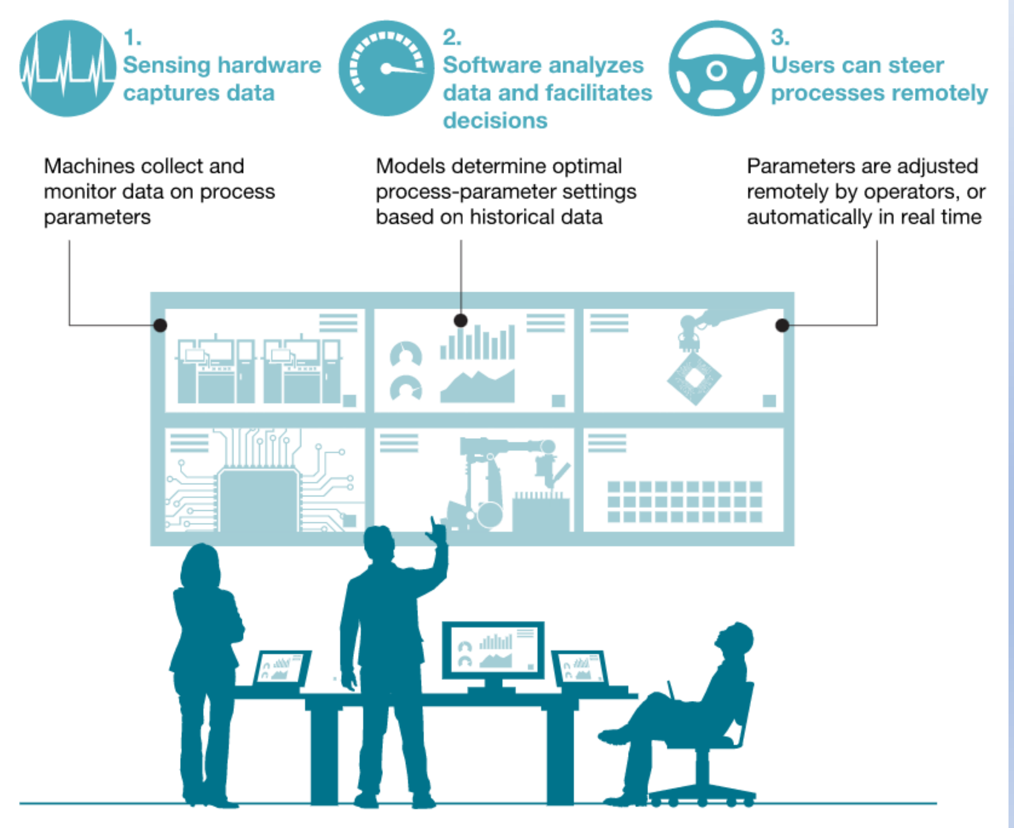
Yield Prediction in Semiconductor Manufacturing

ME 793 Project Stage 2 – 9 April 2023

Team ID: 7

Shivprasad Kathane - 180110076
Tanmay Barhate - 19d170033
Pratham Sehgal - 19d170019



1. Sensing hardware captures data

2. Software analyzes data and facilitates decisions

3. Users can steer processes remotely

Machines collect and monitor data on process parameters

Models determine optimal process-parameter settings based on historical data

Parameters are adjusted remotely by operators, or automatically in real time

Figure: How data can be used to control processes?

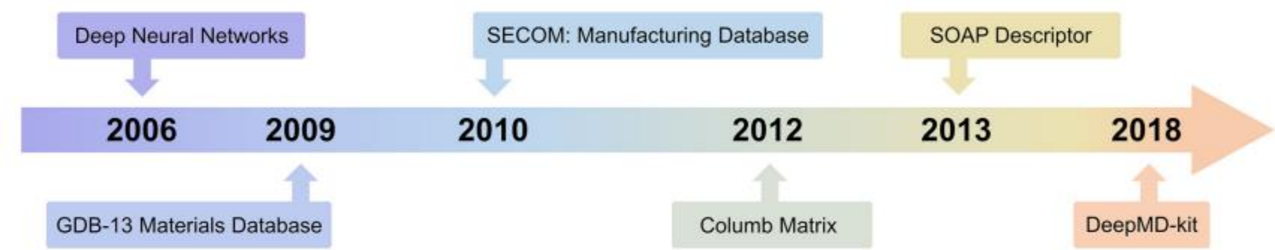
Problem Statement

Yield Prediction in Semiconductor Manufacturing Process via

- **Developing intelligent feature selection/transformation techniques**
- **Modelling the rare occurrence of the defective product using ML/DL.**

Motivation: To enhance the yield by detecting process deviations

- Feature selection identifies key signals for yield excursions in a process.
- Feature analysis and testing can pinpoint vital signals affecting yield, improving efficiency and reducing production costs.



