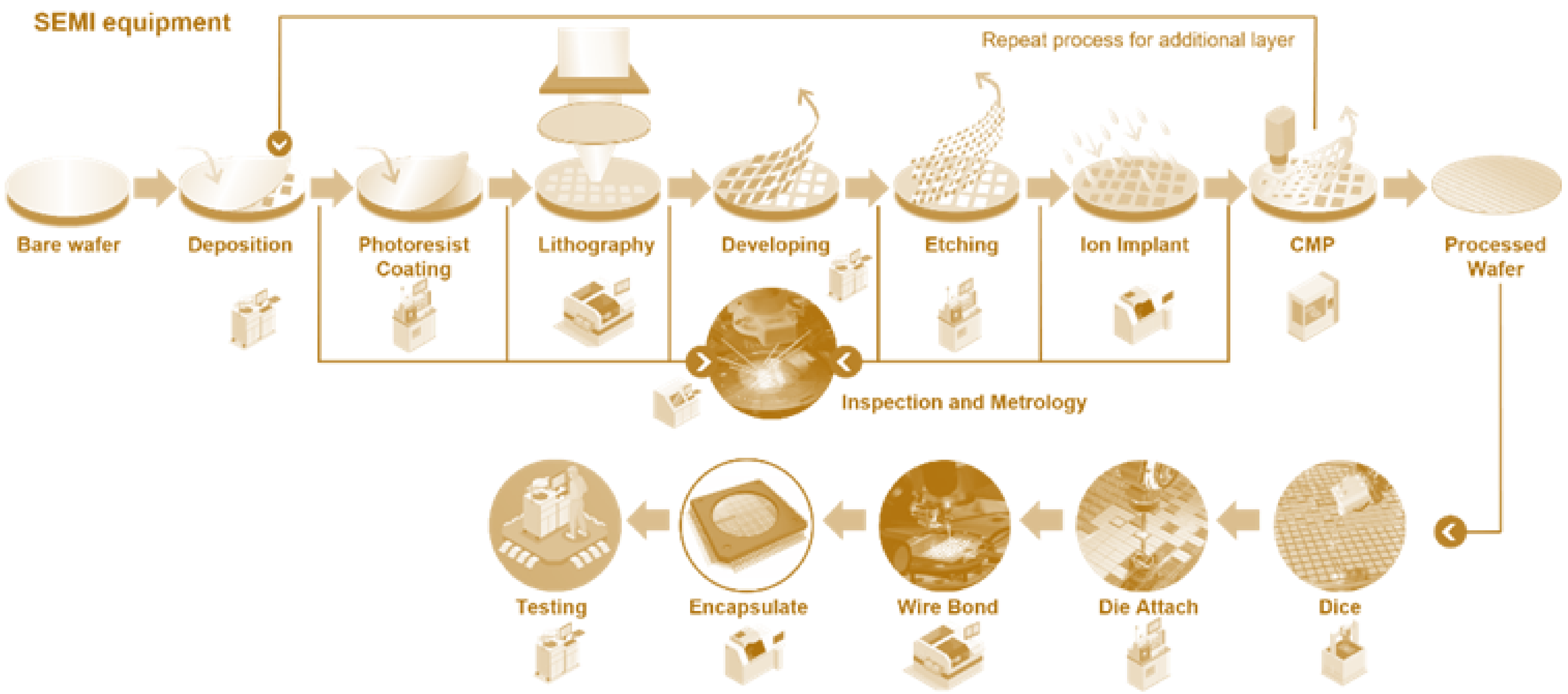
Indian Institute of Technology
Bombay

Pass/Fail Yield Prediction in Semiconductor Manufacturing

ME793 PROJECT STAGE 3

PRATHAM SEHGAL
TANMAY BARHARTE
SHIVPRASAD KATHANE

GUIDE: PROF ALANKAR ALANKAR



A typical semiconductor manufacturing process Ref

PROBLEM STATEMENT

In this project,we build a classifier to predict the Pass/Fail yield of a semiconductor manufacturing product from the sensor measurements data

Detecting deviations in process parameters as recorded by the sensors is helpful to predict defects and a corrective action will help enhance downstream yield/quality

However, a lot of data and measurements (features) may not be relevant (noise)

Hence, application of appropriate feature reduction techniques is crucial

We employ machine learning techniques for defective sample prediction which will help closely approximate the percent yield (# of good quality samples/# of total samples)

Figure: Advantages of Good Yield

Papers such as the following references were reviewed to familiarise with the current/traditional approaches:

1. K Kerdprasop et al., "Feature Selection and Boosting Techniques to Improve Fault Detection Accuracy in the Semiconductor Manufacturing Process", IMECS (2011)
2. AA Nuhu et al., "Machine learning-based techniques for fault diagnosis in the semiconductor manufacturing process: a comparative study", J Supercomput 79, 2031-2081 (2023)

Indian Institute of Technology
Bombay

OUTLINE

Data Visualization

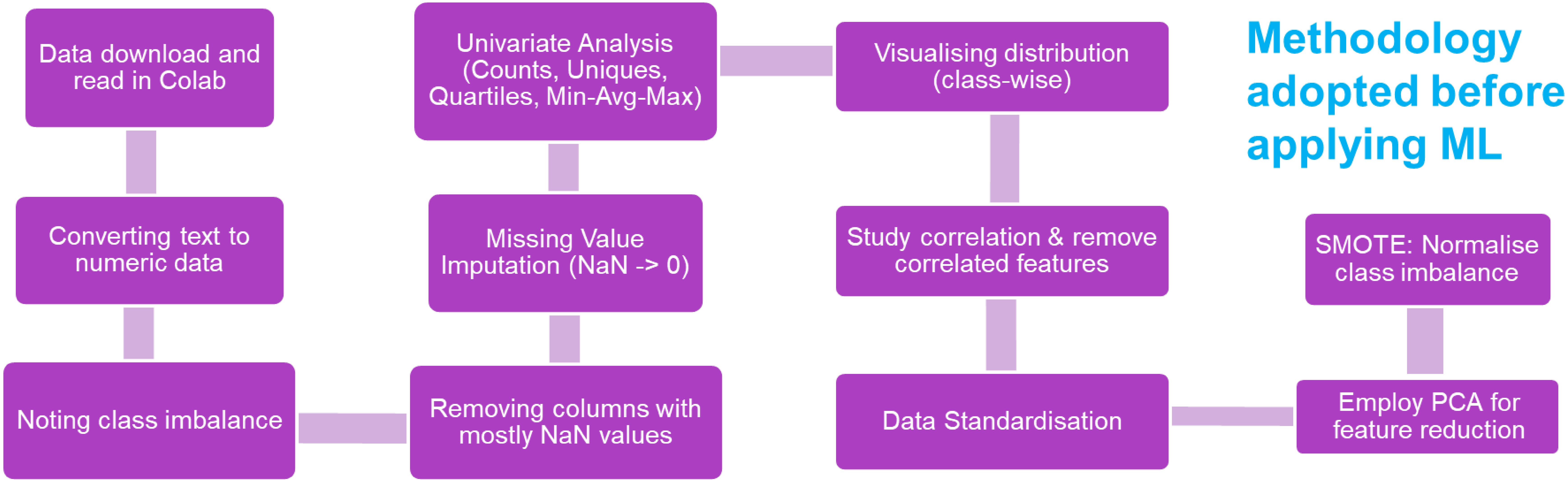
Data Cleaning

Models without PCA

Models with PCA

Conclusions

Methodology adopted before applying ML



Indian Institute of Technology Bombay

Understanding the Data

	0	1	2	3	4	5	6	7	8	9	...	580	581	582	583	584	585	586	587	588	589
0	3095.78	2465.14	2230.4222	1463.6606	0.8294	100	102.3433	0.1247	1.4966	-0.0005	...	0.006	208.2045	0.5019	0.0223	0.0055	4.4447	0.0096	0.0201	0.006	208.2045
1	2932.61	2559.94	2186.4111	1698.0172	1.5102	100	95.4878	0.1241	1.4436	0.0041	...	0.0148	82.8602	0.4958	0.0157	0.0039	3.1745	0.0584	0.0484	0.0148	82.8602
2	2988.72	2479.9	2199.0333	909.7926	1.3204	100	104.2367	0.1217	1.4882	-0.0124	...	0.0044	73.8432	0.499	0.0103	0.0025	2.0544	0.0202	0.0149	0.0044	73.8432
3	3032.24	2502.87	2233.3667	1326.52	1.5334	100	100.3967	0.1235	1.5031	-0.0031	...	NaN	NaN	0.48	0.4766	0.1045	99.3032	0.0202	0.0149	0.0044	73.8432
4	2946.25	2432.84	2233.3667	1326.52	1.5334	100	100.3967	0.1235	1.5287	0.0167	...	0.0052	44.0077	0.4949	0.0189	0.0044	3.8276	0.0342	0.0151	0.0052	44.0077
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
1561	2899.41	2464.36	2179.7333	3085.3781	1.4843	100	82.2467	0.1248	1.3424	-0.0045	...	0.0047	203.172	0.4988	0.0143	0.0039	2.8669	0.0068	0.0138	0.0047	203.172
1562	3052.31	2522.55	2198.5667	1124.6595	0.8763	100	98.4689	0.1205	1.4333	-0.0061	...	NaN	NaN	0.4975	0.0131	0.0036	2.6238	0.0068	0.0138	0.0047	203.172
1563	2978.81	2379.78	2206.3	1110.4967	0.8236	100	99.4122	0.1208	NaN	NaN	...	0.0025	43.5231	0.4987	0.0153	0.0041	3.059	0.0197	0.0086	0.0025	43.5231
1564	2894.92	2532.01	2177.0333	1183.7287	1.5726	100	98.7978	0.1213	1.4622	-0.0072	...	0.0075	93.4941	0.5004	0.0178	0.0038	3.5662	0.0262	0.0245	0.0075	93.4941
1565	2944.92	2450.76	2195.4444	2914.1792	1.5978	100	85.1011	0.1235	NaN	NaN	...	0.0045	137.7844	0.4987	0.0181	0.004	3.6275	0.0117	0.0162	0.0045	137.7844

