STEVENS
INSTITUTE OF TECHNOLOGY
1870

Predicting Annual Income of Individual using Classification Techniques

2023S-AAI - 695-WS2

Team 6

Date: 05/10/2023

STEVENS
INSTITUTE OF TECHNOLOGY
1870

Team members

Darsh Patel(20012159)

Janmejay Mohanty(20009315)

Soumen Sikder Shuvo(20011273)

STEVENS
INSTITUTE OF TECHNOLOGY
1870

Flow of presentation:

- Introduction
- Dataset description
- Feature engineering
- Implementation of ML algorithms
- Comparision and conclusion
- Future work
- Reference

Dataset description and task

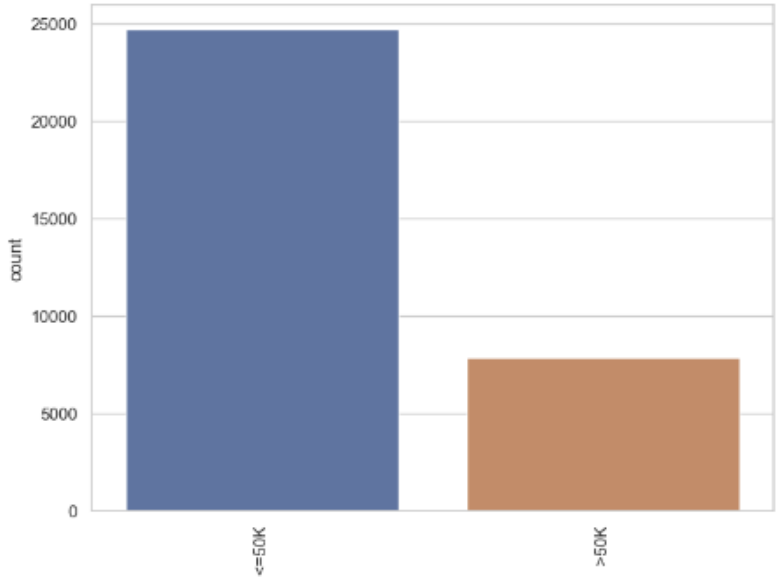
Adult dataset

- https://archive.ics.uci.edu/ml/datasets/adult
- 1994 United States census data
- 48842 rows * 15 columns
- 8 Categorical features and 6 continuous features

Categorical Features	Continuous Features
workclass	age
education	fnlwgt
marital-status	education-num
occupation	capital-gain
realtionship	capital-loss
race	hour-per-week
gender	
native-country	

Task

- Our Task is to determine whether a person makes our $50k given the attributes



Feature Engineering

Feature Engineering is not any ML Algorithm, rather than it is a data processing technique, where we use these processed data to train established ML models.

- Used to extract important features
- Can reduce the error due to imbalanced dataset
- Can be computationally Expensive

There are many Feature Engineering Techniques:

- Imputation.
- Discretization.
- Categorical Encoding.
- Feature Splitting.
- Handling Outliers.
- Variable Transformations.
- Creating Features.

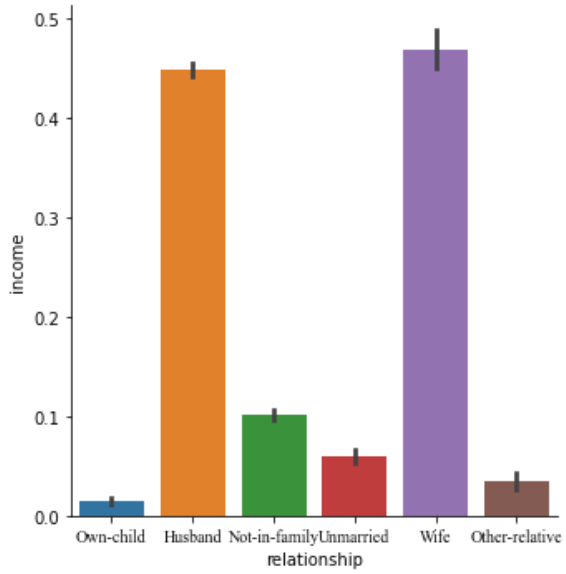
id	color
1	red
2	blue
3	green
4	blue

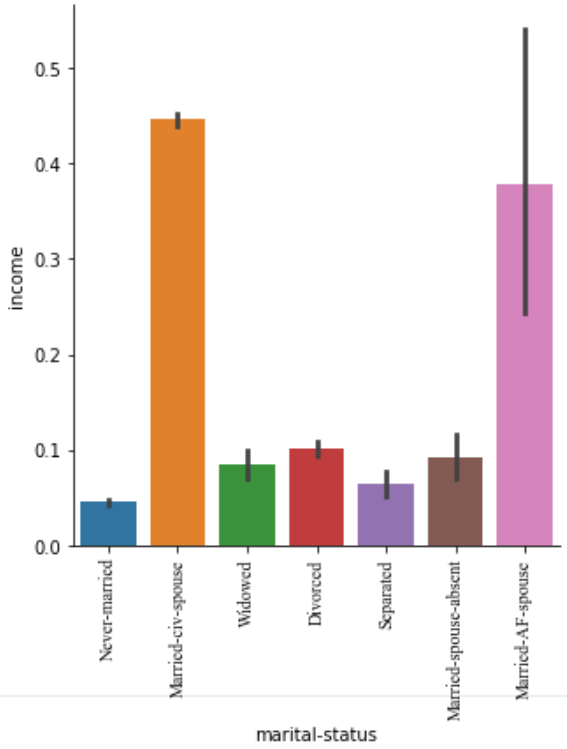
id	color_red	color_blue	color_green
1	1	0	0
2	0	1	0
3	0	0	1
4	0	1	0

Feature Engineering

Categorical Encoding: Used to encode categorical features into numerical values which are usually simpler for an Machine Learning algorithm.
We used One-hot Encoding as Feature Engineering Method. Then used that data in Decision Tree and Random Forest to perform the Classification.

