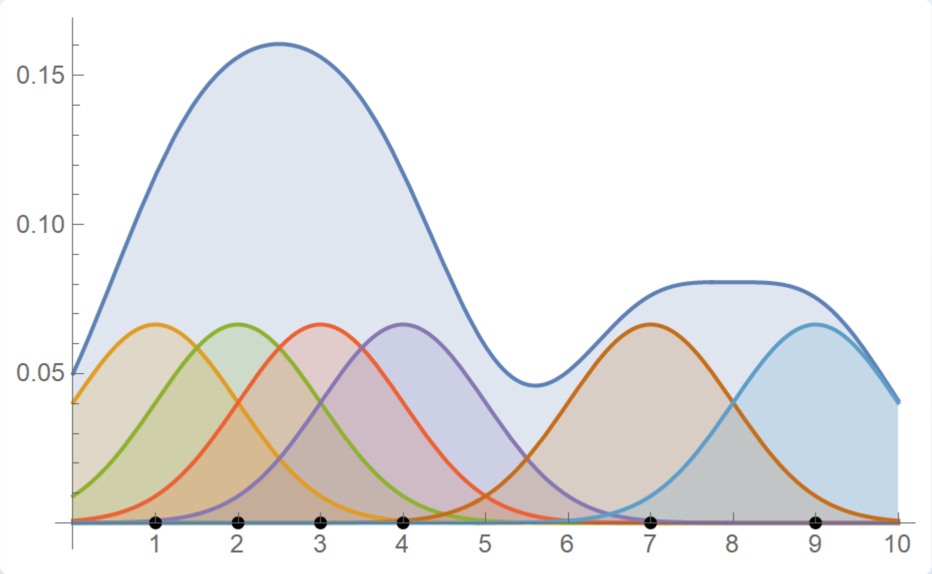
$$\delta = f\left(S_2, S', \mathscr{K}_{S_1}\right) = LLH\left(\mathscr{K}_{S_1}, S'\right) - \frac{|S'|}{|S_2|} \times LLH\left(\mathscr{K}_{S_1}, S_2\right)$$

Step 3: Calculate the critical value, compare δ with critical value.

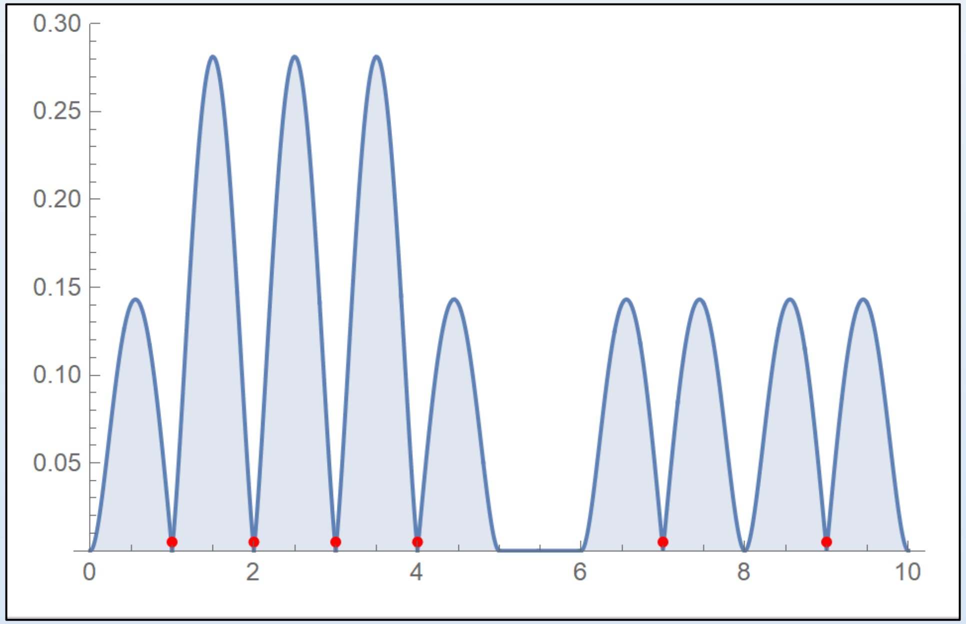
- If δ < critical value, we will declare a change.
- Otherwise, it means no change.

