MACHINE LEARNING ASSIGNMENT

BY ARJUN AGARWAL

INTRODUCTION

Using machine learning algorithms, such as decision trees, to predict income level can offer important insights into understanding the variables that affect a person's earning potential. The goal of the income level prediction issue is to detect whether a person's pay is above or below a predetermined threshold, usually marked as ">50K" or "=50K," using a variety of attributes, including age, education, occupation, and more.

A decision tree algorithm is a good choice for this job since it is versatile enough to analyse a wide range of characteristics and can handle both category and numerical variables. Decision trees create a tree-like structure by repeatedly splitting the dataset based on attribute values. The decision tree method selects the attribute that best separates the data at each node, resulting in a set of understandable decision rules.

The system learns to predict outcomes based on attribute values by training a decision tree on a labelled dataset with known income levels. The resulting decision tree can highlight the most crucial characteristics for predicting income and offer an intuitive understanding of their role in the categorization task. The tree may show, for instance, that certain jobs and greater education levels are excellent indicators of higher income. New instances can be categorised according to the values of their attribute to anticipate their revenue level using the trained decision tree. A predicted revenue level is finally assigned to each instance by the decision tree algorithm after evaluating the qualities in accordance with the learned decision rules. Performance indicators like accuracy, precision, and recall can be used to assess and validate this prediction in more detail.

DATASET The Dataset

The Census Income dataset has 48,842 entries. Each entry contains the following information about an

individual:

- age: the age of an individual
  - Integer greater than 0
- workclass: a general term to represent the employment status of an individual
  - Private, Selfempnotinc, Selfempinc, Federalgov, Localgov, Stategov,

Withoutpay, Neverworked.

- fnlwgt: final weight. In other words, this is the number of people the census believes

the entry represents..

  - Integer greater than 0
- education: the highest level of education achieved by an individual.

○ Bachelors, Somecollege, 11th, HSgrad, Profschool, Assocacdm, Assocvoc, 9th, 7th8th, 12th, Masters, 1st4th, 10th, Doctorate, 5th6th, Preschool.

● educationnum: the highest level of education achieved in numerical form.

○ Integer greater than 0

● maritalstatus: marital status of an individual. Marriedcivspouse corresponds to a civilian spouse while MarriedAFspouse is a spouse in the Armed Forces.

○ Marriedcivspouse, Divorced, Nevermarried, Separated, Widowed, Marriedspouseabsent, MarriedAFspouse.

● occupation: the general type of occupation of an individual

○ Techsupport, Craftrepair, Otherservice, Sales, Execmanagerial, Profspecialty, Handlerscleaners, Machineopinspct, Admclerical, Farmingfishing, Transportmoving, Privhouseserv, Protectiveserv, ArmedForces.

● relationship: represents what this individual is relative to others. For example an individual could be a Husband. Each entry only has one relationship attribute and is somewhat redundant with marital status. We might not make use of this attribute at all

○ Wife, Ownchild, Husband, Notinfamily, Otherrelative, Unmarried.

● race: Descriptions of an individual's race

○ White, AsianPacIslander, AmerIndianEskimo, Other, Black.

● sex: the biological sex of the individual

○ Male, Female

● capitalgain: capital gains for an individual

○ Integer greater than or equal to 0

● capitalloss: capital loss for an individual

○ Integer greater than or equal to 0

● hoursperweek: the hours an individual has reported to work per week

○ continuous.

● nativecountry: country of origin for an individual

UnitedStates, Cambodia, England, PuertoRico, Canada, Germany, OutlyingUS(GuamUSVIetc), India, Japan,

Greece, South, China, Cuba, Iran, Honduras, Philippines, Italy, Poland, Jamaica, Vietnam, Mexico, Portugal,

Ireland, France, DominicanRepublic, Laos, Ecuador, Taiwan, Haiti, Columbia, Hungary, Guatemala, Nicaragua,

Scotland, Thailand, Yugoslavia, ElSalvador,

Trinadad&Tobago, Peru, Hong, HolandNetherlands.

● the income: whether or not an individual makes more than $50,000 annually.

○ <=50k, >50k

METHODOLOGY

Data Cleaning

In the data preprocessing pipeline, data cleaning is a crucial step that aims to enhance data quality and

guarantee its dependability for analysis. With its wide variety of libraries and tools, Python has become a wellliked option for carrying out data cleaning tasks speedily and successfully.

Pandas and NumPy are two eĊective Python libraries that oĊer comprehensive data structures and functions

for data manipulation. The DataFrame, a flexible data structure that enables seamless data management,

exploration, and transformation, is specifically introduced by Pandas. Pandas makes it simple to load data into

a DataFrame, allowing for fast analysis and modification.

The missing values are substituted with statistical values such as mean, median or mode depending on the

data type as well as the distribution of data for a particular data column. Moreover due to varying datatypes in

such a large dataset with multiple covariates, the data is encoded from string to float to ensure uniformity.

Removal of Features

We also opted to not use the features: ‘fnlwgt’, ‘relationships’, and ‘capitalGains/Loss’. These features eitherwere not useful for our analysis or had too much bad data i.e. zerovalues, unknown/private

values.

	age	workclass	fnlwgt	...	native-country	income	age_group
0	50	5	83311	...	36	0	3
1	38	3	215646	...	36	0	2
2	53	3	234721	...	36	0	4
3	28	3	338409	...	4	0	1
4	37	3	284582	...	36	0	2
...	...	...	...	...	...	...	...
6895	37	1	218184	...	36	1	2
6896	27	5	206889	...	36	0	1
6897	35	3	110668	...	36	0	2
6898	30	3	211028	...	36	0	1
6899	64	1	202984	...	36	0	5

	age	workclass	fnlwgt	...	income	age_group	hours-per-week-category
0	25	2	226802	...	0	1	0
1	38	2	89814	...	0	2	1
2	28	1	336951	...	1	1	0
3	44	2	160323	...	1	3	0
4	18	2	103497	...	0	0	2
...	...	...	...	...	...	...	...
6895	31	2	289228	...	0	2	2
6896	28	2	38918	...	0	1	1
6897	23	2	194630	...	0	1	1
6898	31	2	262848	...	0	2	0
6899	21	2	157595	...	0	1	2