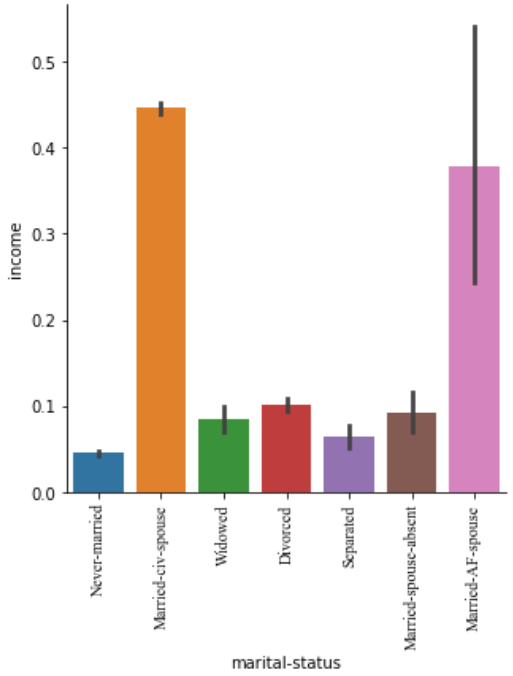
Discretization: Takes a set of values of data and groups the sets of them together in some logical criteria.

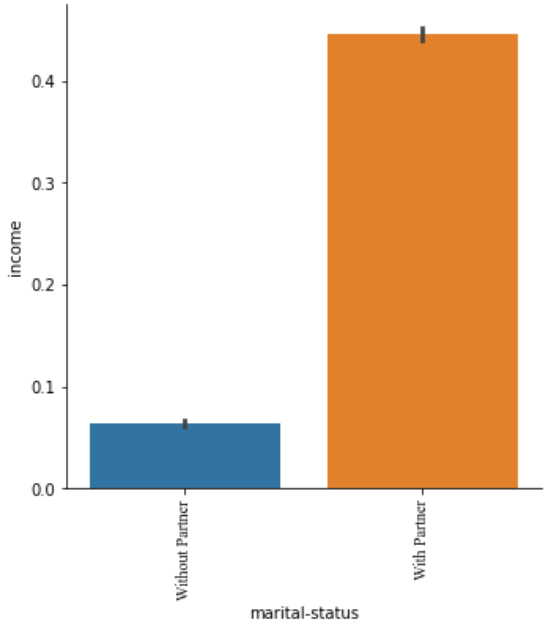




Here we can see some values have very high probability of earning more than 50k, and others have very less. We can merge them to make new features.

Feature Engineering





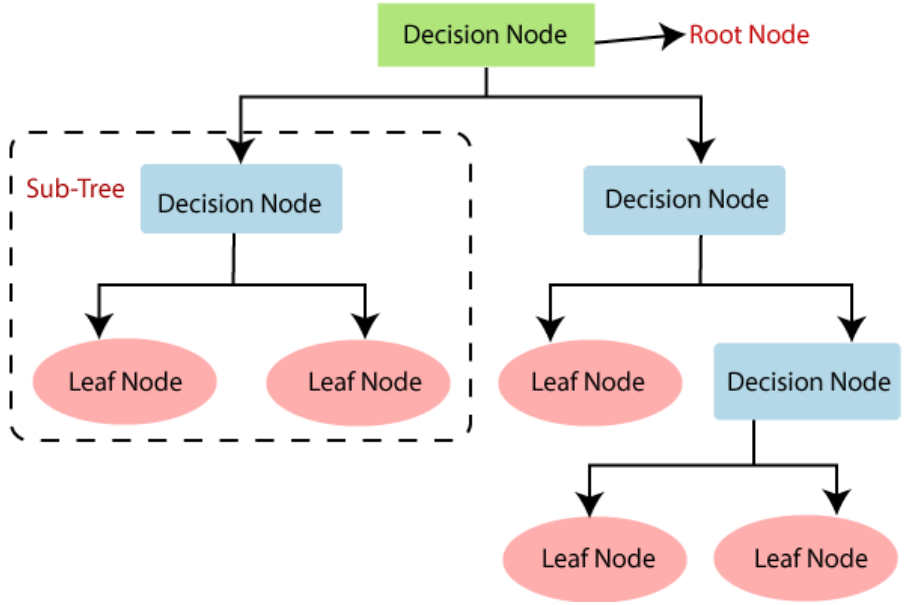
Decision Tree

Here, the data is recursively split into subsets based on the most informative features for predicting the target variable.