1566 rows × 590 columns

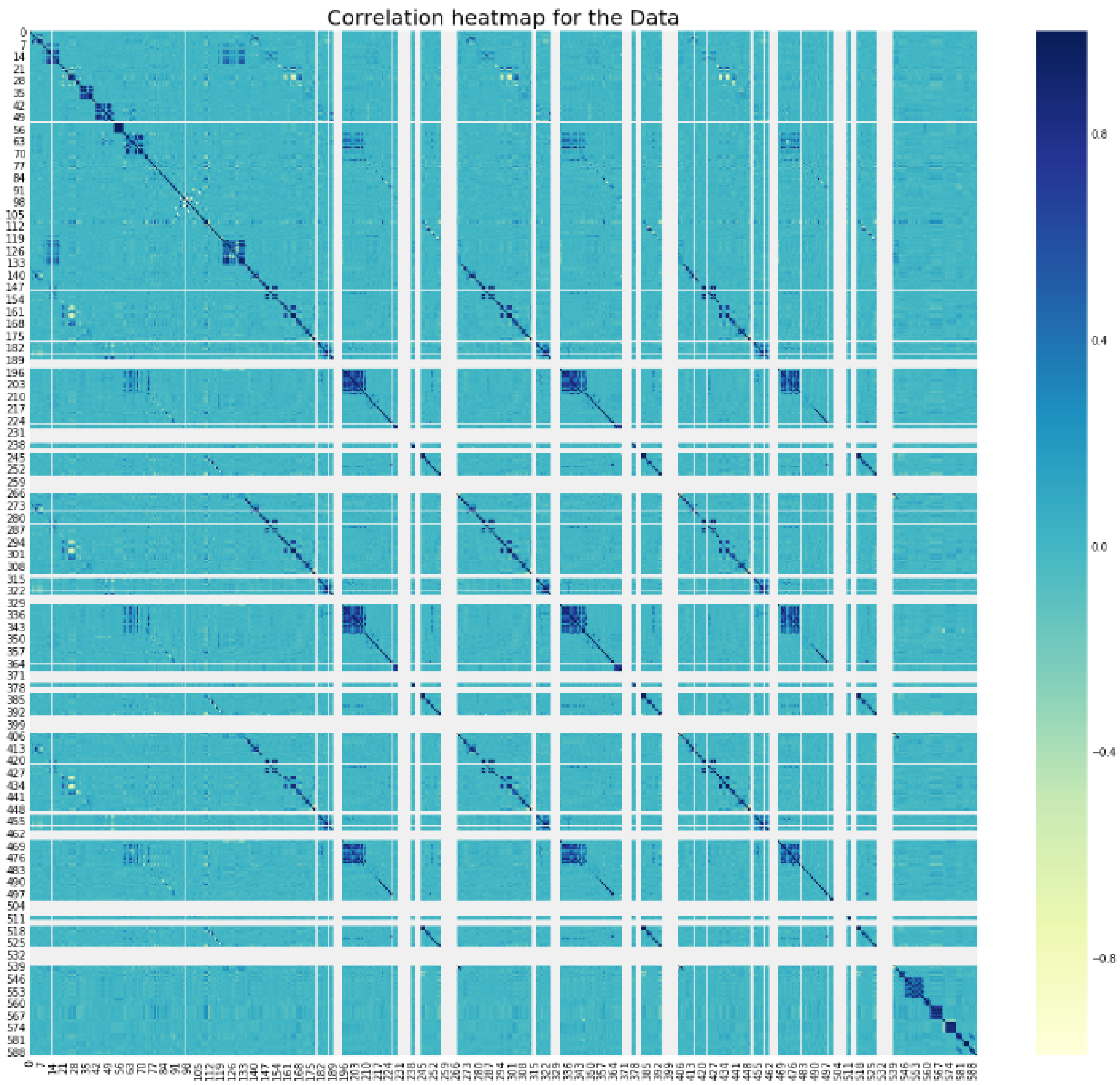
A snapshot of measurement data from first and last 10 sensors for all the 1566 samples

0	-1 "19/07/2008 12:32:00"
1	1 "19/07/2008 13:17:00"
2	-1 "19/07/2008 14:43:00"
3	-1 "19/07/2008 15:22:00"
4	-1 "19/07/2008 17:53:00"
...	...
1561	-1 "16/10/2008 15:13:00"
1562	-1 "16/10/2008 20:49:00"
1563	-1 "17/10/2008 05:26:00"
1564	-1 "17/10/2008 06:01:00"
1565	-1 "17/10/2008 06:07:00"

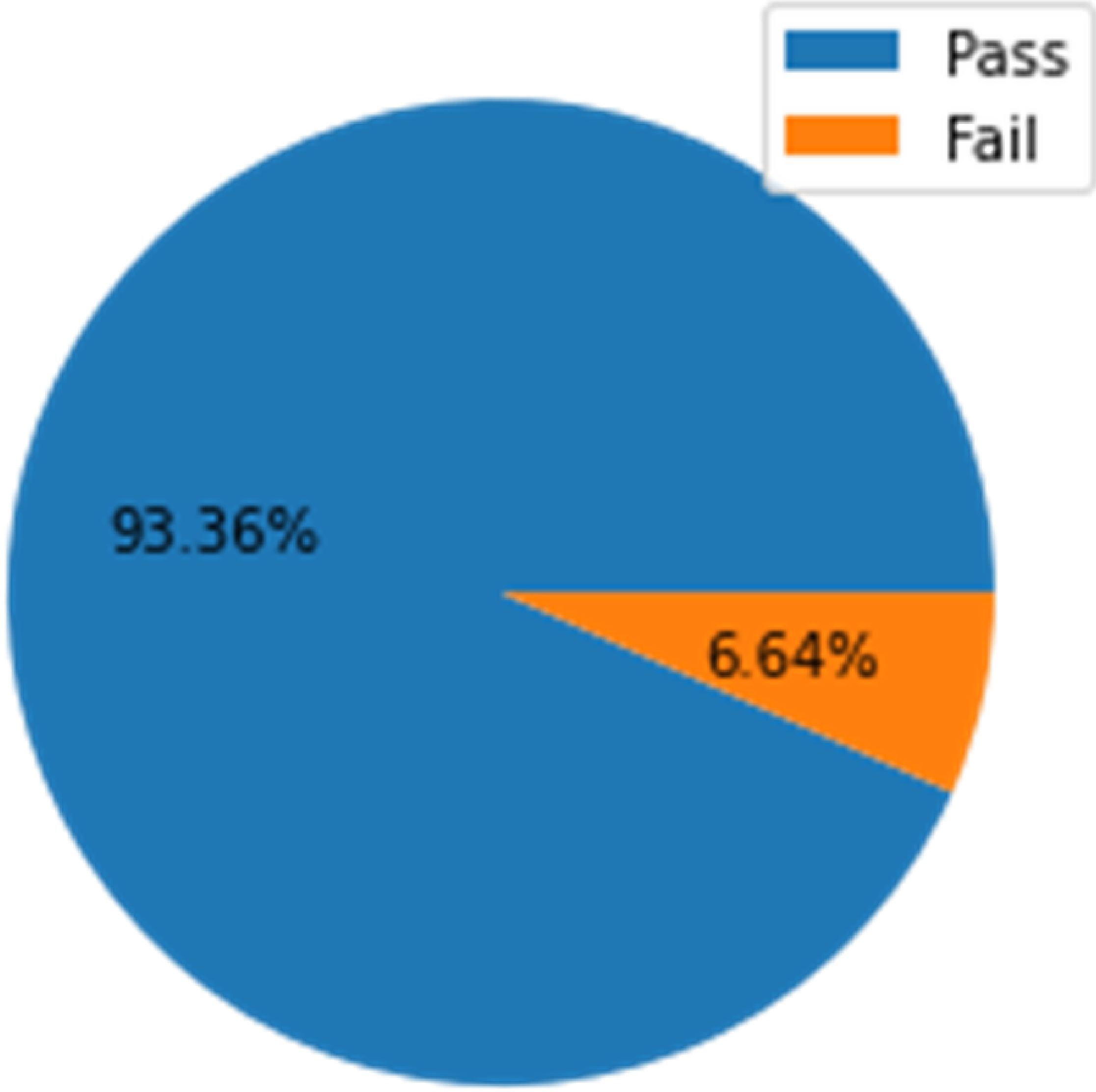
Pass (-1) and Fail (1) Yield Data with timestamps for the 1566 samples