Deep Neural Networks

SECOM: Manufacturing Database

SOAP Descriptor

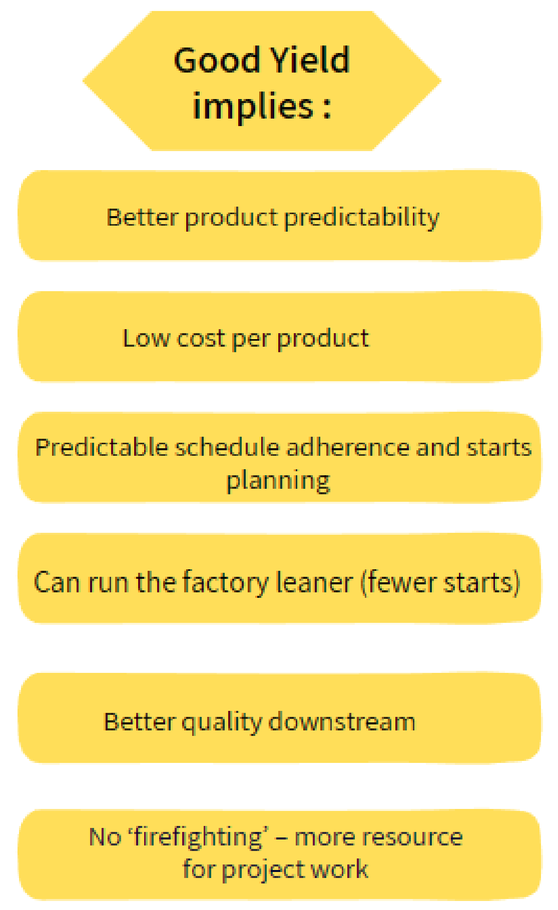
2006 2009 2010 2012 2013 2018

GDB-13 Materials Database

Columb Matrix

DeepMD-kit

- The data and the problem was discovered during the literature review on ML applications for semiconductors Ref: DY Liu et al. "Machine learning for semiconductors". Chip 1 (2022)
- The data is collected from various sensors installed in the production line, such as temperature sensors, voltage sensors, and pressure sensors, among others.



Good Yield implies :

Better product predictability

Low cost per product

Predictable schedule adherence and starts planning

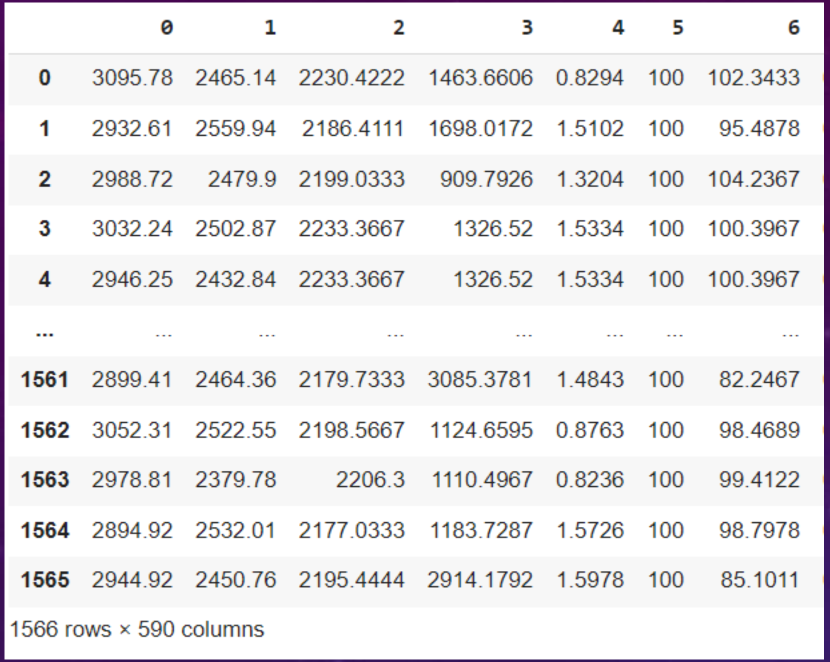
Can run the factory leaner (fewer starts)