- Robust algorithm for binary classification tasks
- Better for Complex Feature Structures
- Can Handle Both Categorical and Continuous Data
- Have the tendency to overfit training data

We used various parameters and got best results at,
max_depth = 9,
min_samples_leaf = 4,
min_samples_split = 2

Accuracy = 85.6%
F-1 Score = 0.687



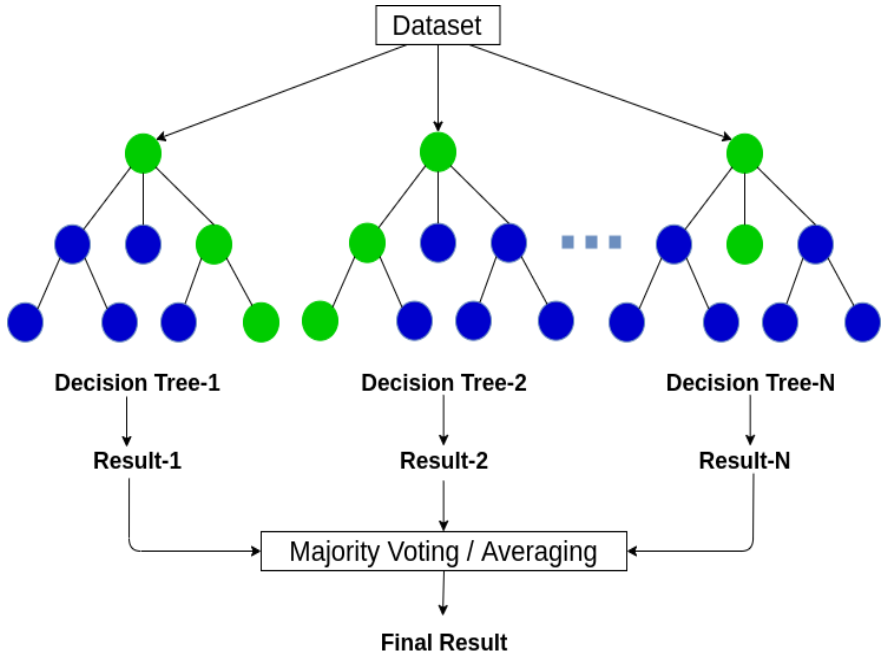
Random Forest

Random Forest is a supervised algorithm that uses ensemble learning method consisting of a multitude of decision trees.

- Each tree in a random forest is trained on a random subset of the training data
- Can automatically identify important features in the data
- Can handle both categorical and continuous data
- Has a lower risk of overfitting compared to a single decision tree

We used Grid Search algorithm of Scikit-learn to search for the best parameters. At, n_estimators = 300 and max_features = 6, we get the best result.

Accuracy = 85.59%
F-1 Score = 0.688

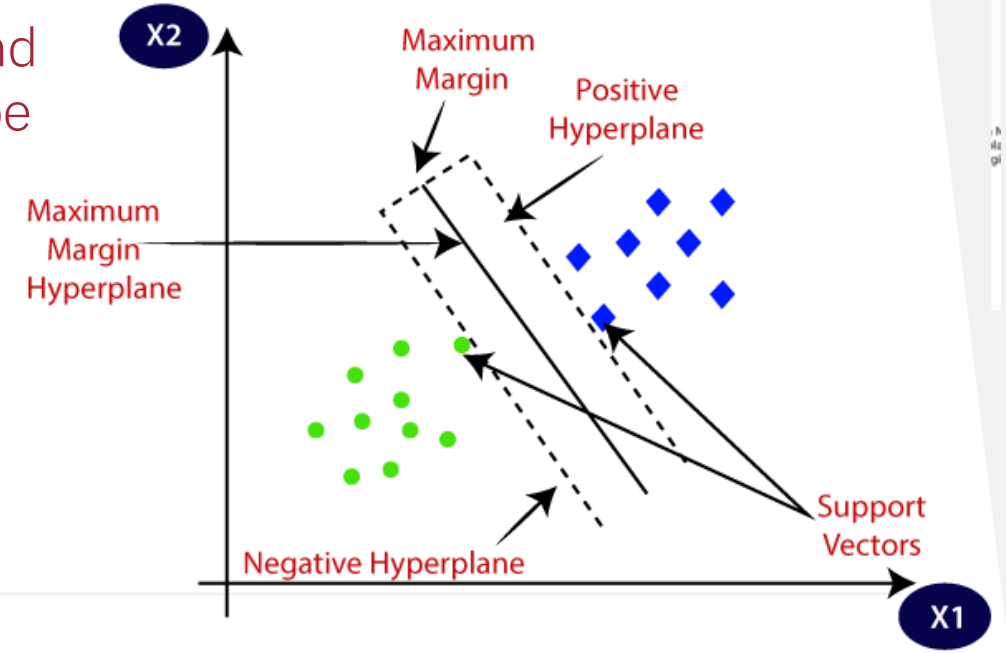


SVM

Support Vector Machine is a supervised learning algorithm that works by finding a hyperplane in a high- dimensional space that best separates the data into different classes.

- Effective with more features
- Highly dependent on Hyperparameters , such as kernel function
- Computationally Expensive

We used SVM using linear and rbf kernels. The accuracy and F-1 score in SVM was not good. The probable reason can be the features are more overlapping and hugely unbalanced Dataset.