Step 1: Kernel Density Estimate (KDE)

Bandwidth selection



Source: ekamperi.github.io



Source: ekamperi.github.io

- Data-driven bandwidth: converge better to the true distribution.
- Accuracy and power of test is increased when estimate is accurate.

Optimal bandwidth by MLE/EM

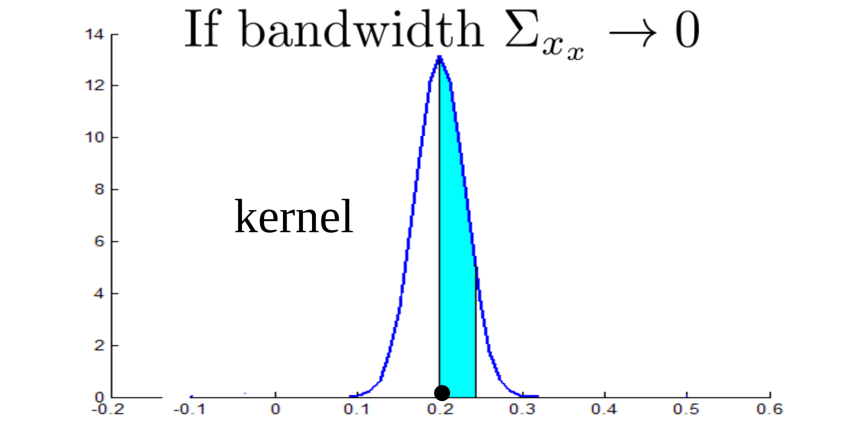
(maximum likelihood estimation / Expectation Maximization)

$$\text{Log Likelihood}(\mathscr{K}_{S_1}, S_1) = \sum_{x_j \in S_1} \log \left[\sum_{x_i \in S_1 \wedge i \neq j} \frac{1}{|S_1| - 1} G\left(\Sigma_{x_i}, x_j - x_i\right) \right]$$

∞

Pseudo – LLH Function

$G(\Sigma_{x_i}, x_i - x_i) \to \infty$



Source: [3]

Adding Constraints $G\left(\Sigma_{x_i}, x_i - x_i\right) = 0$

for all $x_i \in S_1, \quad i = 1, 2, \ldots, |S_1|$

Algorithm 1: Learn Bandwidth

L_{new}

$$\frac{L_{new} - L_{prev}}{L_{prev}} < \phi$$

L_{prev}

Loop Breaks

$$\text{Loglikelihood}(\mathcal{K}_{S_1}(s), \mathcal{S}_1) = L = \sum_{x_j \in S_1} \log \left[\sum_{x_i \in S_1 \wedge i \neq j} \frac{1}{|S_1| - 1} G(\Sigma_{x_i}, x_j - x_i) \right]$$

$$\Sigma_{x_i} = \frac{\sum_{x_j \in S_1} P(\omega_i \mid x_j)(x_j - x_i)(x_j - x_i)^T}{\sum_{x_j \in S_1} P(\omega_i \mid x_j)}$$

$$\begin{cases} P(\omega_i \mid x_j) = \frac{p(x_j \mid \omega_i)}{\sum_{t=1}^{|S_1|} p(x_j \mid \omega_t)} & i, j = 1, 2, \ldots |S_1|, i \neq j \\ P(\omega_i \mid x_i) = 0 & i = 1, 2, \ldots |S_1| \end{cases}$$

Initial Guess Bandwidth

S_1 Data Set

$$\begin{cases} p(x_j \mid \omega_i) = \frac{1}{(2\pi)^{k/2} |\Sigma_{x_i}|^{1/2}} \exp\left(-\frac{1}{2}(x_j - x_i)^T \Sigma_{x_i}^{-1} (x_j - x_i)\right) \\ p(x_i \mid \omega_i) = 0, \quad i, j = 1, 2, \ldots |S_1|, i \neq j \\ \quad i = 1, 2, \ldots |S_1| \end{cases}$$

Step 2: Calculate Test Statistic (δ)

Data set S

Data set S'