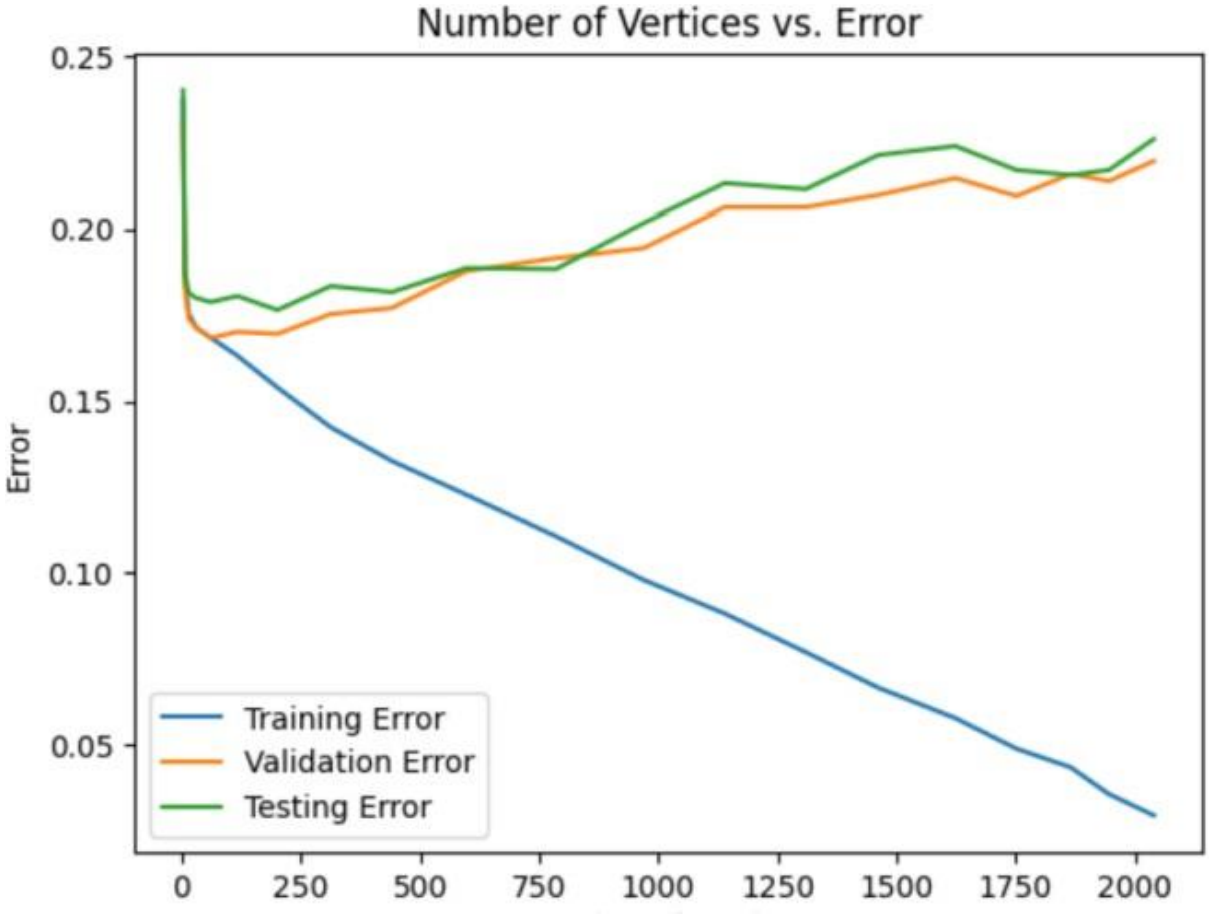
Decision Tree

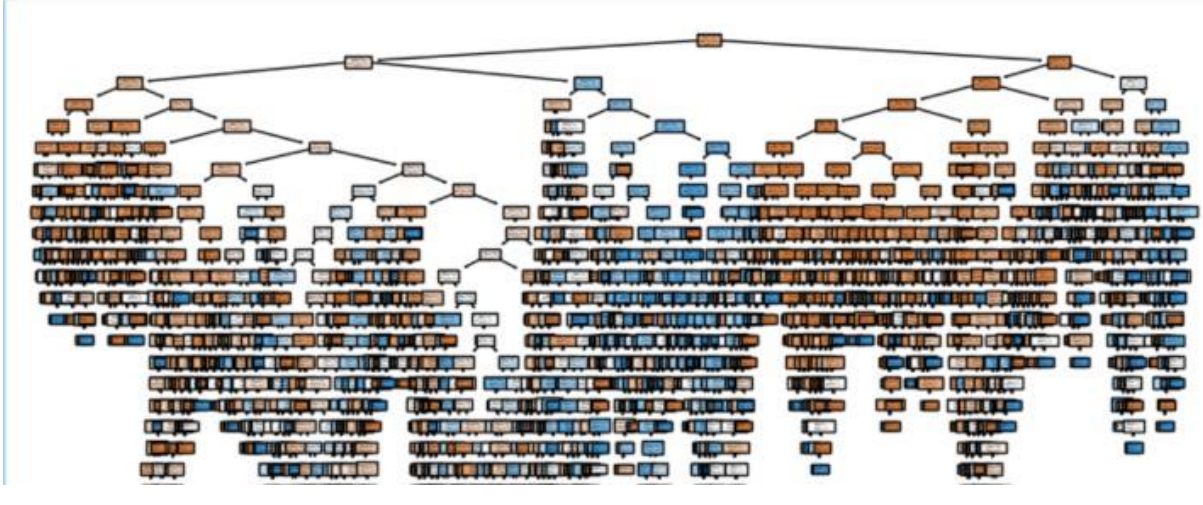
Popular supervised machine learning algorithms for classification and regression tasks include decision trees. It is a diagram that resembles a tree that displays choices and potential outcomes. The tree is made up of nodes and branches, where each node stands for a choice or feature and each branch for the result or potential course of action based on that choice. The tree's leaf nodes stand in for the result's final form.

The procedure typically entails three primary steps: data preparation, tree construction, and tree pruning, to produce a decision tree from a given dataset. Preprocessing the dataset, which may involve handling missing values, encoding categorical variables, and dividing the data into training and testing sets, is done during the data preparation step.