Indian Institute of Technology
Bombay

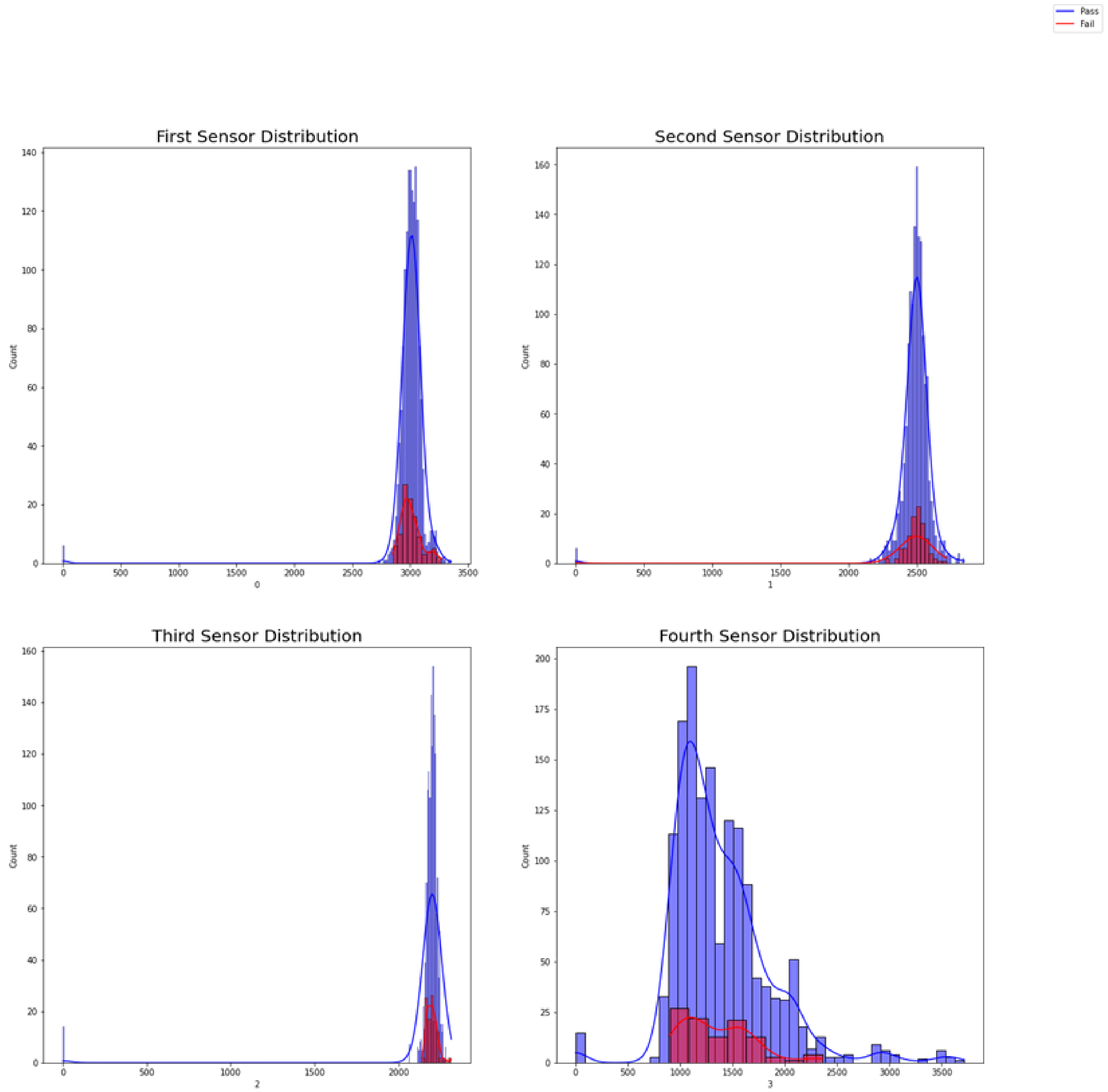
CORRELATION HEATMAP



Violet (dark) regions do appear in the heatmap implying presence of **correlated features**

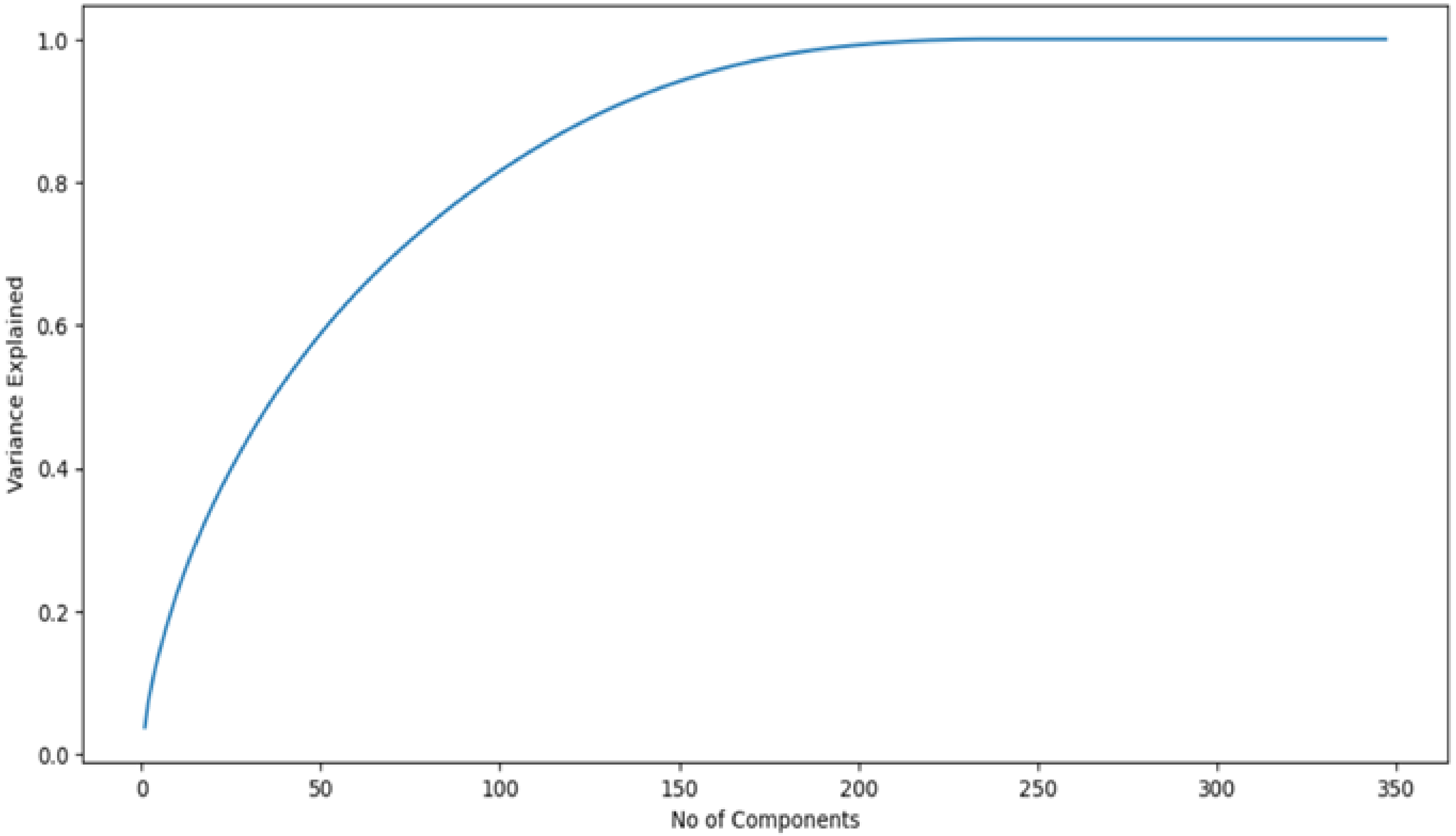


Fail percentage is just 6.65% and pass percentage is 93.35% which clearly indicates a high class imbalance



Class-wise data distribution from first 4 sensors

Feature Reduction



- Removed 28 features which had more than half data as null values
- Presence of correlated features found in correlation matrix / heatmap
- Removed 215 variables with correlation between a pair of features > 0.9
- Employed PCA ---> Top 200 components explain almost all of the variance
- Transformed data has 200 features down from initial 590 sensor variables

Figure : Variance Explained vs No of Components in PCA

FITTING MODELS WITHOUT PCA

Model	Accuracy	F1 Score	Yield %
Logistic Regression	0.80	0.15	84.04
XGBoost	0.91	0.16	96.38
RandomForest	0.93	-	100.00
K Neighbour	0.33	0.155	27.45
SVM	0.92	0.09	97.85