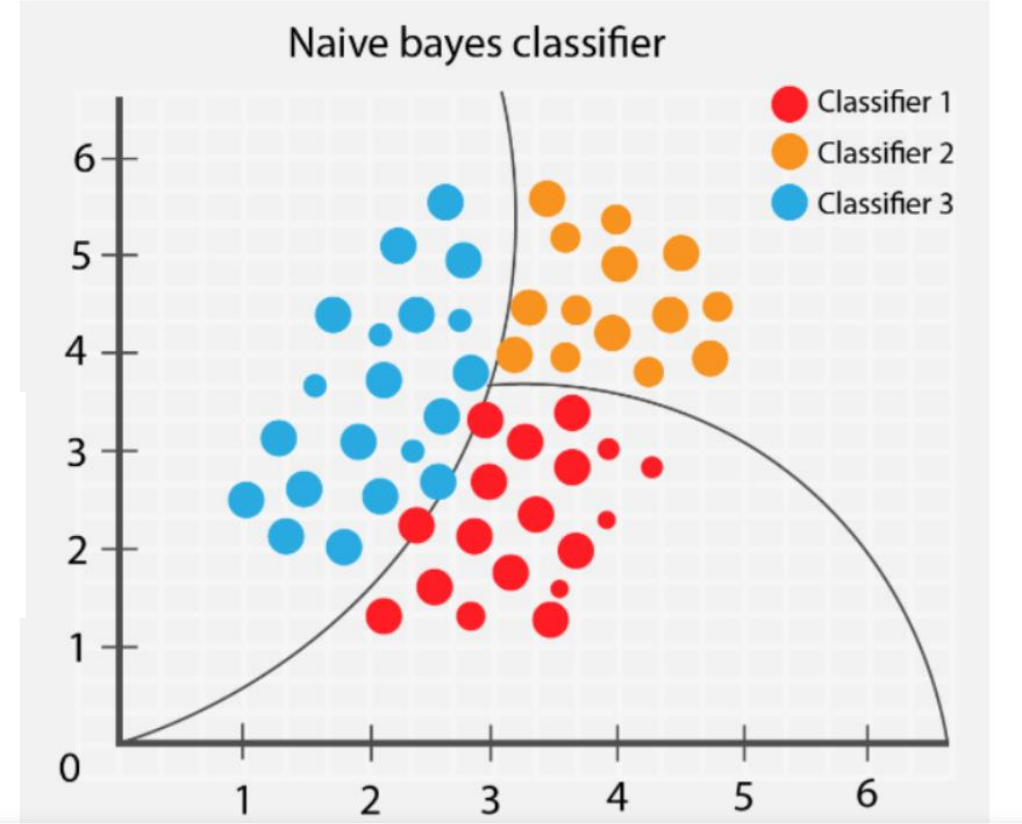
Naive Bayes Algorithm

What is Naïve bayes algorithm?

- Probabilistic machine learning algorithm
- It calculates the probability of a data point belonging to a certain class by assuming the independence between its features.
- It uses bayes theorem

$$P(class/features) = \frac{P(class) * P(features/class)}{P(features)}$$

$$P(c \mid \mathbf{X}) = P(x_1 \mid c) \times P(x_2 \mid c) \times \cdots \times P(x_n \mid c) \times P(c)$$



Implementation and results

- Divided our training data into 80-20 split
- 80% training and 20% testing
- GaussianNB()
- Accuracy : 79.9%
- Precision: 64.95 %
- Recall: 31 %
- F1 score: 41.97 %

Hyperparameter Tunning

- Grid Search
- Var smoothing: [1e-9, 1e-8, 1e-7, 1e-6, 1e-5, 1e-4, 1e-3]
- CV = 5
- Total fit= 7*5 = 35 fits

var_smoothing	1e-9	1e-8	1e-7
Accuracy	79.39%	79.34%	79.37%
var_smoothing	1e-6	1e-5	1e-4
Accuracy	79.4%	80.3%	79.67%

Best performance

- Var smoothing = 1e-5
- Accuracy : 80.3%
- Precision: 72.54 %
- Recall: 30.34 %
- F1 score: 42.8 %