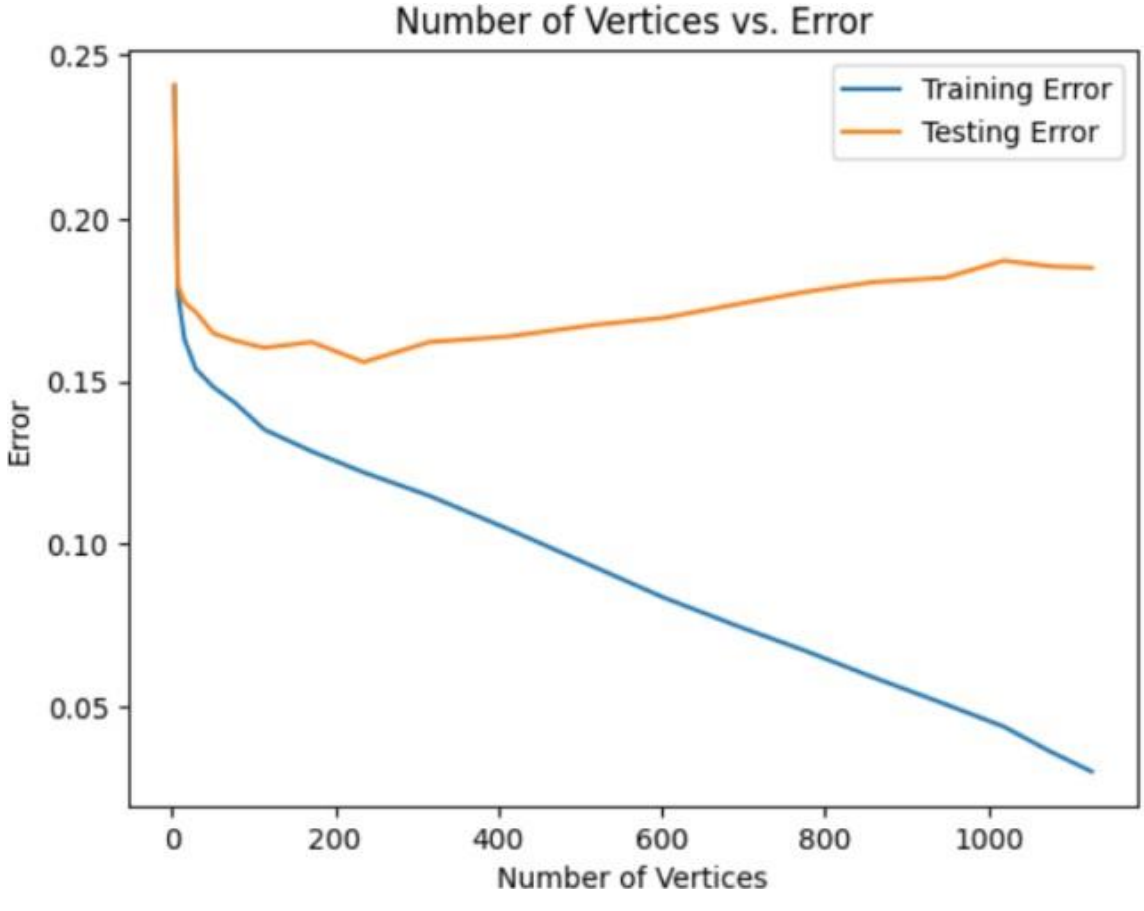
The decision tree algorithm separates the data recursively based on several traits or features to generate a tree structure during the tree construction phase. To increase the prediction potential of the tree, it chooses the best attribute at each step using measures like information gain or Gini impurity. Until a stopping requirement is satisfied, such as reaching a maximum depth or a minimum amount of samples in a leaf node, the recursive process keeps going.

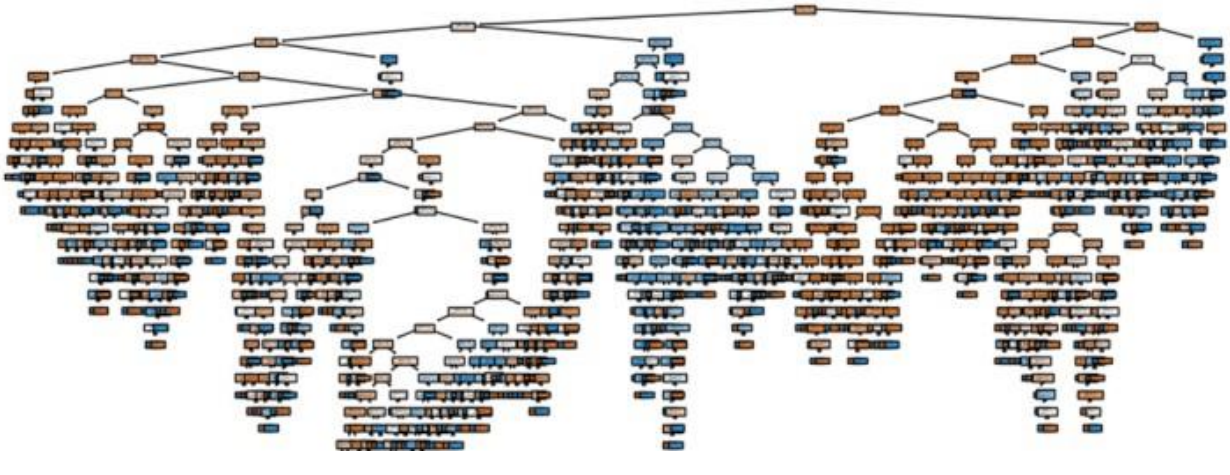
1)





2)


Number of Vertices vs. Error
Training Error
Testing Error
Error
0.25
0.20
0.15
0.10
0.05
0
200
400
600
800
1000
Number of Vertices



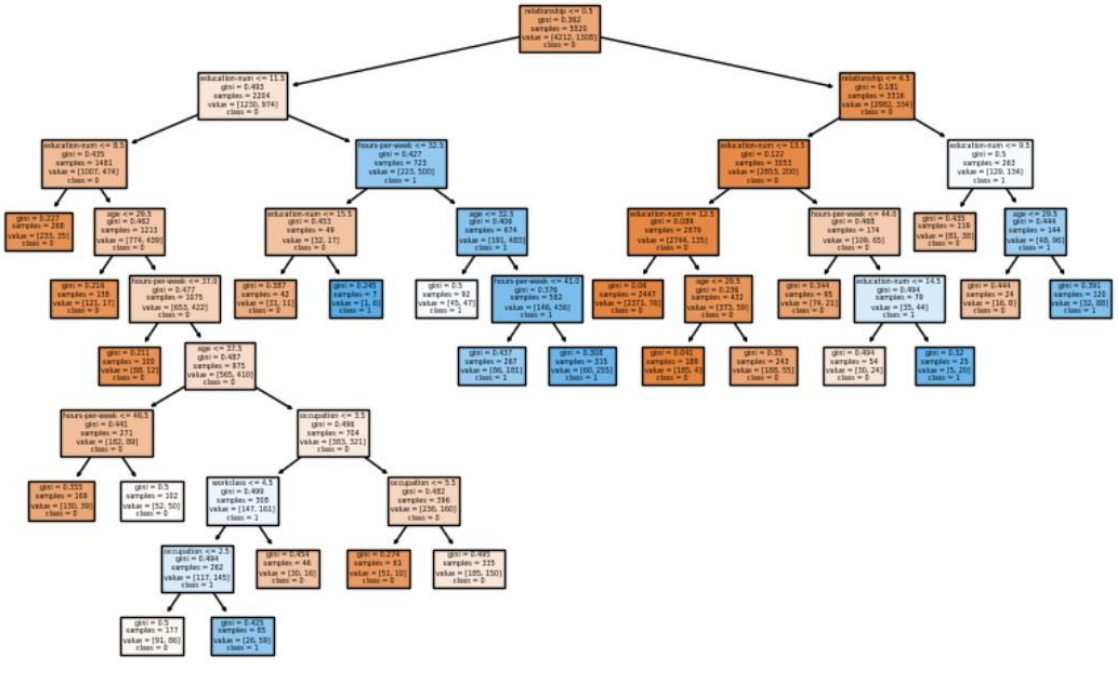
Pruning

Pruning is done after the decision tree is built to maximise its size and avoid overfitting. Pruning reduces the tree's complexity and gets rid of extra branches that can cause it to overfit the training set, which aids with generalisation. Reduced Error Pruning is one common technique used for pruning decision trees.