Better quality downstream

No 'firefighting' – more resource for project work

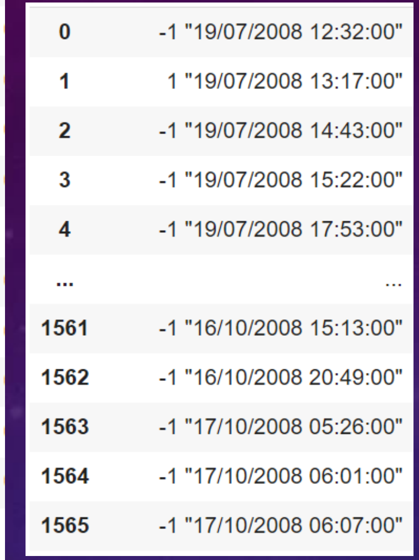
Figure: Advantages of Good Yield

Data



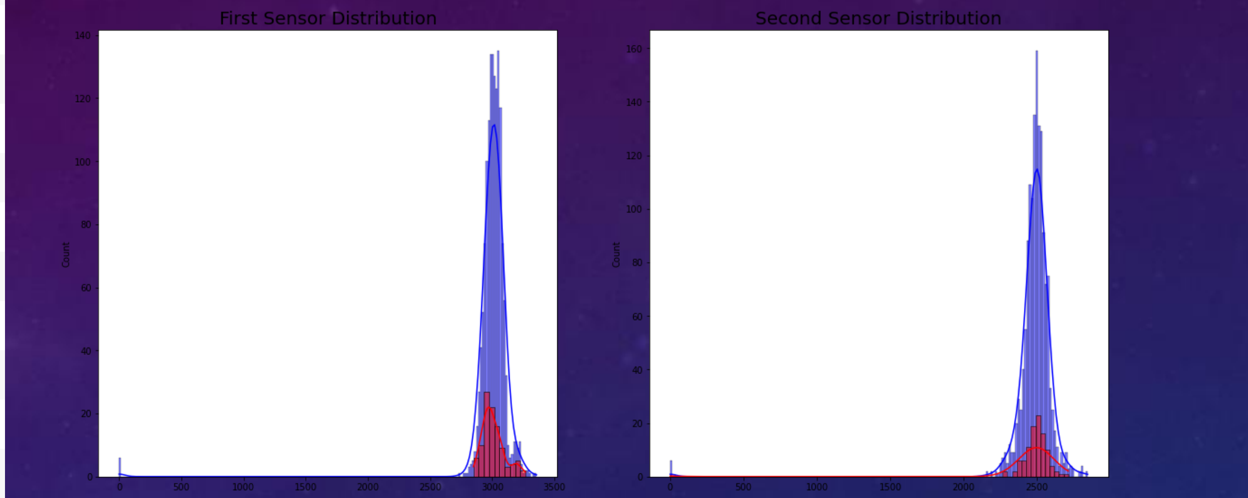
	0	1	2	3	4	5	6
0	3095.78	2465.14	2230.4222	1463.6606	0.8294	100	102.3433
1	2932.61	2559.94	2186.4111	1698.0172	1.5102	100	95.4878
2	2988.72	2479.9	2199.0333	909.7926	1.3204	100	104.2367
3	3032.24	2502.87	2233.3667	1326.52	1.5334	100	100.3967
4	2946.25	2432.84	2233.3667	1326.52	1.5334	100	100.3967
...	...	...	...	...	...	...	...
1561	2899.41	2464.36	2179.7333	3085.3781	1.4843	100	82.2467
1562	3052.31	2522.55	2198.5667	1124.6595	0.8763	100	98.4689
1563	2978.81	2379.78	2206.3	1110.4967	0.8236	100	99.4122
1564	2894.92	2532.01	2177.0333	1183.7287	1.5726	100	98.7978
1565	2944.92	2450.76	2195.4444	2914.1792	1.5978	100	85.1011