Actual Yield = 92.98%

ML Methodology:

Applied 5 ML (classification) models

- Example based
  - K-Nearest Neighbour

    class sklearn.neighbors.KNeighborsClassifier
- Decision boundary based
  - Logistic Regression

    class sklearn.linear_model.LogisticRegression
  - Support Vector Classification

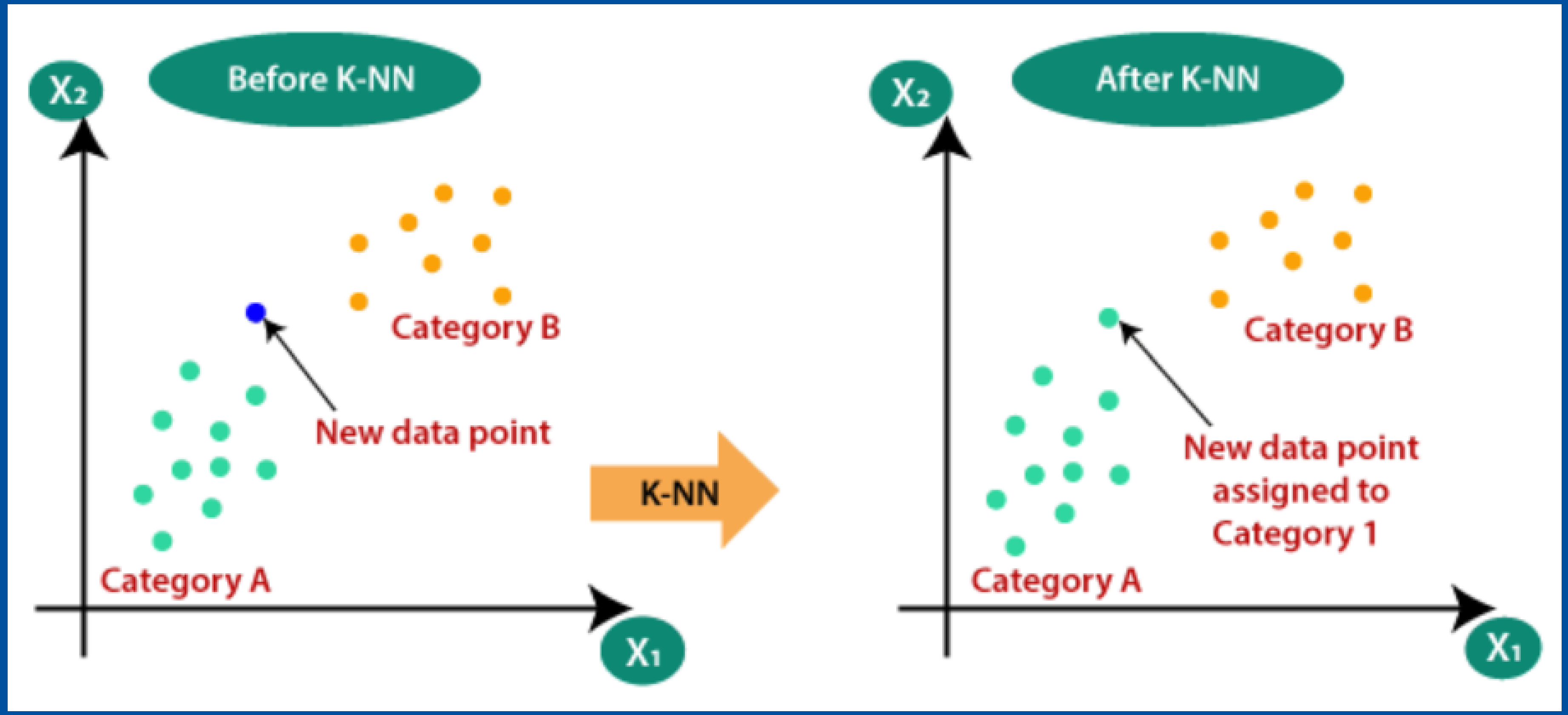
    class sklearn.svm.SVC
- Tree (decision rule) ensemble based
  - Random Forest (Bagging)

    class sklearn.ensemble.RandomForestClassifier
  - Extreme Gradient Boosting

    class xgboost.sklearn.XGBClassifier

Kept the default parameters and computed the accuracy, confusion matrix and f1 score

K-Nearest Neighbour (KNN)



Ex: How a test data point is classified by KNN Ref

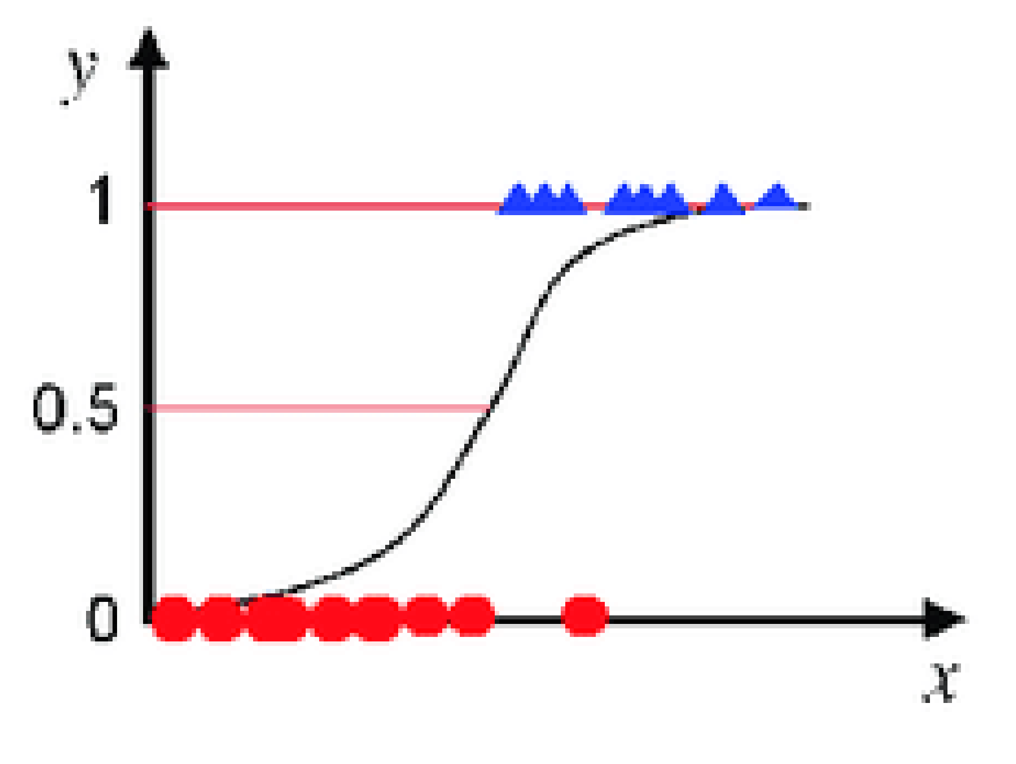
Default Parameters:

1. n_neighbours = 5
2. weights = ‘uniform’
3. algorithm = ‘auto’
4. leaf_size = 30
5. p = 2
6. metric = ‘minkowski’

Algorithm:

1. Choose k
2. Compute distance of test point from all the training data points
3. Choose k points with least distances as the neighbours
4. Assign the majority class

Logistic Regression



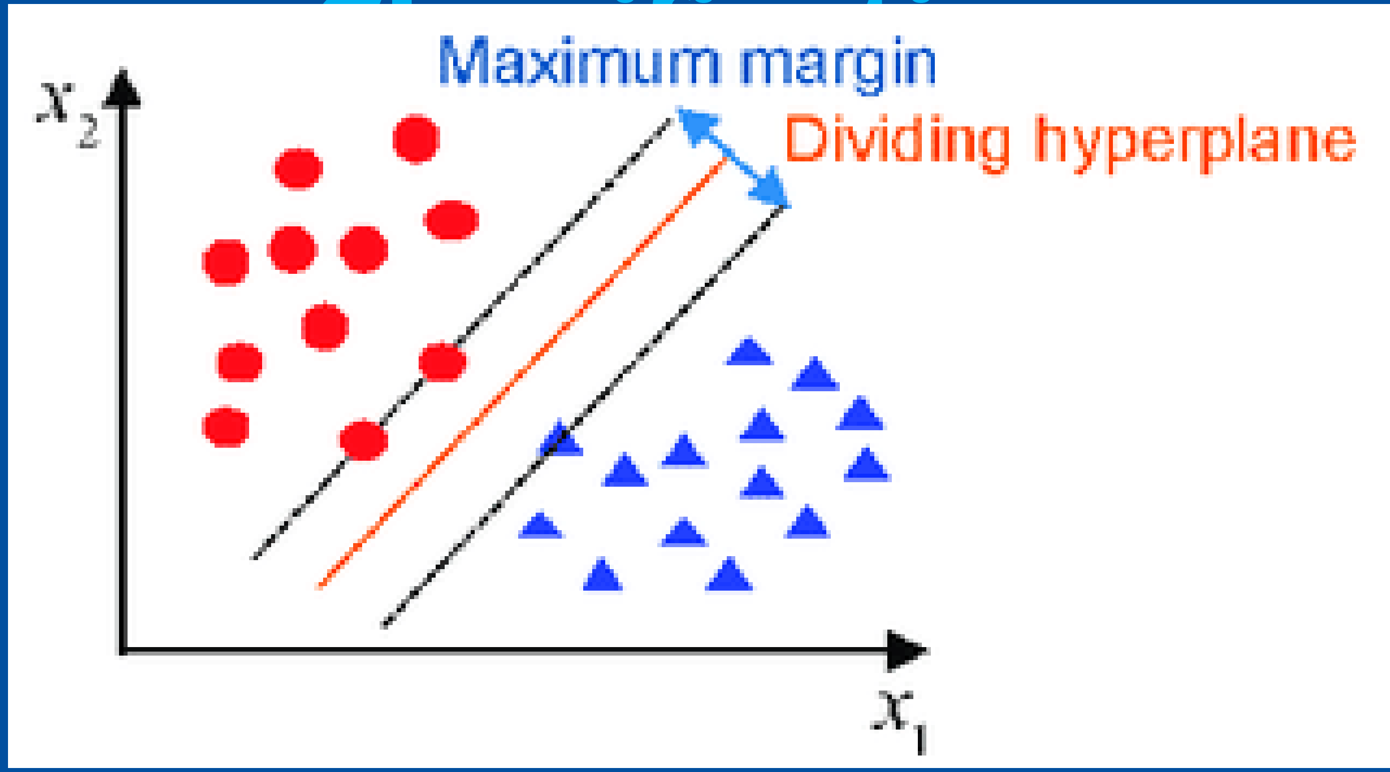
Use of logistic regression to classify Ref