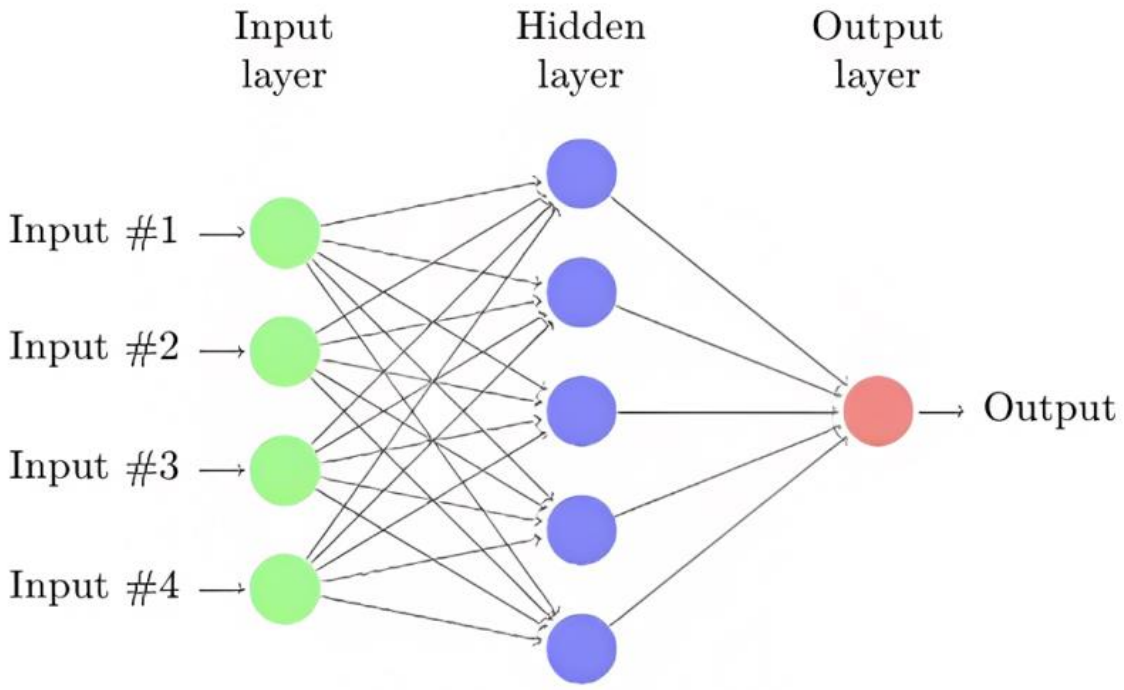
Artificial neural network

What is ANN?

- It is a computational model that mimics the way nerve cells work in the human brain

Architecture of ANN

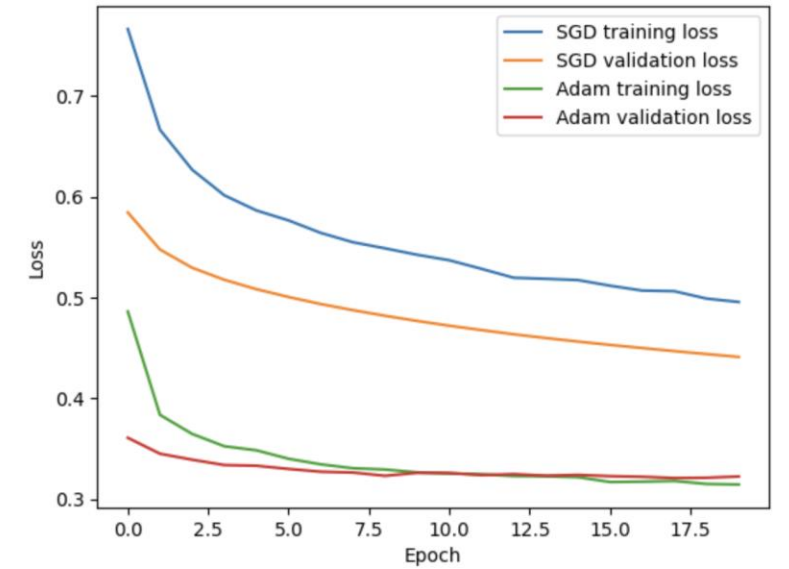
- Input Layer
- Output Layer
- Hidden Layer
- Neurons
- Weights
- Biases

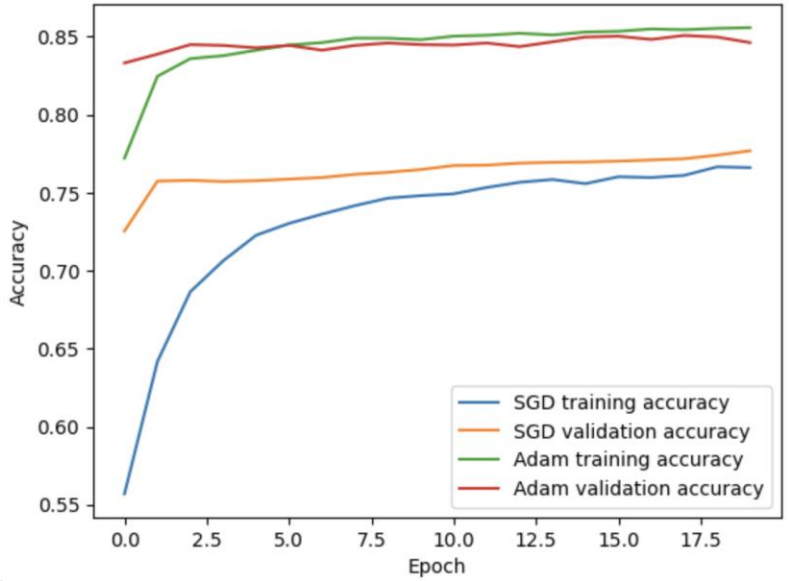


Our model Architecture and training parameters

- Input layer: neurons = 64, activation = Relu
- Dropout = 0.5
- Hidden layer : neurons = 32, activation = Relu
- Dropout = 0.5
- Output layer: neurons=1, activation = sigmoid

- Optimizers used are Adam and Stochastic Gradient Descent
- Learning rate = 0.01
- Loss = Binary cross entropy
- Epochs = 20
- Batch size = 128
- Validation split = 0.1