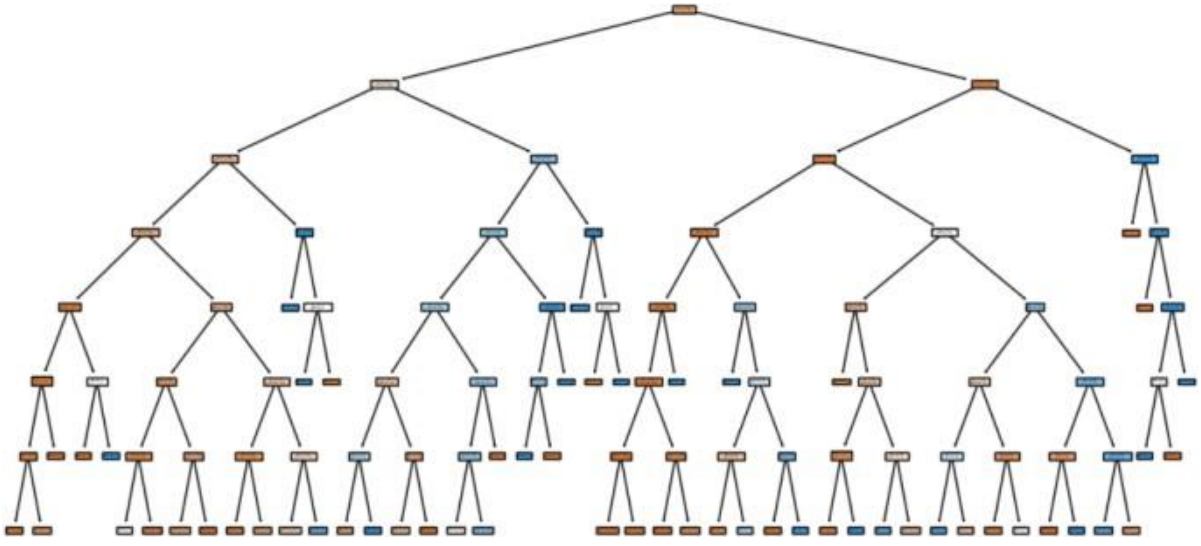
In Reduced Error Pruning, the decision tree is initially trained using the training dataset. Then, a separate validation dataset is used to evaluate the performance of the tree by calculating the error rate. Starting from the leaf nodes, each subtree is replaced with a leaf node if doing so results in a lower error rate on the validation data. This process is repeated iteratively until no further improvement is observed or a predefined stopping criterion is met.

Pruning helps to simplify the decision tree, making it more interpretable and improving its generalization capabilities on unseen data. It reduces the risk of overfitting, where the tree becomes too specific to the training data and performs poorly on new instances. By finding the optimal balance between complexity and accuracy, pruning ensures that the decision tree captures the underlying patterns in the data while avoiding excessive complexity and over-reliance on noisy or irrelevant features.

1)



2)



Process

We used two methods :

1)The validation data set should be taken to 50% of testing dataset and the remaining 50% should be used as

testing data.

2)Combine training and testing data points. Randomly select 67% of the data points as training data set and

the remaining data points as testing data set.

The two decision trees differ slightly as accuracy in 1) is 0.8307246376811595 whereas the accuracy in the second is 0.839262187088274. Hence the second method is slightly better as data categorization into training and testing is more randomized. Pruning involves removing unnecessary branches or subtrees from the tree to prevent overfitting and improve generalization.

The random trees also differ slightly as accuracy in 1) is 0.8171014492753623 whereas the accuracy in 2) is0.8541941150636803

By moving from the root node to the leaf nodes of a decision tree and then taking the path that leads to a particular result, one can retrieve the rules generated from the tree. The decision tree's internal nodes each stand for a condition or a query based on a feature, while the branches indicate various options for that feature's values. A decision tree could produce the following examples of rules:

1)If age > 35 and education = "Bachelor's Degree" and hours-per-week > 40, then the predicted income level is >50K.

2) If age <= 35 and education = "High School Diploma" and occupation = "Sales", then the predicted income level is <=50K.

3) If age > 40 and education = "Master's Degree", then the predicted income level is >50K.

These rules are simple to understand because they take into account the connections between various characteristics and how they affect the expected income level. For instance, according to the first rule, those who are older, have a bachelor's degree, and put in more than 40 hours a week are more likely to earn more than $50,000. These rules are simple enough for humans to comprehend and interpret, and they offer insights into the variables that affect income.

One benefit of decision trees is that their rules are naturally intuitive. They offer a clear and understandable illustration of the fundamental decision-making process. It is simpler to comprehend the reasoning behind the forecasts when complicated choice problems are broken down into a sequence of straightforward if-then conditions using decision trees. This interpretability is especially beneficial in fields like banking, healthcare, or legal applications where model transparency and explainability are essential.

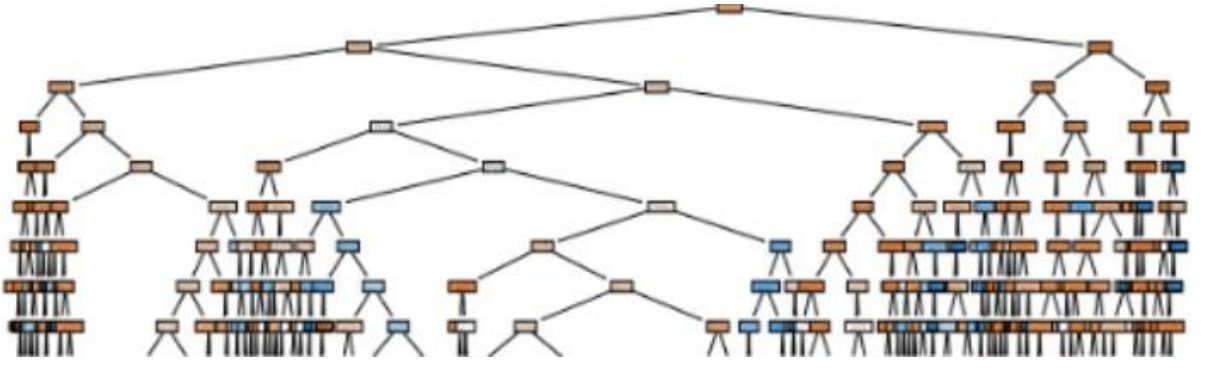
It's crucial to remember that as decision trees get bigger and more complicated, their interpretability may suffer. In such circumstances, the rules obtained from the decision tree may lengthen and grow more complex. Decision trees may also miss more intricate interactions or nonlinear correlations between variables, which could restrict their ability to predict outcomes in some circumstances. The principles produced from decision trees are nevertheless useful for comprehending the decision-making process and provide practical insights due to their intuitive character.