Randomly partition

S_2

S_1

Kernel Density Estimate

$\mathscr{K}_{S_1}$

- Small if S' different from S1.
- Large otherwise

Always Large

$$\delta = \mathrm{LLH}\left(\mathscr{K}_{S_1}, S'\right) - \frac{|S'|}{|S_2|} \times \mathrm{LLH}\left(\mathscr{K}_{S_1}, S_2\right)$$

$$= \log\left\{\prod_{y\in S'} \mathscr{K}_{S_1}(y)\right\} - \frac{|S'|}{|S_2|} \times \log\left\{\prod_{y\in S_2} \mathscr{K}_{S_1}(y)\right\}$$

$$= \sum_{y\in S'} \log \sum_{x\in S_1} \frac{1}{|S_1|} G\left(\Sigma_x, y-x\right)$$

$$- \sum_{y\in S_2} \frac{|S'|}{|S_2|} \times \log\left\{\sum_{x\in S_1} \frac{1}{|S_1|} G\left(\Sigma_x, y-x\right)\right\}$$

Step 3: Derive the null distribution (Algorithm 2)

$$\Delta = \underbrace{\sum_{i=1}^{|S'|} \log \sum_{x \in S_1} \frac{1}{|S_1|} G(\Sigma_x, T_i - x)}_{\Delta_1 \sim \text{normal}} - \underbrace{\sum_{i=|S'|+1}^{|S'|+|S_2|} \frac{|S'|}{|S_2|} \times \log \left\{ \sum_{x \in S_1} \frac{1}{|S_1|} G(\Sigma_x, T_i - x) \right\}}_{\Delta_2 \sim \text{normal}}$$

$\Delta \sim$ Normal Distribution by Central Limit$Theorm$.

$$E[\Delta] = 0, \mathrm{Var}[\Delta] = \left(|S'| + \frac{|S'|^2}{|S_2|} \right) \sigma^2$$

$$Where, \sigma^2 = Var[f(T_i)]$$

Need to be estimated

Estimated using S2

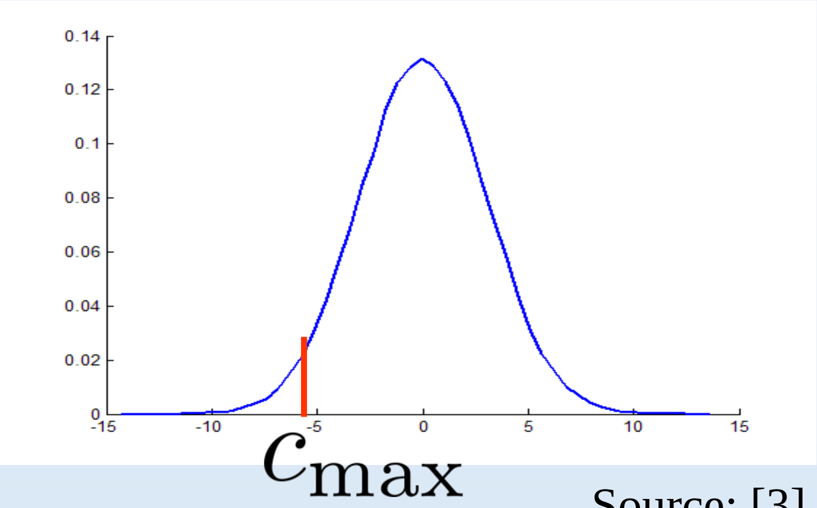
$$f(T_i) = \log \sum_{x \in S_1} \frac{1}{|S_1|} G(\Sigma_x, T_i - x)$$

All random variables $T_i's, i = 1, 2, \ldots$ follow the same distribution F_S.

Step 4: Calculate critical value and make a decision

- $c_{\max}$ Is chosen in such a way that maximum false positive rate is less than user supplied value.
- This is done using algorithm 3.