Evaluation matrix

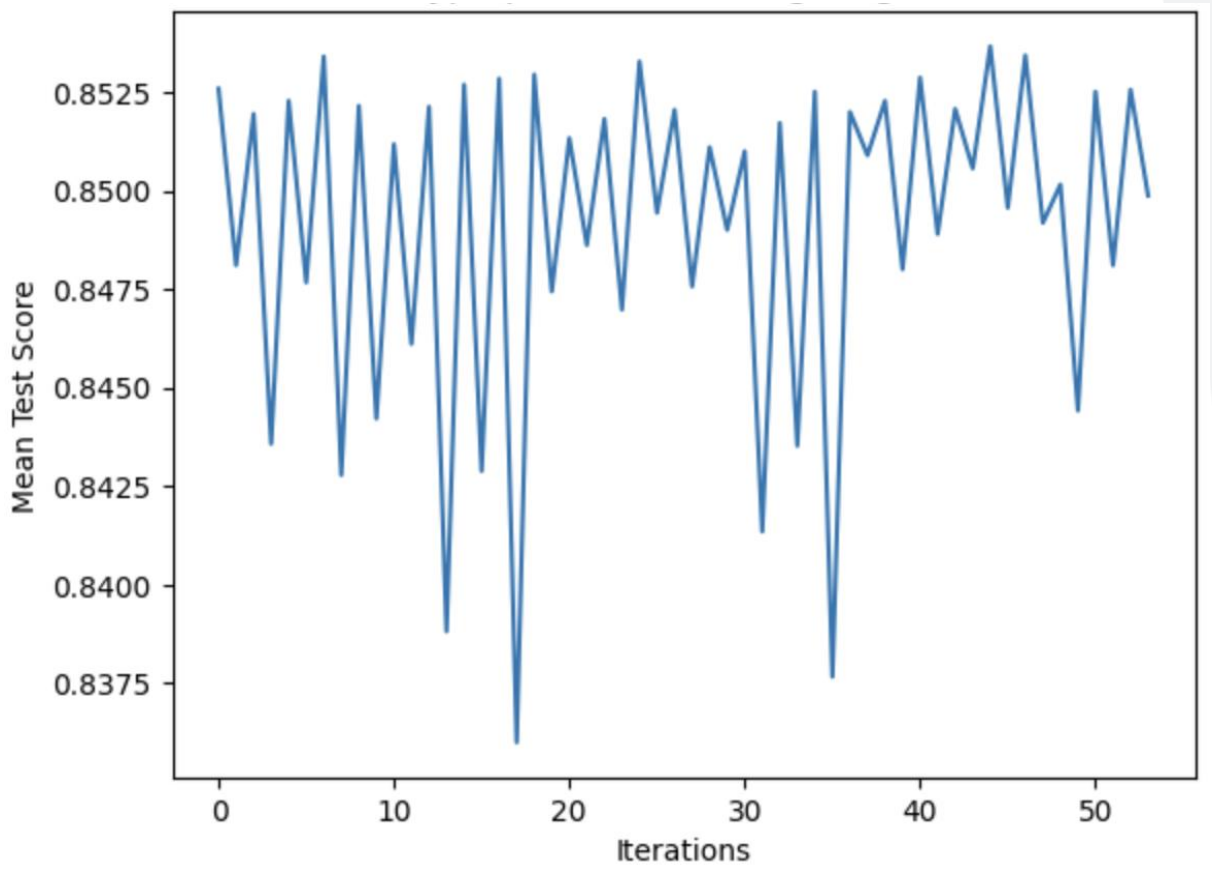
Key Things to Notice:

- Accuracy of Adam optimizer is more
- Precision of SGD optimizer is more but not significantly more
- Recall of Adam optimizer is more than that of SGD
- F1 score of Adam optimizer is more
- Bad F1 and Recall score shows that SGD optimizer is not good for our model

	SGD	Adam
Accuracy	0.795	0.860
Precision	0.771	0.759
Recall	0.177	0.594
F1-Score	0.288	0.666

Hyper Parameter tunning

- GridSearch
- Batch Size: 32, 64, 128
- Epochs: 10, 20, 30
- Optimizer: Adam(Learning rate=0.001), Adam(Learning rate= 0.01)
- Dropout rate: 0.2, 0.3, 0.5
- CV = 3
- Total fit = 3 * 3 * 2 * 3 * 3 = 162 fits
- Best Test score : batch size = 128, Dropout = 0.3, epochs 20, learning rate = 0.001