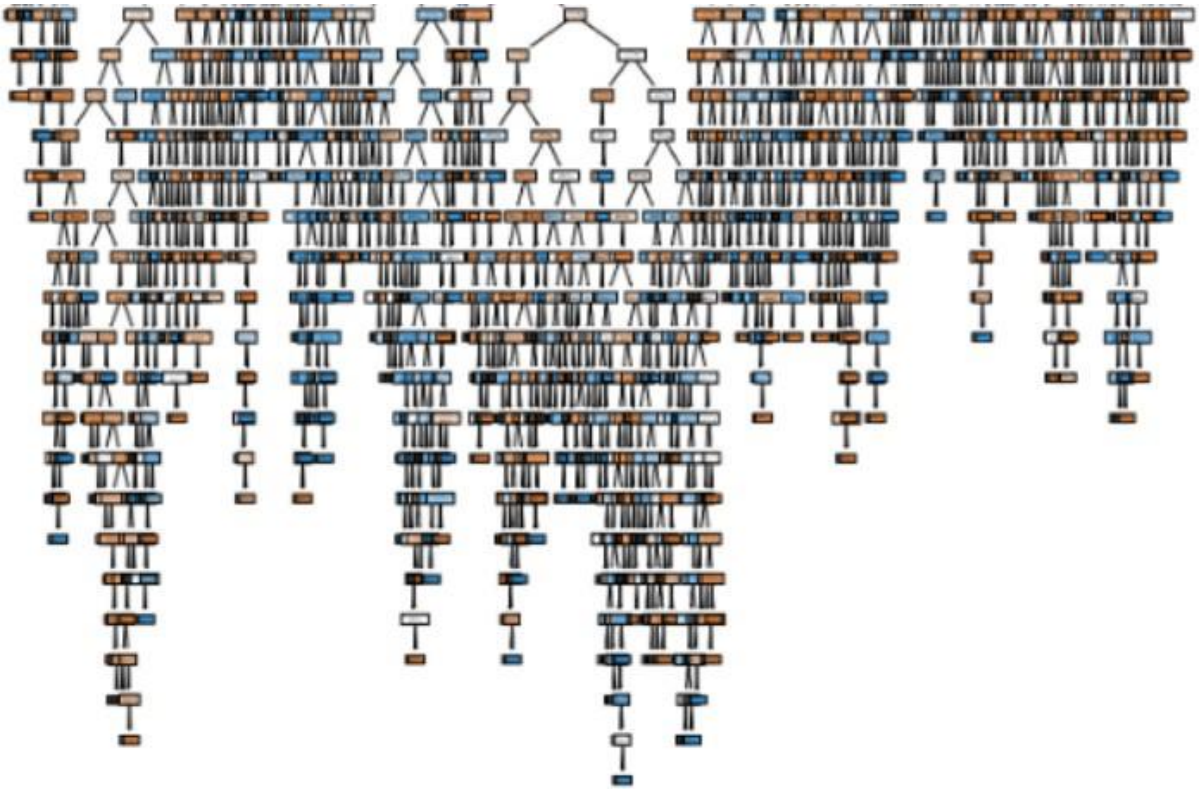
Random Forest Tree

A Decision Tree is a supervised learning method used in machine learning. Both classification and regression approaches are workable with it. As suggested by the name, it resembles a tree with nodes. The branches are determined by the number of criteria. Data is split into these branches up until a certain threshold unit. A decision tree has root nodes, child nodes, and leaf nodes. Despite its immense strength, random forest is also used for supervised learning. It's quite well-liked. The fundamental difference is that it is not dependent on a single choice. A final choice is made based on the majority of the assembled randomised decisions after several decisions have been made.

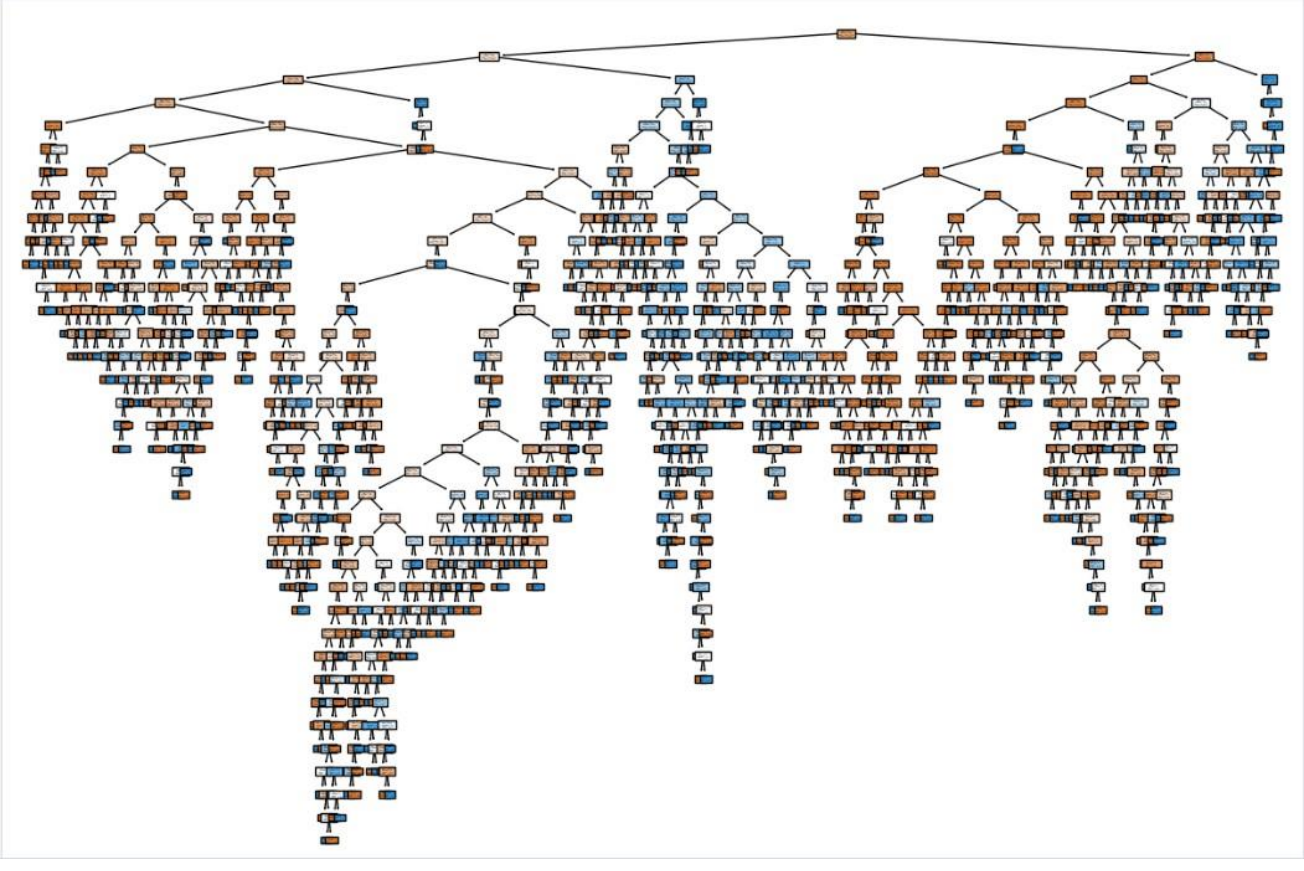
Comparing decision trees to a random forest, decision trees are much simpler. While a random forest combines many decision trees, a decision tree combines some decisions. As a result, it is a slow yet lengthy process. A decision tree, especially a linear one, is quick and works well with huge data sets.