1566 rows × 590 columns

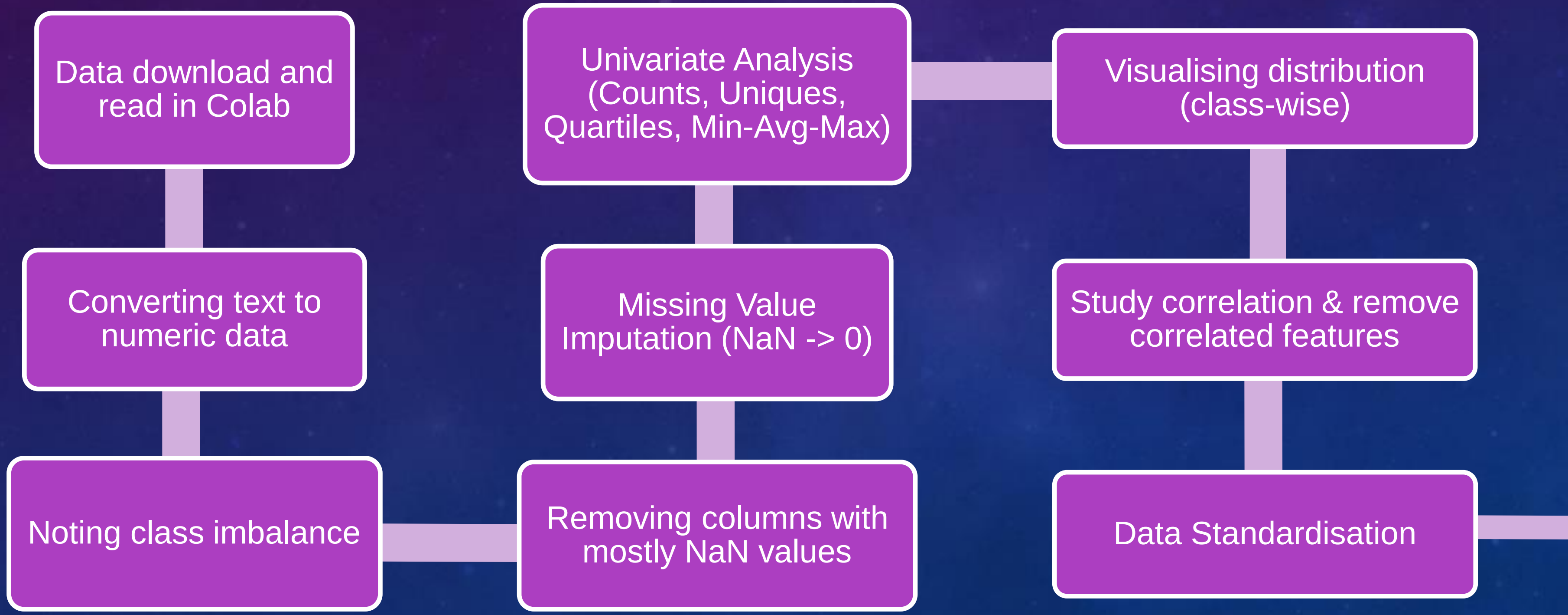


0	-1 "19/07/2008 12:32:00"
1	1 "19/07/2008 13:17:00"
2	-1 "19/07/2008 14:43:00"
3	-1 "19/07/2008 15:22:00"
4	-1 "19/07/2008 17:53:00"
...	...
1561	-1 "16/10/2008 15:13:00"
1562	-1 "16/10/2008 20:49:00"
1563	-1 "17/10/2008 05:26:00"
1564	-1 "17/10/2008 06:01:00"
1565	-1 "17/10/2008 06:07:00"

Distributions of first 2 sensor measurements



Methodology adopted before applying ML



Data download and read in Colab

Converting text to numeric data

Noting class imbalance

Removing columns with mostly NaN values

Missing Value Imputation (NaN -> 0)

Univariate Analysis (Counts, Uniques, Quartiles, Min-Avg-Max)

Visualising distribution (class-wise)

Study correlation & remove correlated features

Data Standardisation

Employ PCA for feature reduction

SMOTE: Normalise class imbalance

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
- PCA is preferred for its simplicity, efficiency, and ease of interpretation
- Transformed data has 200 features down from initial 590 sensor variables

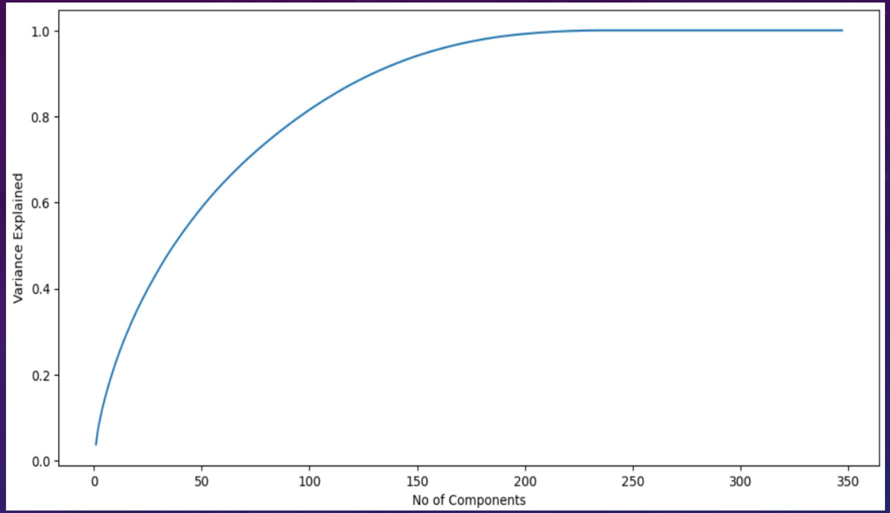


Figure : Curve explaining Variance with respect to Components

Tackling Class-Imbalance

SMOTE (Synthetic Minority Over-Sampling Technique) is used to remove the problem of imbalanced data.

The advantage of using SMOTE is that it can improve the overall performance of the machine learning model on the minority class by providing it with more representative and diverse data.