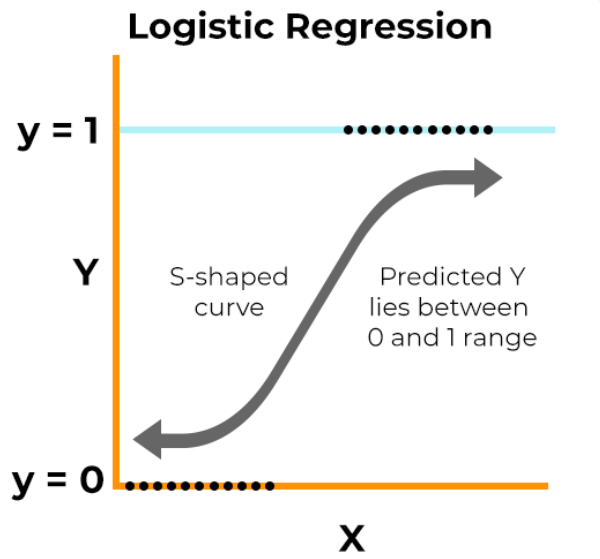


Logistic Regression

- The logistic model is a statistical model that models the probability of an event taking place by having the log-odds for the event be a linear combination of one or more independent variables
- Logistic regression estimates the parameters of a logistic model
- General Logistic Regression expression:

$$P(y = 1|x) = \frac{1}{1+e^{-(w^T x)}}$$

- Logistic Regression helps us to identify the data anomalies



Results Of Hyperparameter Logistic Regression

Performed Hyperparameter tuning

Regularization -> 'L1'
Optimizer -> 'liblinear'
Inverse Regularization Strength -> '10'

Accuracy Rate -> 83%

Comparing Logistic Regression Results Between Hyperparameter and Without Hyperparameter

	Confusion Matrix Without Hyperparameter Tuning		
True	True	4673	269
	False	1102	469
		True	False
	Predicted		

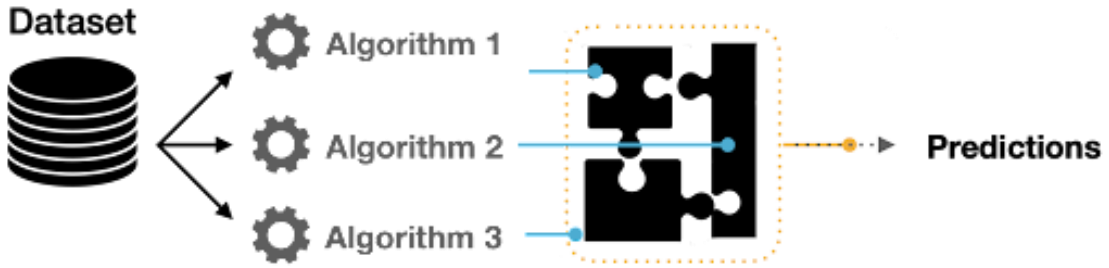
	Confusion Matrix With Hyperparameter Tuning		
True	True	4668	274
	False	862	709
		True	False
	Predicted		

Classification Report Without Hyperparameter Tuning				
	Precision	**Recall**	**F1-Score**	**Support**
0	**0.81**	**0.95**	**0.87**	**4942**
1	**0.64**	**0.30**	**0.41**	**1571**
Accuracy			**0.79**	**6513**
Macro Avg	**0.72**	**0.62**	**0.64**	**6513**
Weighted Avg	**0.77**	**0.79**	**0.76**	**6513**

Classification Report With Hyperparameter Tuning				
	Precision	**Recall**	**F1-Score**	**Support**
0	**0.84**	**0.94**	**0.89**	**4942**
1	**0.72**	**0.45**	**0.56**	**1571**
Accuracy			**0.83**	**6513**
Macro Avg	**0.78**	**0.70**	**0.72**	**6513**
Weighted Avg	**0.81**	**0.83**	**0.81**	**6513**