1)





2)



In 1) accuracy of decision tree is more than random forest whereas in 2) the accuracy of random forest is morethan that of decision tree. Overall the accuracy of random forest is the best and should

be used as our machine learning model.

Naive Bayes

Naive Bayes is a popular machine learning model based on the Bayes' theorem. It is known for its simplicity and efficiency, particularly in text classification and spam filtering tasks. Despite its "naive" assumption of feature independence, it often performs well in practice.

Here's a high-level overview of how the Naive Bayes algorithm works:

Data preprocessing: The training data is preprocessed by converting it into a suitable format for the algorithm. In the case of text classification, this might involve tokenization, removing stop words, and creating a bag-of-words representation.

Calculating class probabilities: The algorithm starts by calculating the prior probabilities of each class based on the training data. This involves counting the occurrences of each class in the training set and dividing it by the total number of samples.

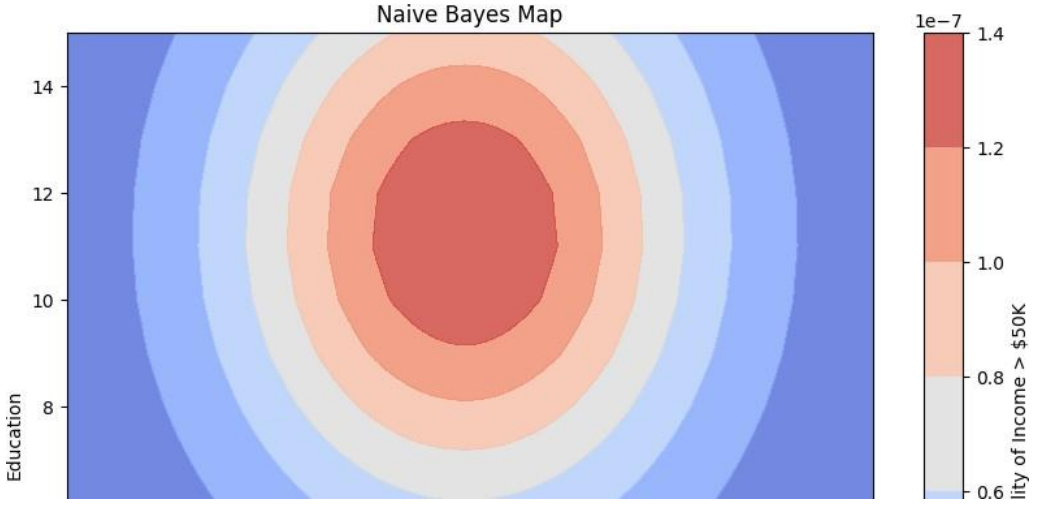
Calculating feature probabilities: For each feature (e.g., word) in the dataset, the algorithm calculates the likelihood of observing that feature given each class. This is done by counting the occurrences of the feature within each class and dividing it by the total number of samples in that class.

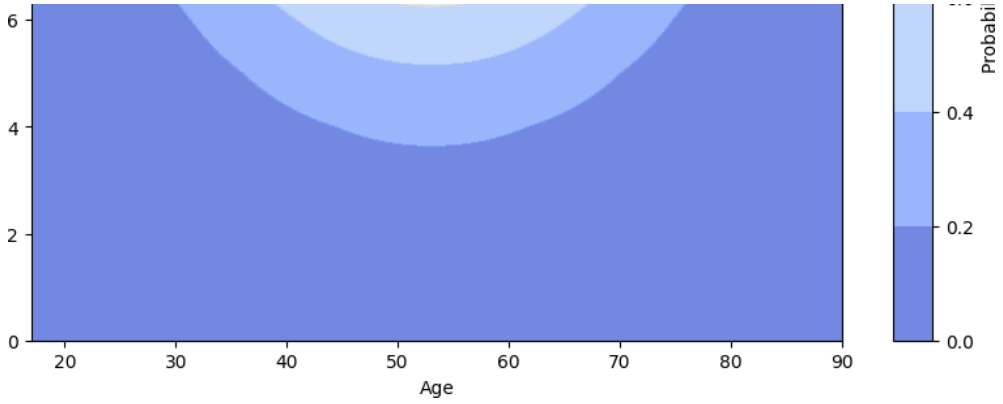
Making predictions: To predict the class of a new sample, the algorithm applies Bayes' theorem. It multiplies the prior probability of each class by the likelihood of observing the features in that class. The class with the highest resulting probability is chosen as the predicted class for the new sample.

Model evaluation: After training the model, it can be evaluated using a separate test set. Common evaluation metrics for classification tasks include accuracy, precision, recall, and F1 score.