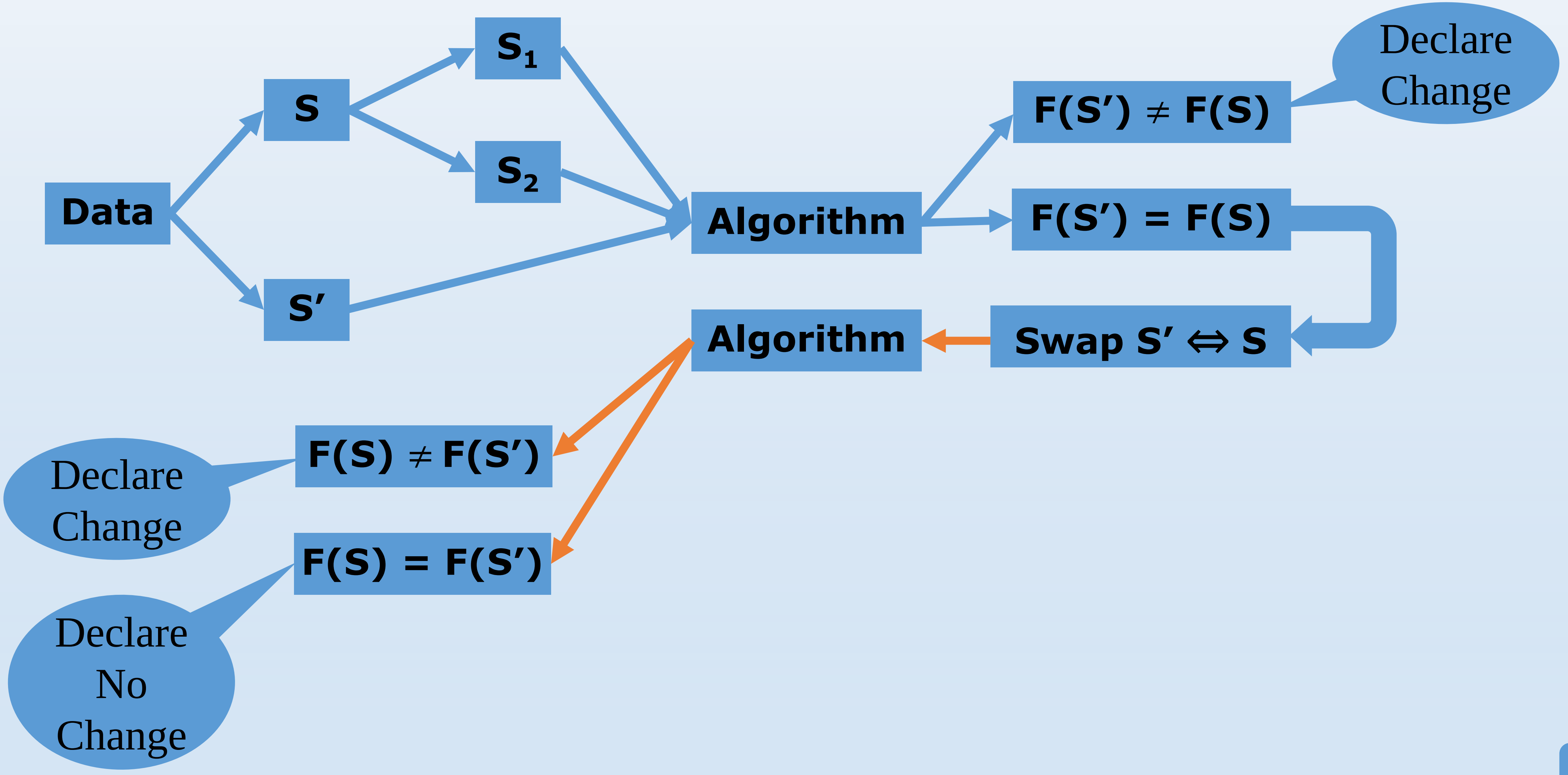
estimated null distribution Δ



Source: [3]

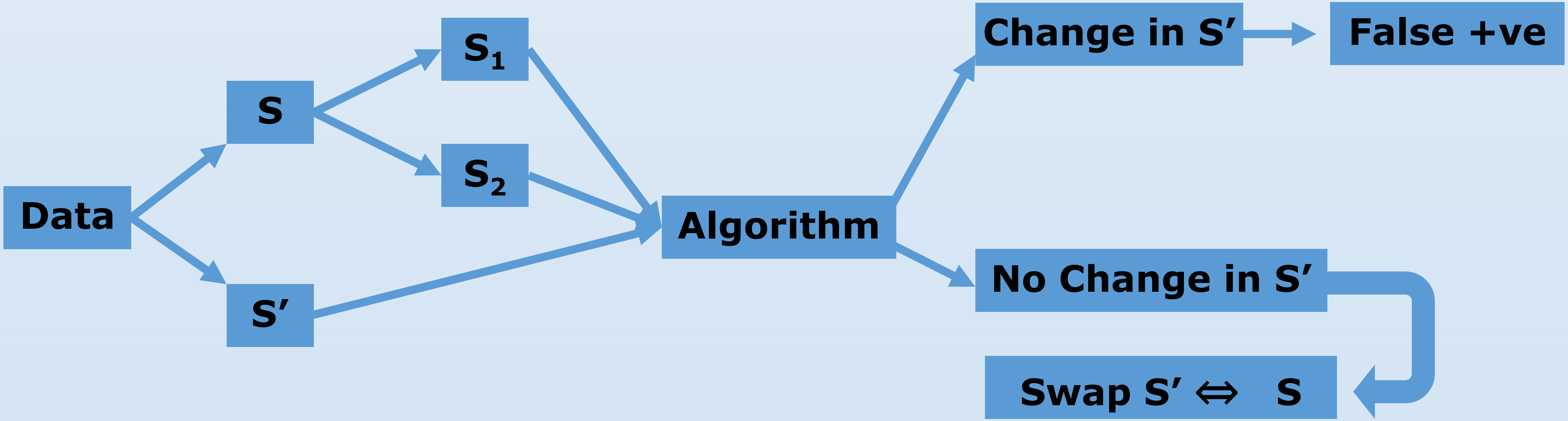
- **Inference:**
  - If $\delta < C_{\max}$ — Declare Change
  - Otherwise — Declare No Change