$$P = \frac{e^{a+bX}}{1+e^{a+bX}}$$

Default Parameters:

1. solver = ‘lbfgs’
2. penalty = ‘l2’
3. C = 1

Support Vector



Max margin hyperplane obtained using SVC Ref

$$\min \frac{1}{2}\|w\|^2 + C\sum_i \xi_i$$

Default Parameters:

1. kernel = ‘rbf’
2. gamma = ‘scale’
3. C = 1
4. degree = 3

Random Forest Model (Decision Trees Ensemble)

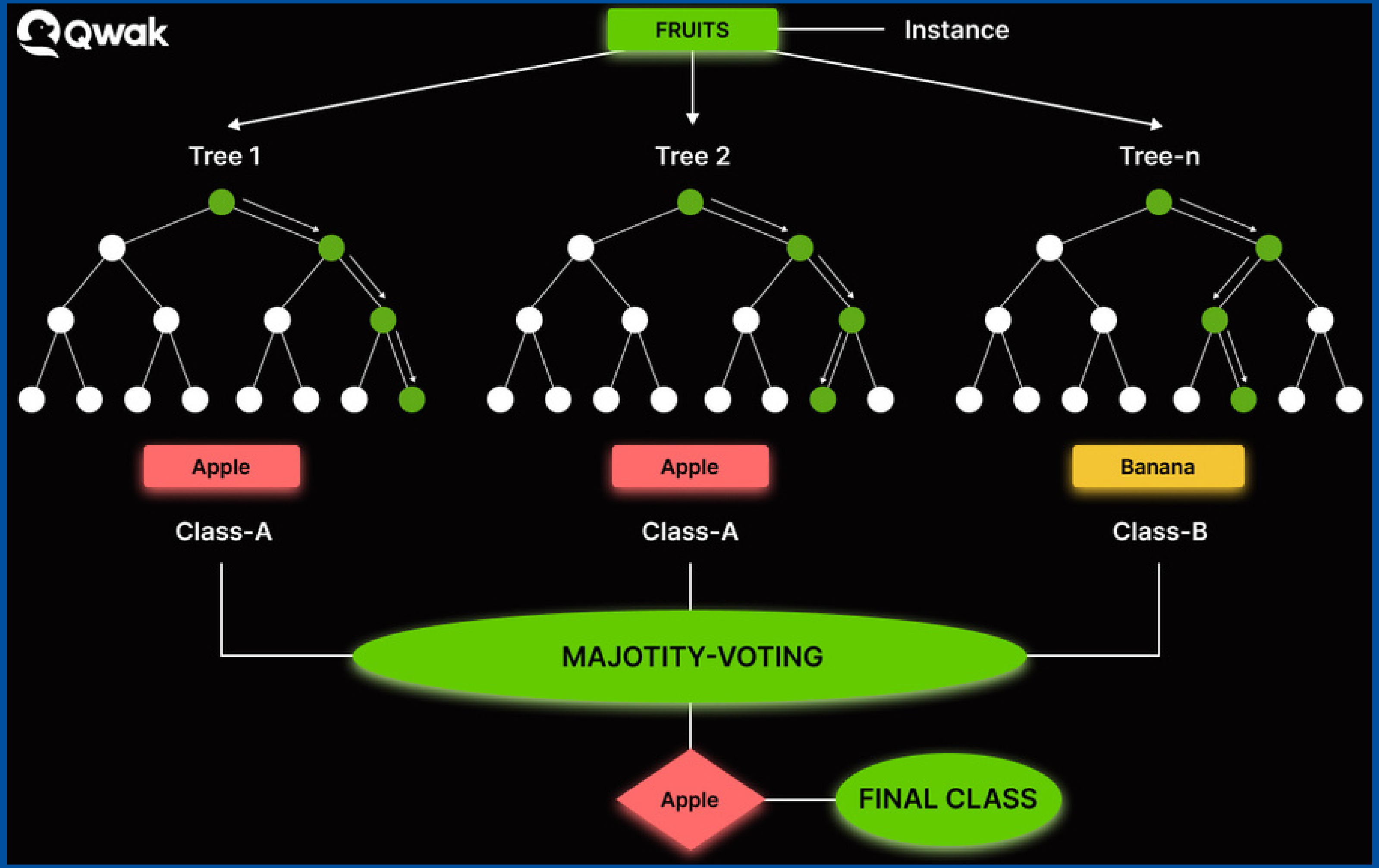


Illustration of a random forest Ref

Default Parameters:

1. n_estimators = 100
2. max_depth = None
3. max_features = ‘sqrt’
4. min_samples_split = 2
5. min_samples_leaf = 1

Extreme Gradient Boosting (XGBoost)

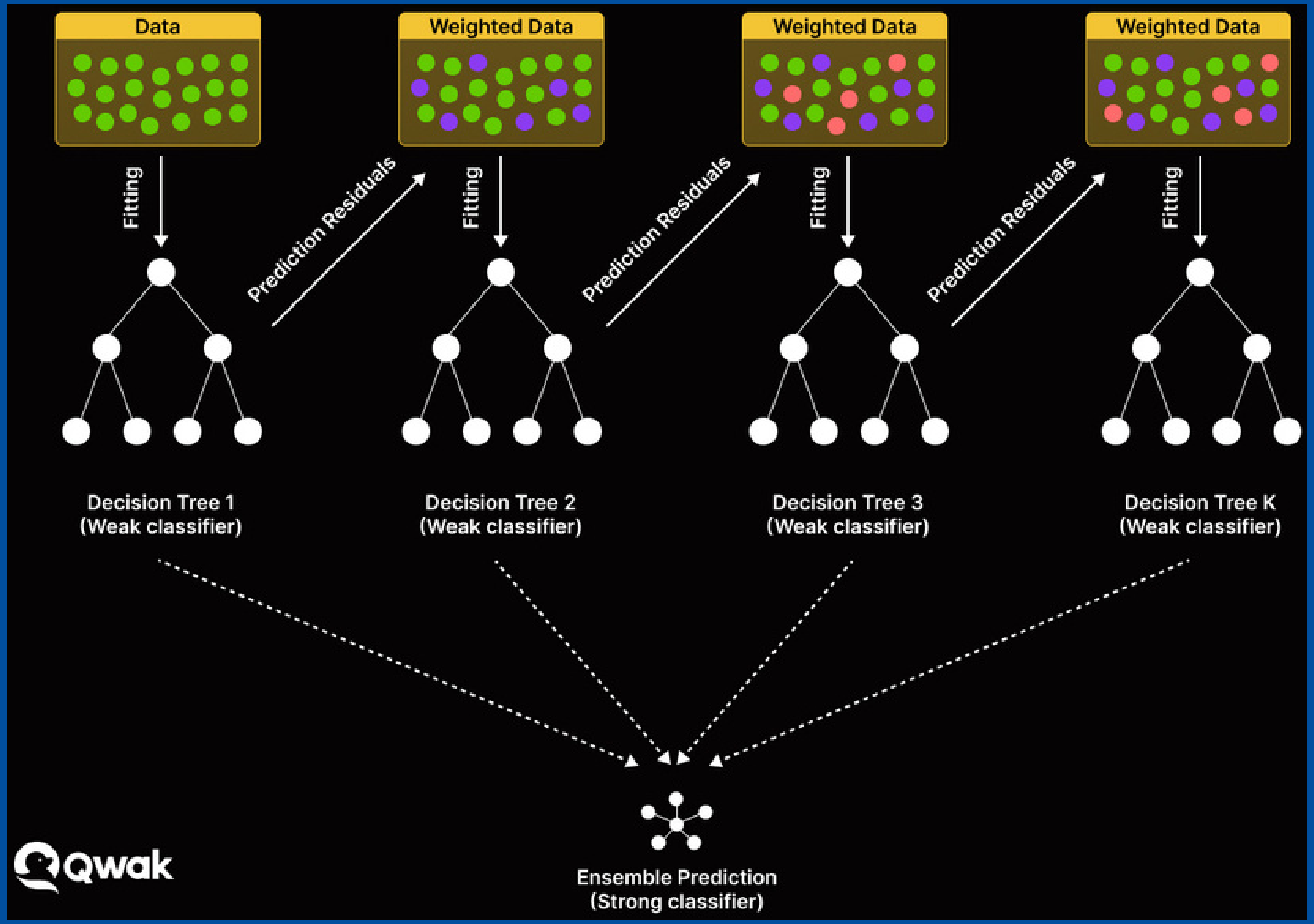


Illustration of a gradient boosting ensemble Ref

Default Parameters:

1. n_estimators = 100
2. max_depth = 6
3. learning_rate = 0.3
4. subsample = 1, colsample_bytree = 1
5. alpha = 0, lambda = 1, gamma = 1

Interpretation of results

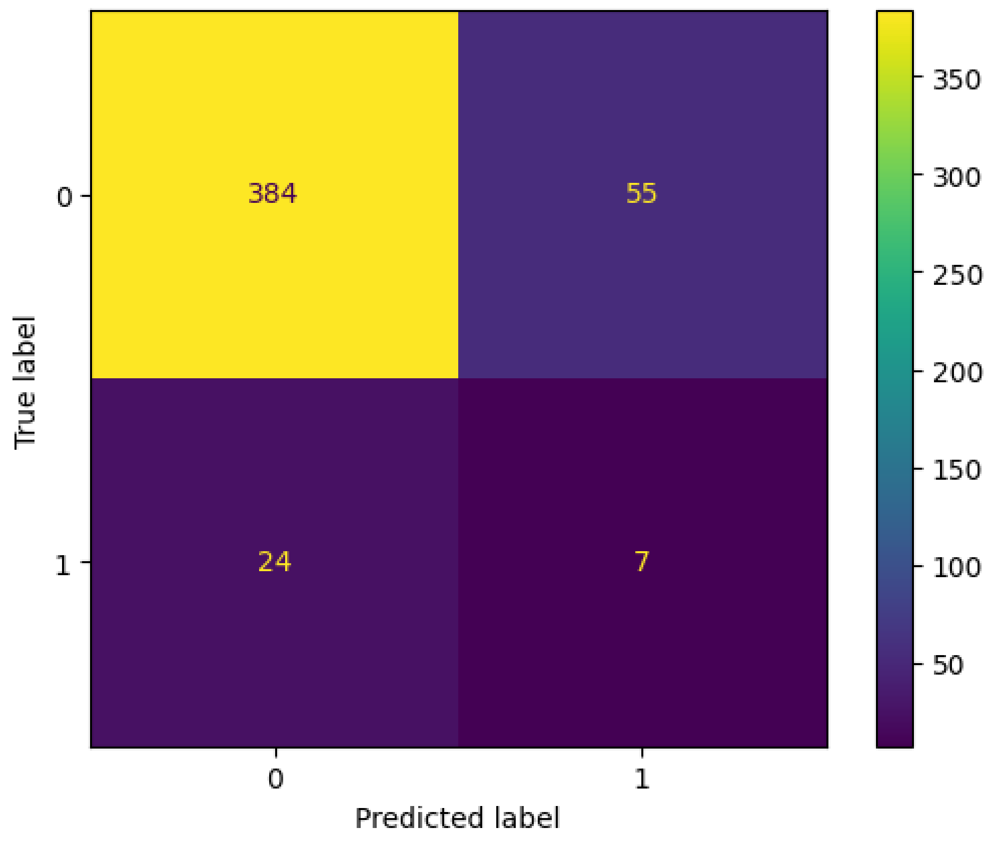
- An extremely high accuracy for Random Forest – Good or Overfitting?
- A low accuracy and F1 score for KNN – needs thorough feature selection
- KNN yielded high amount of false negatives – penalizing false negatives
- SVM longest runtime, gives exhaustive results
- SVM accuracy higher than LR as it tries to find the best hyperplane (max margin classifier) compared to LR which yields any possible hyperplane
- F1 score highest for tree based models (XGBoost and Random Forest)

FITTING MODELS WITH PCA

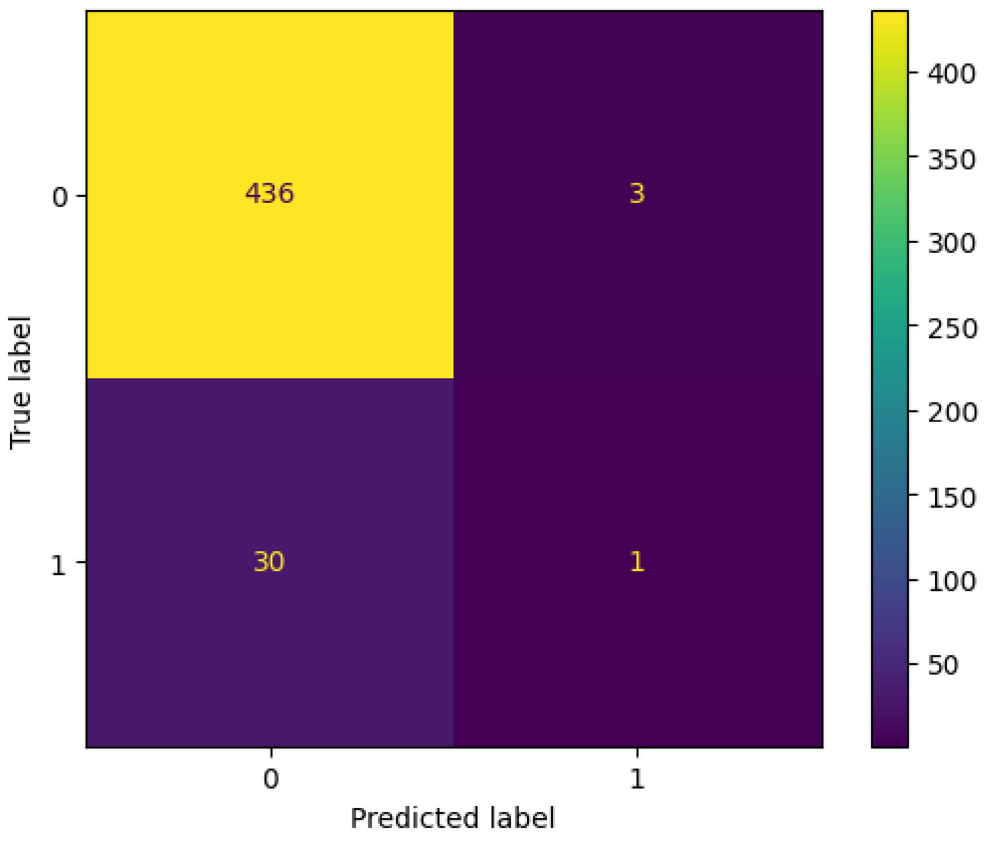
Model	Precision	Recall	F1 Score	Accuracy	Yield %
Logistic Regression	0.94	0.87	0.91/0.53MA	0.83	86.80
XGBoost	0.94	0.99	0.96/0.51MA	0.93	99.14
RandomForest	0.94	1.00	0.97/0.51MA	0.93	99.57
K Neighbour	0.96	0.28	0.44/0.29MA	0.32	27.659
SVM	0.93	0.99	0.96/0.48MA	0.93	99.1489