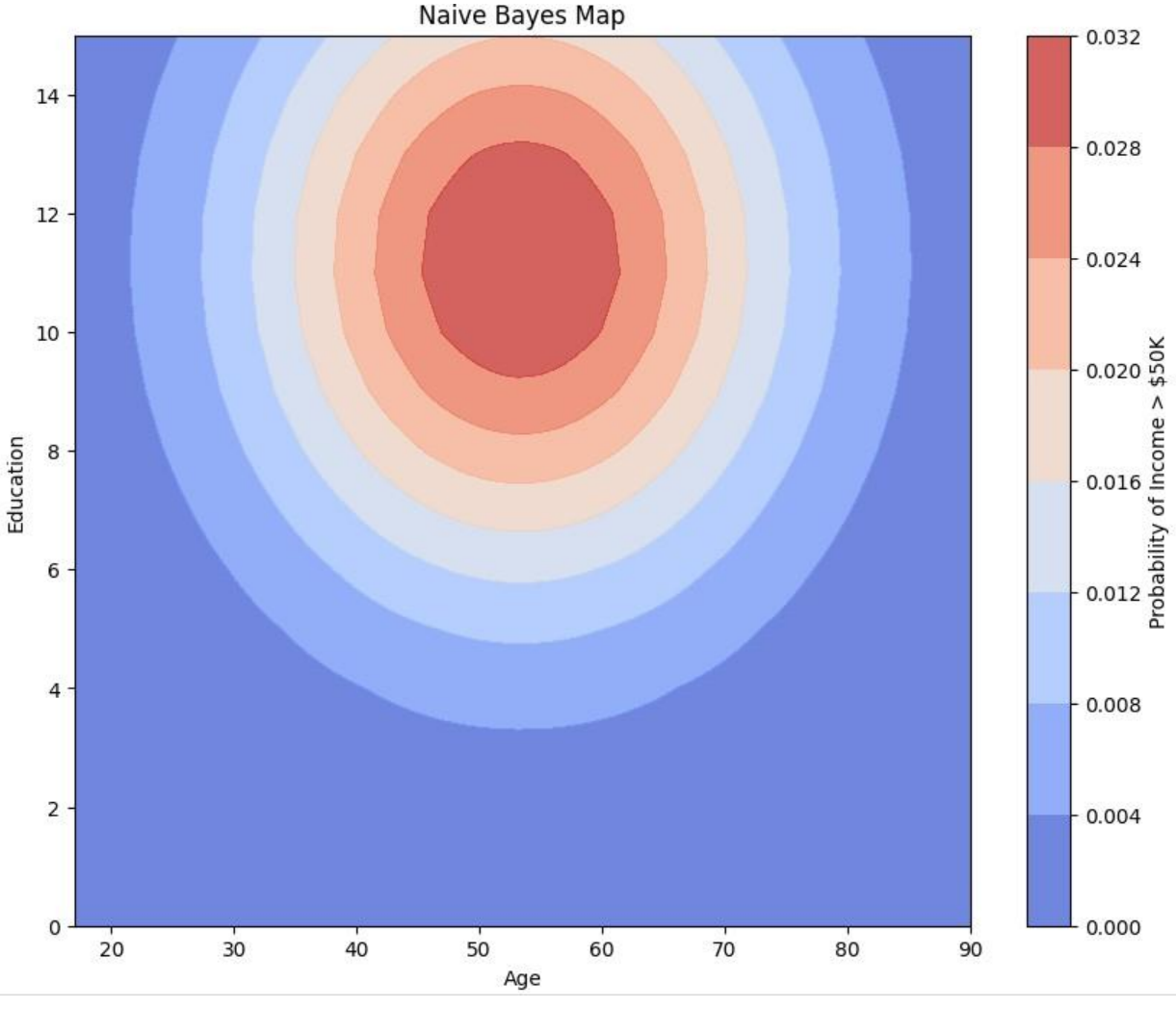
Naive Bayes models are relatively fast to train and make predictions, especially compared to more complex models like deep neural networks. However, they make the assumption of feature independence, which may not always hold in real-world scenarios. Despite this limitation, Naive Bayes can still perform well, particularly on text and categorical data classification tasks.

1)





2)



accuracy of 1) is 0.782608695652174 and that of 2) is 0.821256038647343

Logistic Regressionn

Logistic regression is a popular statistical model used for binary classification tasks. It is a supervised learning algorithm that predicts the probability of an instance belonging to a particular class. Despite its name, logistic regression is primarily used for classification rather than regression tasks.

Here's a high-level overview of how the logistic regression model works:

Data representation: The input data is represented as a feature matrix, where each row represents an instance and each column represents a feature. The features can be continuous, categorical, or binary.

Hypothesis function: Logistic regression uses a logistic or sigmoid function to model the relationship between the features and the binary output variable. The sigmoid function maps any real-valued number to a value between 0 and 1, representing the probability of the instance belonging to the positive class.

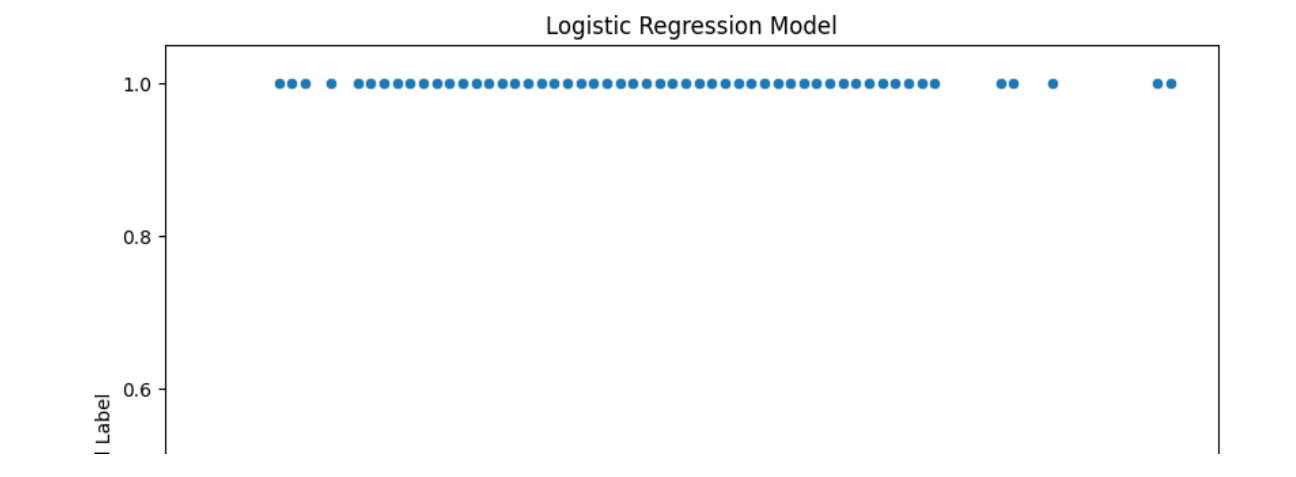
Model training: The logistic regression model is trained by estimating the optimal parameters that minimize the difference between the predicted probabilities and the actual class labels in the training data. This is typically done using optimization techniques like maximum likelihood estimation or gradient descent.