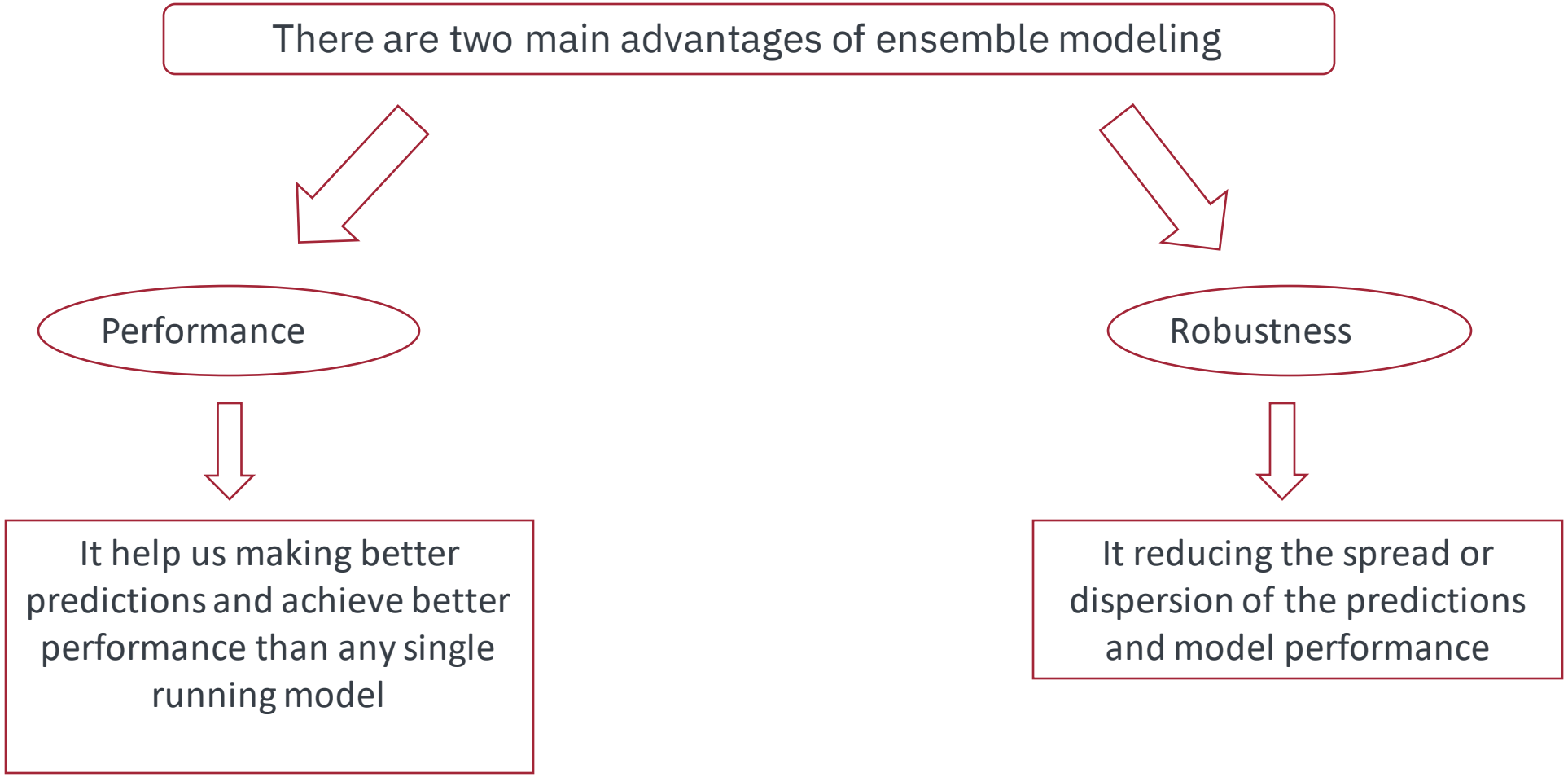
Ensemble Modeling

- Ensemble modelling is a technique in which various multiple models are created to predict an outcome, either by using various modeling algorithms or using different training datasets.
- In this project we are using different modeling algorithms approach.
- We have use three different algorithms for ensemble modeling which are listed below:
    1. Decision Tree
    2. Random Forest
    3. Logistic Regression



- We have use Vote Classifier for performing Ensemble modeling.

Advantages of Ensemble Modeling



Results of Hyperparameter Ensemble Modeling

Performed Hyperparameter tuning

Decision Tree -> max depth '10'
Logistic Regression -> inverse regularization strength '1.0'
Random Forest -> estimator '200'

Accuracy Rate -> 86%

Comparing Ensemble Model Results Between Hyperparameter and Without Hyperparameter

	Confusion Matrix Without Hyperparameter Tuning		
True	True	4627	315
	False	662	909
		True	False
	Predicted		

	Confusion Matrix With Hyperparameter Tuning		
True	True	4716	226
	False	687	884
		True	False
	Predicted		

Classification Report Without Hyperparameter Tuning				
	Precision	**Recall**	**F1-Score**	**Support**
0	**0.87**	**0.94**	**0.90**	**4942**
1	**0.74**	**0.58**	**0.65**	**1571**
Accuracy			**0.85**	**6513**
Macro Avg	**0.81**	**0.76**	**0.78**	**6513**
Weighted Avg	**0.84**	**0.85**	**0.84**	**6513**

Classification Report With Hyperparameter Tuning				
	Precision	**Recall**	**F1-Score**	**Support**
0	**0.87**	**0.95**	**0.91**	**4942**
1	**0.80**	**0.56**	**0.66**	**1571**
Accuracy			**0.86**	**6513**
Macro Avg	**0.83**	**0.76**	**0.79**	**6513**
Weighted Avg	**0.85**	**0.86**	**0.85**	**6513**

Comparison

ML Algorithm Used	Accuracy	F1-Score
Decision Tree	85.60%	67.54%
Random Forest	85.59%	67.86%
SVM	79.71%	63.68%
ANN with ADAM	86.10%	66.60%
ANN with SGD	78.40%	28.80%
Naive Bayes	80.30%	42.8%
Logistic Regression	82.60%	55.52%
Ensemble Learning	85.98%	65.94%
Decision Tree with Feature Engineering	85.77	66.56%
Random Forest with Feature Engineering	85.73	67.84%

Future Potentials

- Imbalanced Data Handling
- Explainable AI
- Additional Data Sources

References

- S. Deepajothi and S. Selvarajan, “A comparative study of classification techniques on adult data set,” International Journal of Engineering Research & Technology (IJERT), vol. 1, no. 8, 2012.
- R. Sabitha and S. Karthik, “Performance assessment of feature selection methods using k-means on adult dataset,” CCIIT, vol. 4, pp. 606–612, 2013.
- N. Chakrabarty and S. Biswas, “A statistical approach to adult census income level prediction,” in 2018 International Conference on Advances in Computing, Communication Control and Networking (ICACCCN). IEEE, 2018, pp. 207–212.

STEVENS
INSTITUTE OF TECHNOLOGY
1870

THANK **YOU**

Stevens Institute of Technology
1 Castle Point Terrace, Hoboken, NJ 07030