Actual Yield = 93.40%

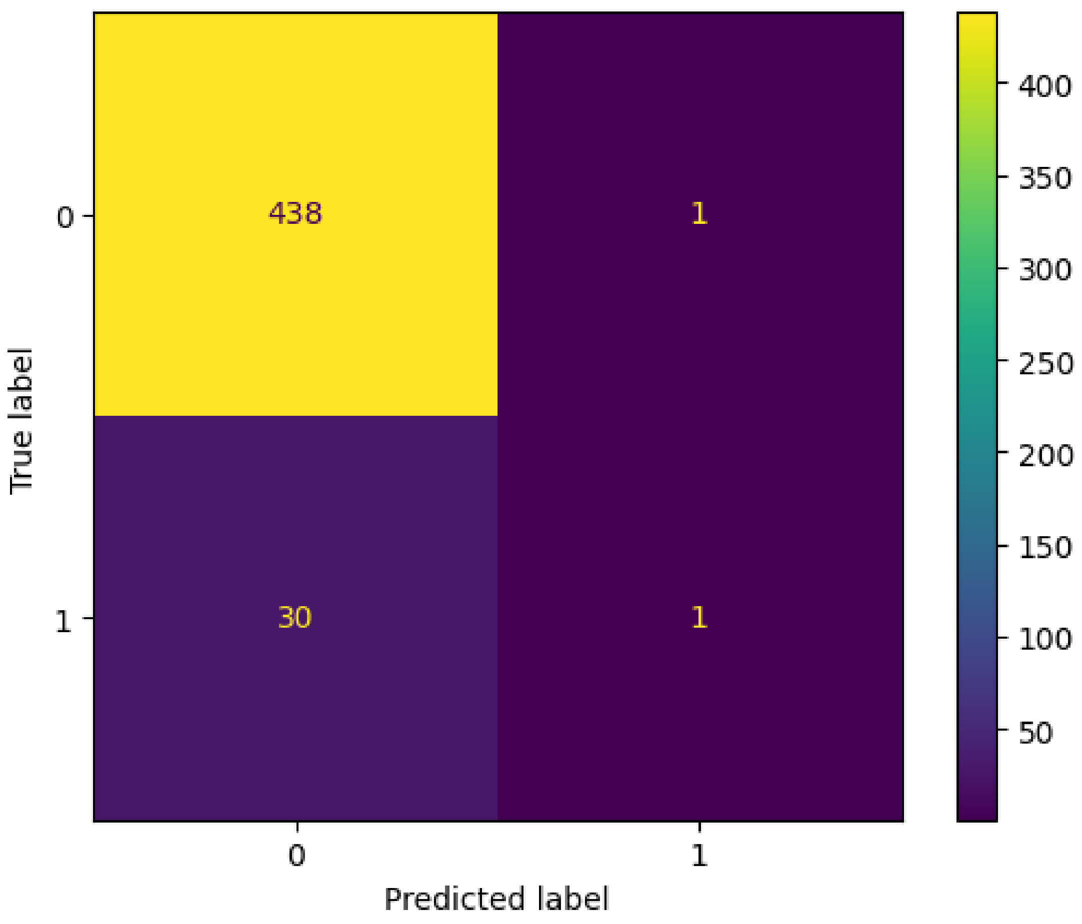
CONFUSION MATRICES



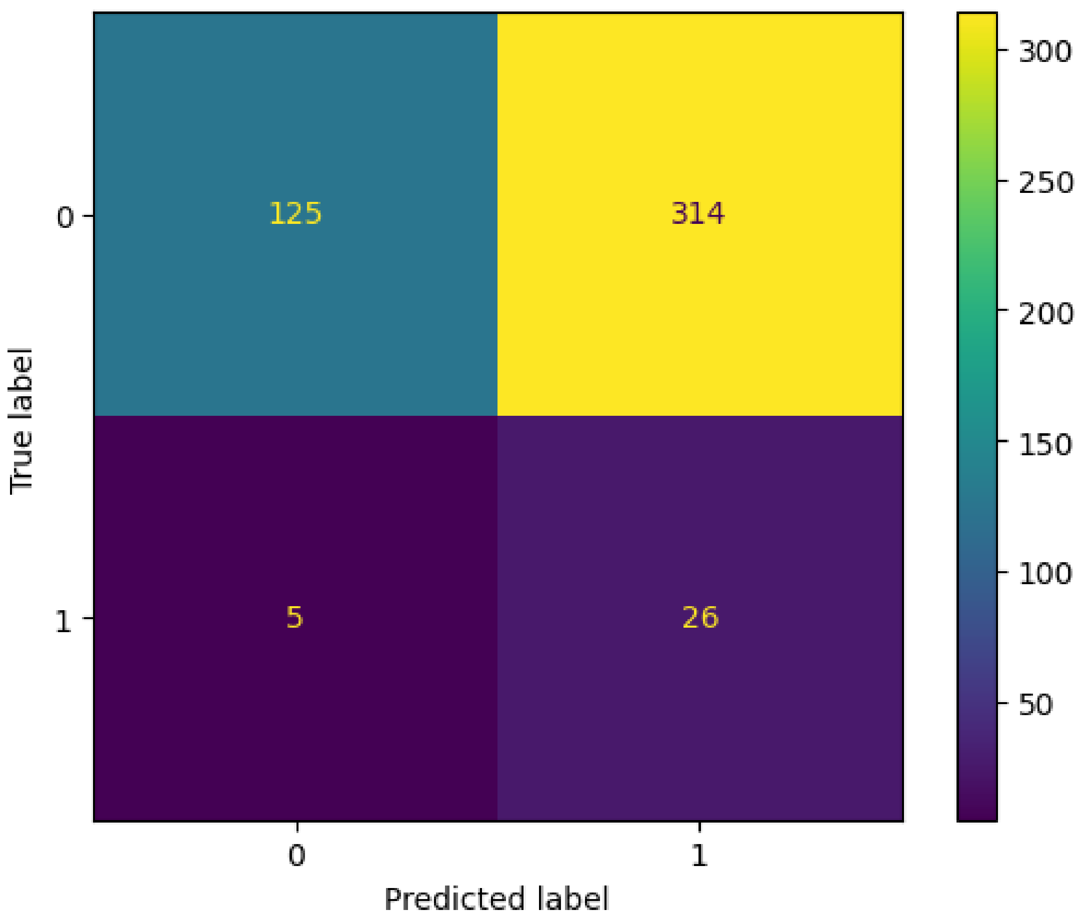
LRM



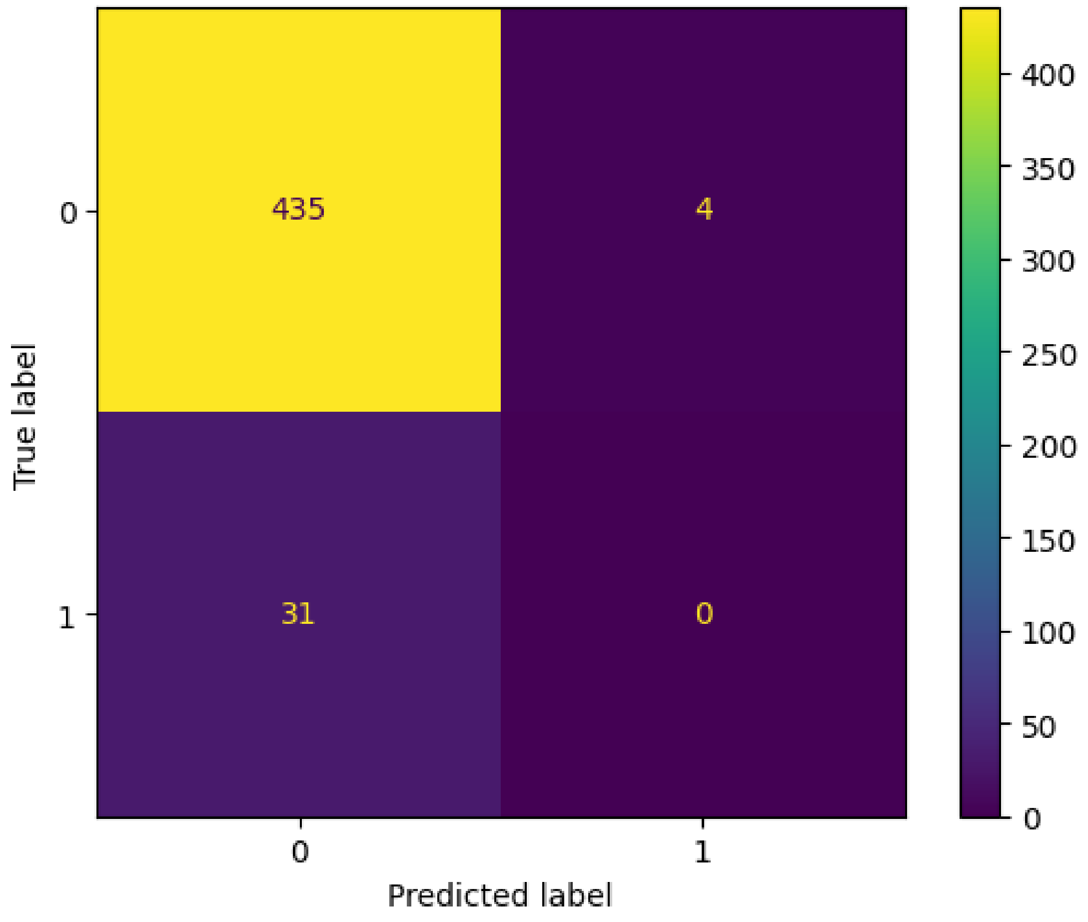
XGB



RFM



KNN



SVM

HYPER PARAMETERTUNING USING GRID SEARCH AND RANDOM SEARCH WITH 4-FOLD CV SCORED ON MA F1

Random Search was applied for RF as there are a lot values for the 6 parameters varied and Grid Search was applied for LRM (5 vals for 2 params)