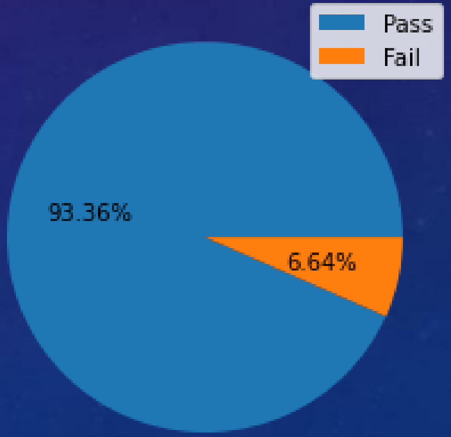


Figure : Pie Chart representing class imbalance

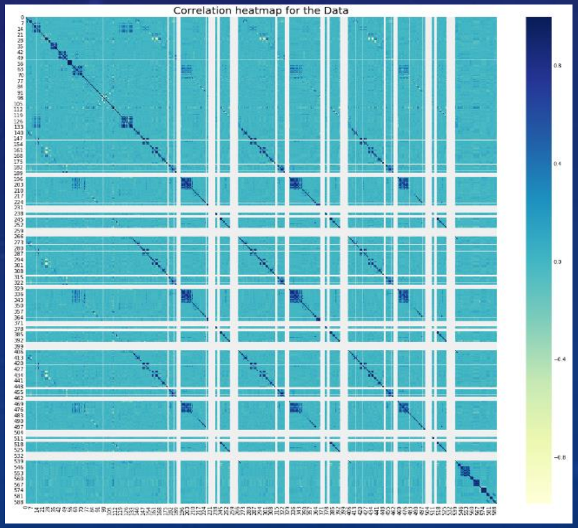


Figure : Heatmap (dark regions indicate correlation)

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)

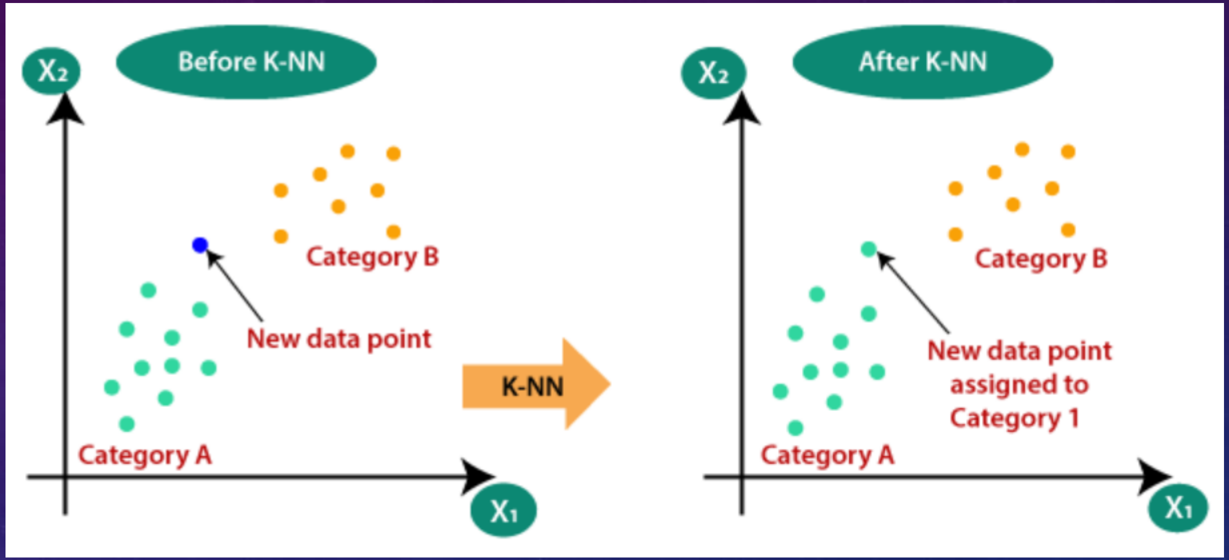
  class sklearn.ensemble.RandomForestClassifier

  - 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

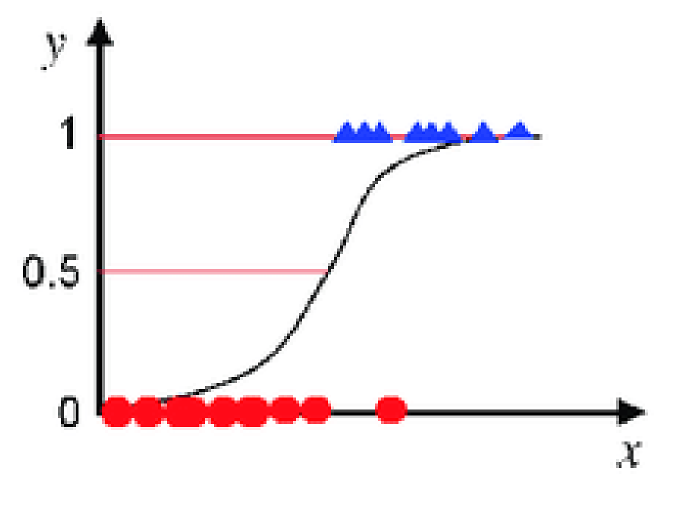
Default Parameters:

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



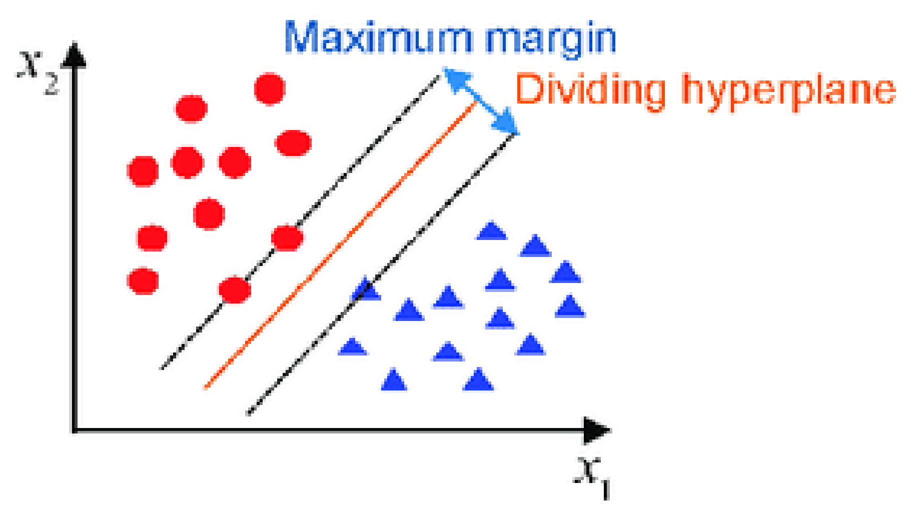
Use of logistic regression to classify Ref

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

Default Parameters:

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

Support Vector Classification



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)

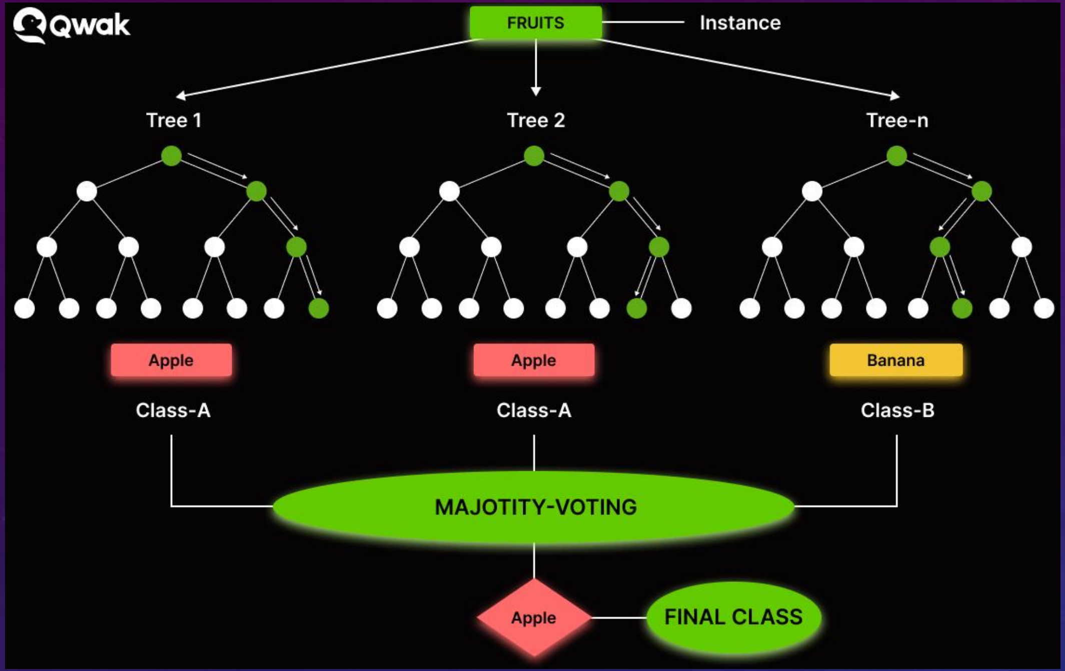


Illustration of a random forest Ref

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)