Two way density test implementation



Experiment 1

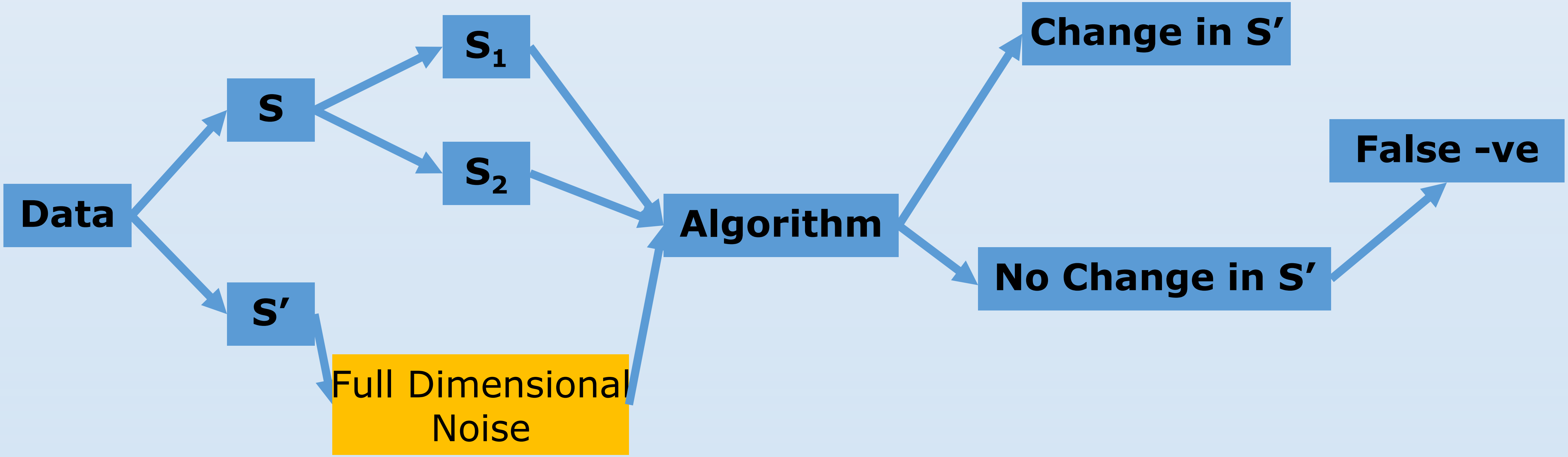
- False Positive Rate Estimation
  - Test for samples from the same distribution.
  - Run for 20 instances.
  - Test distributional change in each instance using two way test.
  - False positive rate = (Number of times it detects change when actually there is no change) / 20.