Best Parameters for LR

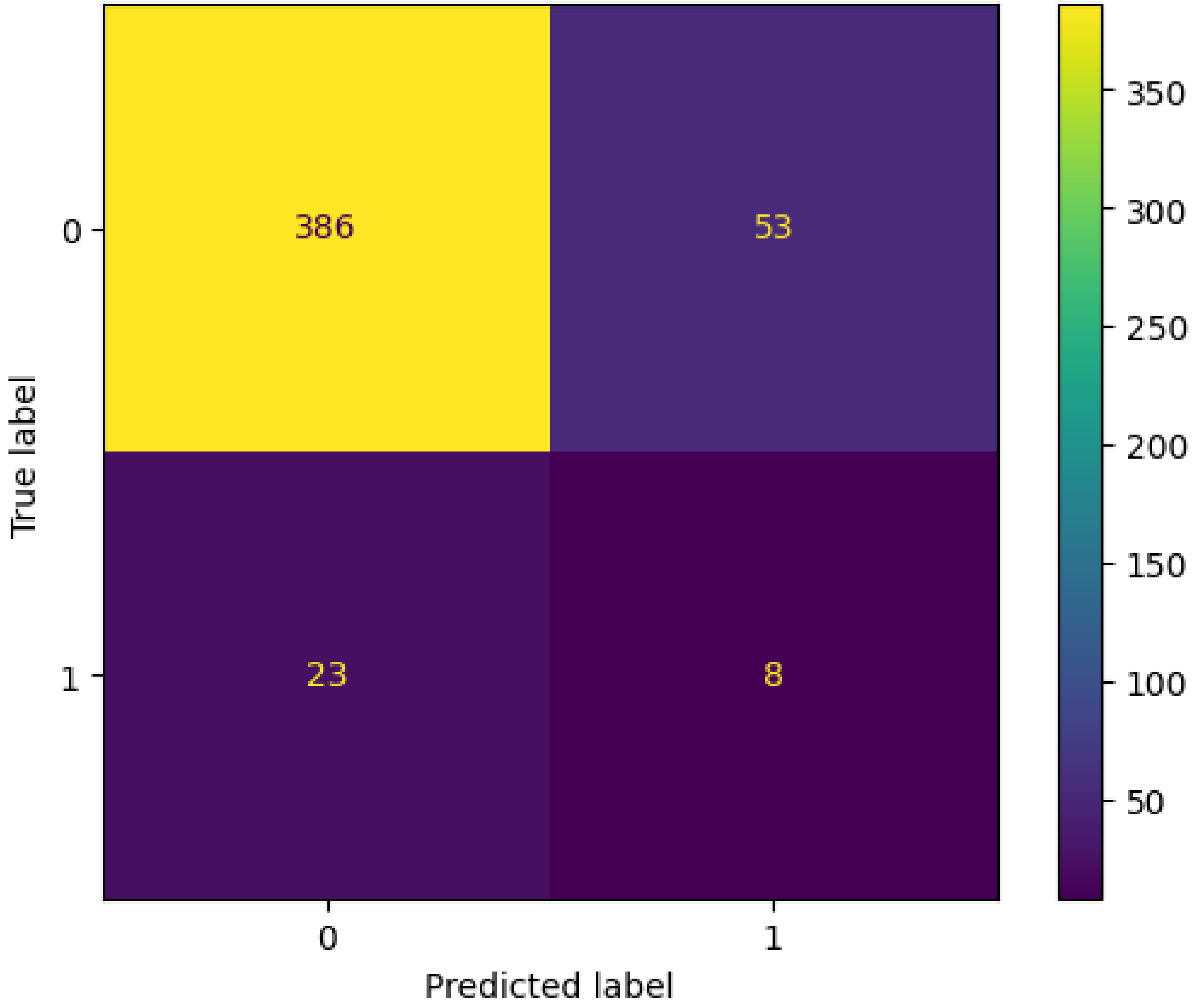
C	10
Solver	Newton_cg

Best Parameters for RF

n_estimators	400
Min_samples_leaf	2
Min_Samples_Split	2

RESULTS AFTER HYPERPARAMETER TUNING

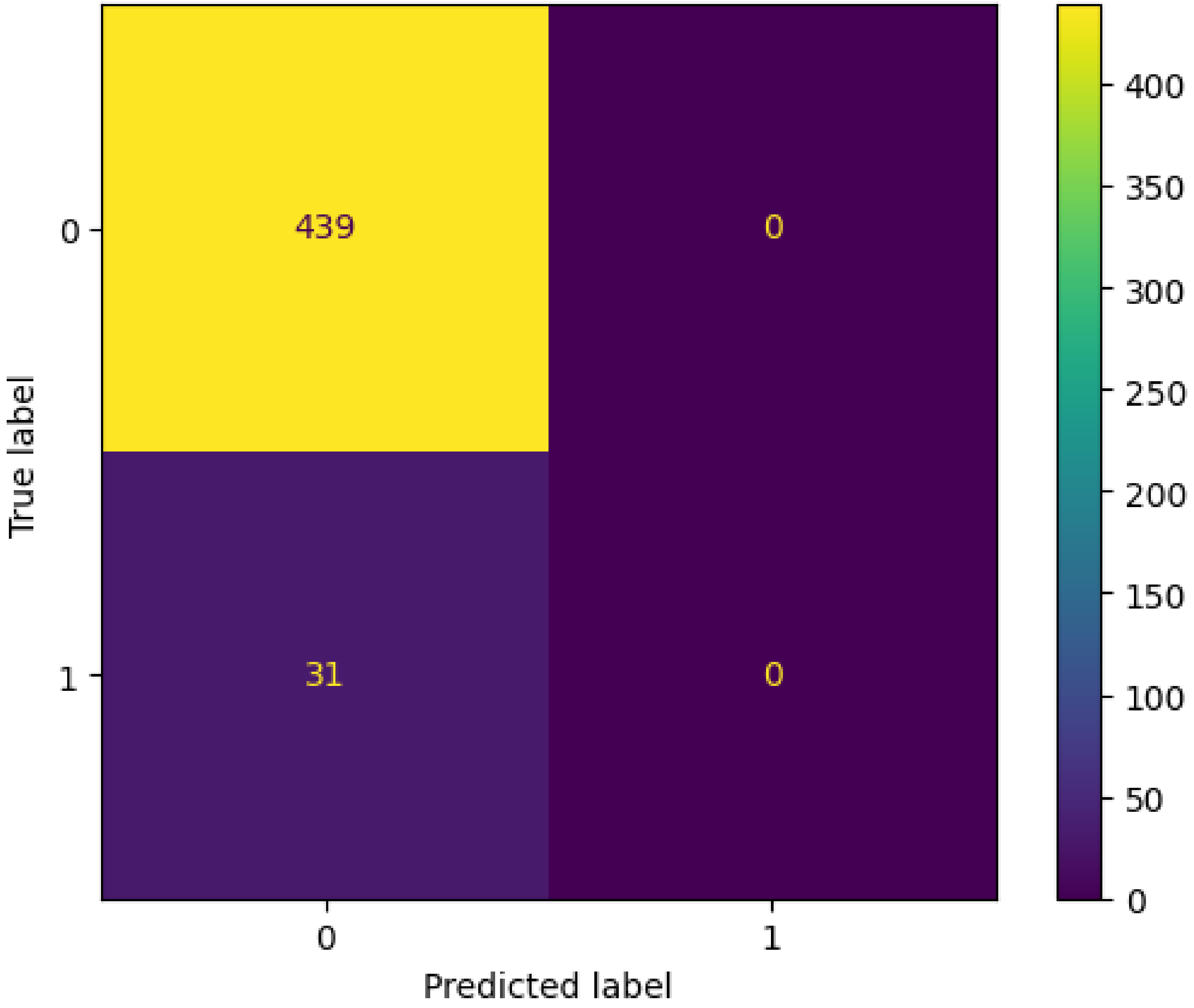
Model	Precision (Pass/Macro-avg)	Recall (Pass/Macro-avg)	F1 Score (Pass/Macro-avg)	Yield %
Logistic Regression	0.94/0.54	0.88/0.57	0.91/0.54	87.02
RandomForest	0.93/0.47	1.00/0.50	0.97/0.48	100.0



<<< best LRM

best RFM >>>

With the highest macro-averaged F1 observed yet Logistic Regression is the best model for our objective



COMPARING MODELS

Model	Precision	Recall	F1 Score	Accuracy	Yield
LRM (without PCA)	0.933	0.846	0.881	0.804	84.04
LRM (with PCA)	0.94	0.87	0.91/0.53 MA	0.83	86.808
LRM (with PCA, with hypertuning)	0.94	0.88	0.91/0.54 MA	0.84	87.021

An increase in the F-1 score and accuracy is observed with the implementation of PCA and after hyperparameter tuning the macro-average F1 & accuracy increased leading to an increased predicted yield which is closer to the actual.

CONCLUSIONS

- The features were reduced from 591 to 347 by using many techniques such as checking for the presence of empty values, correlated columns etc.
- There are 200 principal components which explains ~99% of variance, and are sufficient to predict the pass/fail yield of a process.
- There is no feature/sensor that highly attributes with the output.
- It is important to account for class imbalance, hence we employed oversampling (SMOTE) and used a better metric (macro-averaged F1) for evaluation.
- 5 different ML models of 3 types employed to experiment with data.
- Hyperparameter-tuned logistic regression model using principal component analysis performed the best, evidenced from results. Note: It is also very fast.
- Best LR model: 84% acc. & 0.54 MA F1 on test data with a realistic yield prediction.

THANK YOU