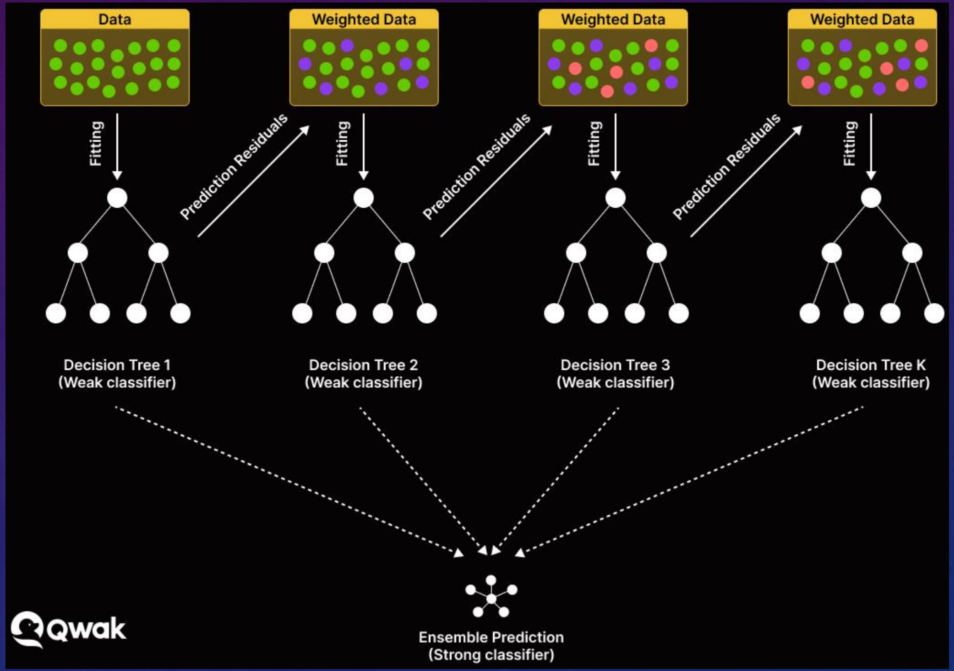
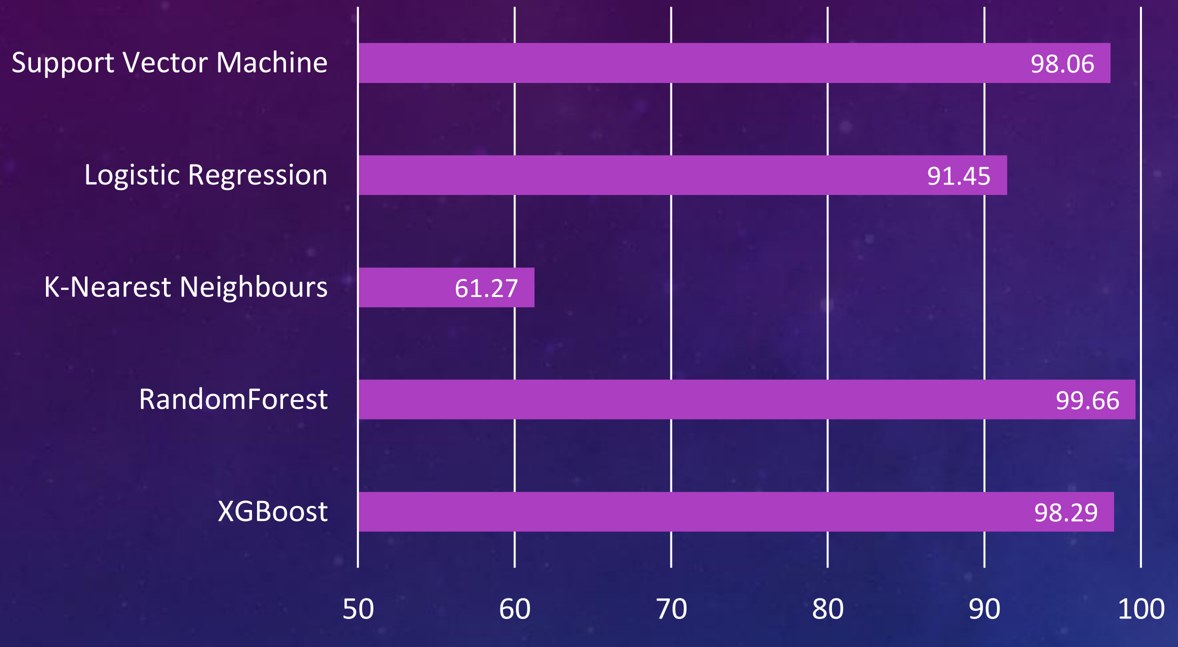

Data
Weighted Data
Weighted Data
Weighted Data
Fitting
Prediction Residuals
Fitting
Prediction Residuals
Fitting
Prediction Residuals
Fitting
Decision Tree 1 (Weak classifier)
Decision Tree 2 (Weak classifier)
Decision Tree 3 (Weak classifier)
Decision Tree K (Weak classifier)
Qwak
Ensemble Prediction (Strong classifier)

Illustration of a gradient boosting ensemble Ref

Default Parameters:

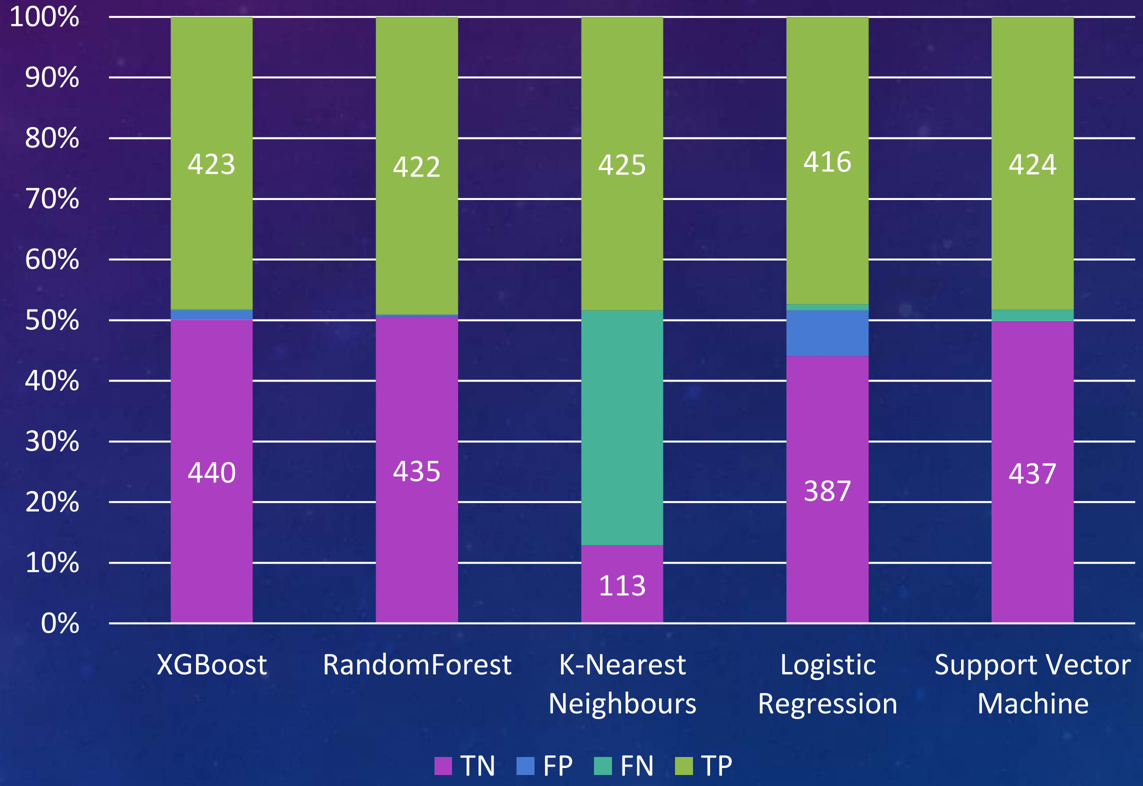
1. n_estimators = 100
2. max_depth = 6
3. learning_rate = 0.3
4. subsample = 1, colsample_bytree = 1
5. alpha = 0, lambda = 1, gamma = 1

ML Model Results



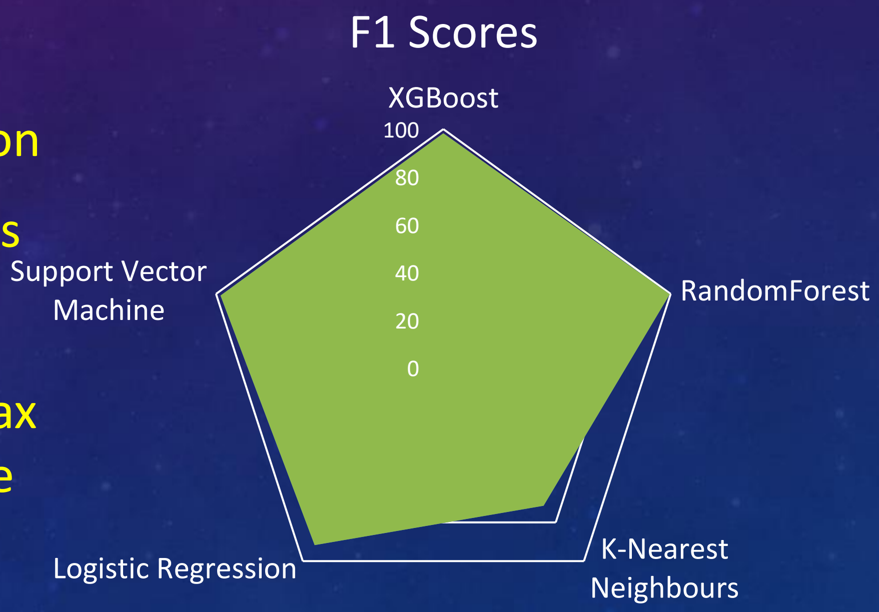
Accuracy distribution (%)

Confusion Matrix Data



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)



Possible Improvements

- Scaling to get a good picture of k-NN classification
- Trying out other methods for feature selection
- Hyperparameter tuning with cross-validation to improve ML models
- Implementing regularization to avoid overfitting
- Consider transfer learning from literature referred and implement it
- Select the best model / approach, execute and report the results

References:

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)

Conclusions

- Predicting process deviations (which affect semiconductor yield) from data is critical
- Removal of correlated features + PCA to yield better & reduced feature-set
- Oversampling used to equalize class imbalance
- 5 different ML models of 3 types employed to experiment with data
- Random Forest method gave the best accuracy (99.66%) and F1 score
- This was followed by XGBoost (98.29%) and SVM (98.06%)
- Hyperparameter Tuning with Cross-Validation may yield a good generalized model
- Video Link: Link to the Presentation Recording