Experiment 2

Full-Dimensional Changes:

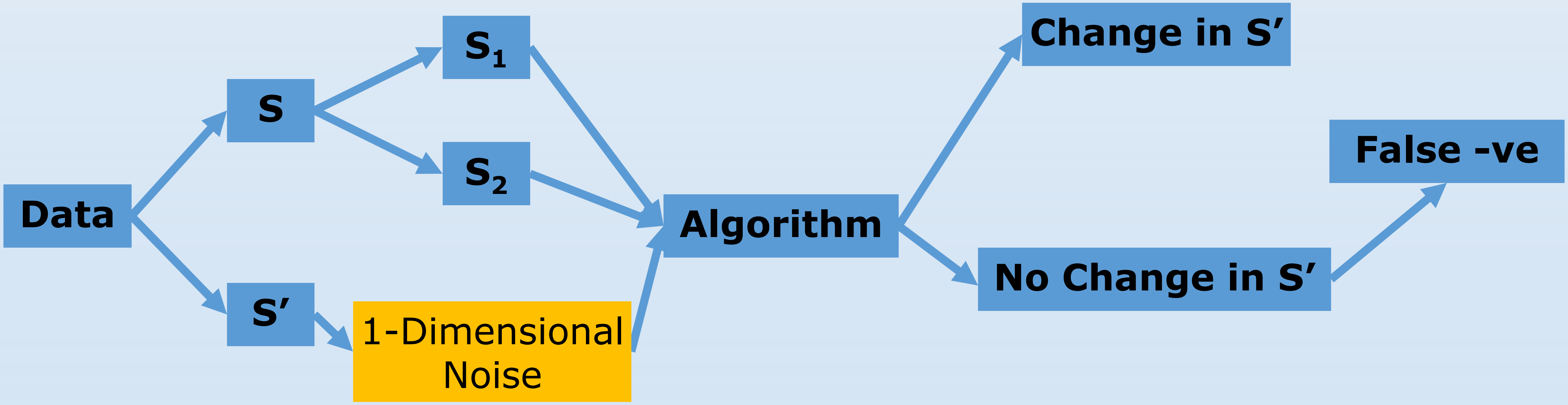
- Add some gaussian noise to all dimensions of some sample points.



Experiment 2

Single-dimensional changes:

- Add Gaussian Noise to a randomly selected dimension.
- Add Noise by multiplying by 2 to a randomly selected dimension.



Our Experiment Results

Experiment 1 Results:

- Expected false positives according to paper is 3 in 100 instances.
- False positives we got is 0 in 20 instances.

Experiment 2 Results:

- Expected false negatives according to paper for gaussian noise is 0 in 100 instances, for add 1-d noise addition is 33 in 100 instances, for scale-1d noise addition is 2 in 100 instances.
- False negatives we got for full dimensional gaussian noise is 0 in 10 instances,
- For add 1-d noise is 3 in 10 instances.
- For scale 1-d noise is 0 in 10 instances.

1) **Data Sets:**
- Worked on El-Nino dataset.

2) **Data Pre-processing:**
- Removed rows having missing values.
- Removed spatio - temporal attributes.
- Final dimensions of dataset: 93935 rows, 5 columns (5-D dataset).

3) **Computational Framework:**
- Intel(R) Xeon(R) CPU E5506, 2.13 GHz, Number of cores = 32, Ram = 128GB

4) **Programming Language:**
- Python

Work done by the team.

Shivam Negi (23M1508):

- Data preprocessing.
- Algorithm 2.
- Experiment 2.1
- Experiment 2.2

Mayur Dhanawade (23M1512):

- Algorithm 1
- Algorithm 3.
- Two way test.
- Experiment 1.

Modification/ Future work

- Currently work done on low dimensional dataset which was 'El-Nino' (5 dimensional).
- Further we intend to work on higher dimensional datasets, which are given in the research paper.

References

[1] Dasu, Tamraparni et al. "An Information-Theoretic Approach to Detecting Changes in Multi-Dimensional Data Streams." (2006). Accessed 30th Mar. 2024.

[2] Rosenbaum, Paul R. "An Exact Distribution-Free Test Comparing Two Multivariate Distributions Based on Adjacency." Journal of the Royal Statistical Society. Series B (Statistical Methodology), vol. 67, no. 4, 2005, pp. 515–30. JSTOR, http://www.jstor.org/stable/3647642, Accessed 30th Mar. 2024.

[3] Image Material: https://www.merlot.org/merlot/viewMaterial, Accessed on: 5th Feb, 2024.

[4] Alicia Horsch towardsdatascience.com, Accessed on: 5th Feb 2024.

[5] Prof Manjesh Kumar Hanawal, youtube.com/playlist, Accessed on: 5th Feb 2024.

[7] Jaroslaw Drapala: towardsdatascience.com, Accessed on: 5th Feb 2024.

[8] Central Limit Theorem: web.stanford.edu, Accessed 2th Feb 2024.

Thank You