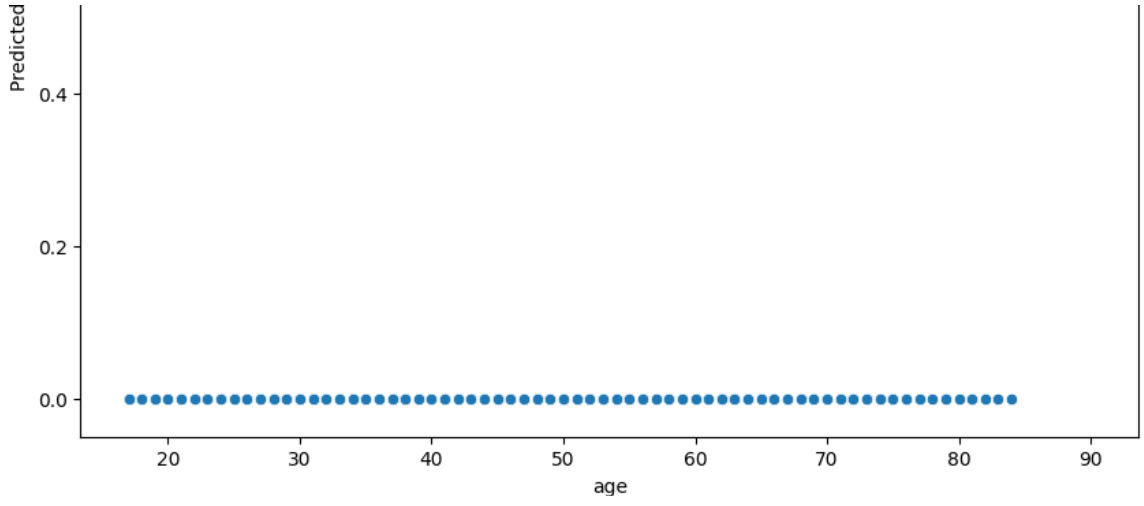
Decision boundary: Once the model is trained, it can be used to make predictions on new instances. The model calculates the probability of each instance belonging to the positive class, and a decision threshold (usually 0.5) is applied to determine the predicted class. If the probability is above the threshold, the instance is classified as belonging to the positive class; otherwise, it is classified as belonging to the negative class.

Model evaluation: The performance of the logistic regression model is evaluated using various metrics such as accuracy, precision, recall, and F1 score. These metrics assess how well the model is able to correctly classify instances from the test data.

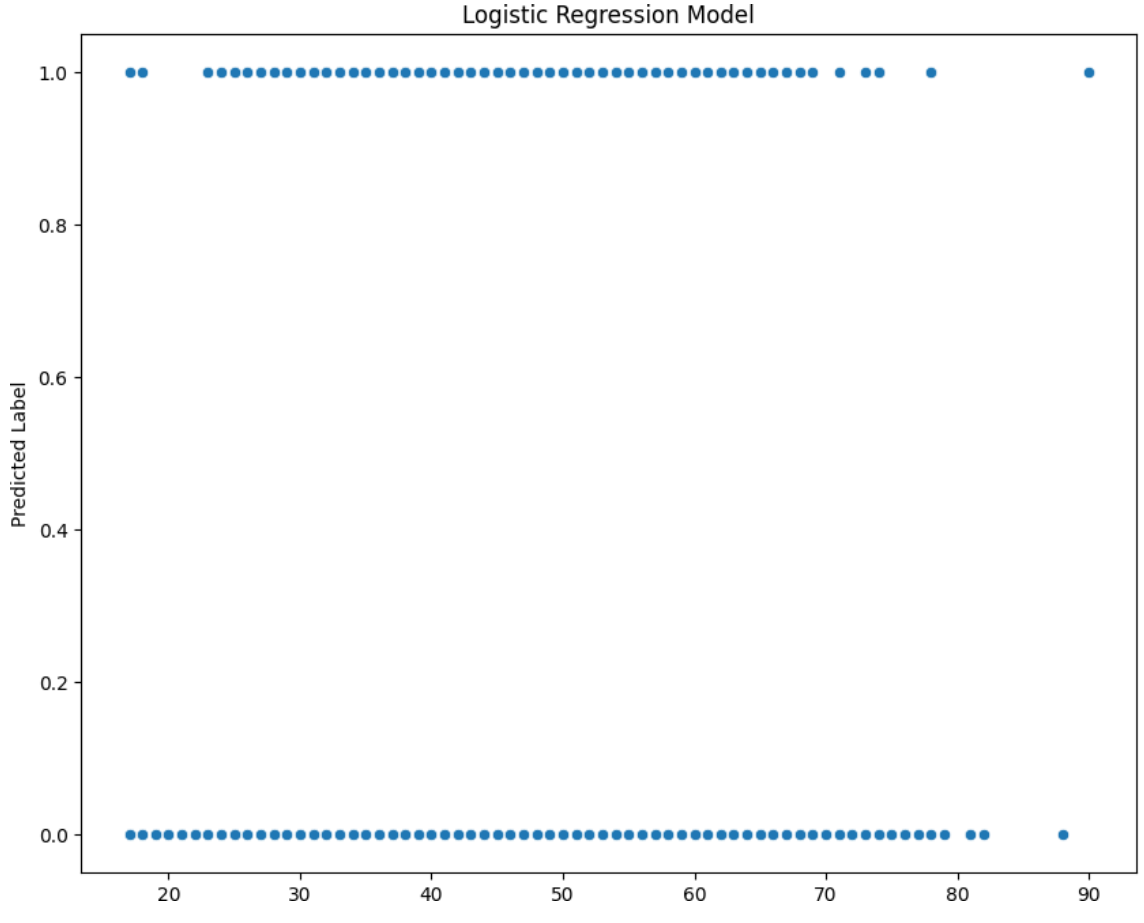
Logistic regression has several advantages, including simplicity, interpretability, and efficiency. It can handle both continuous and categorical features, and it provides probabilistic outputs, which can be useful in certain applications. However, logistic regression assumes a linear relationship between the features and the log-odds of the target variable, and it may not perform well in cases where the relationship is highly nonlinear or when there are complex interactions among the features. In such cases, more advanced models like decision trees or neural networks may be more suitable.

1)





2)



accuracy of 1) is 0.7921739130434783 and that of 2) 0.7878787878787878

Neural Network

A neural network is a powerful machine learning model inspired by the structure and function of biological neurons. It is widely used for a variety of tasks, including image recognition, natural language processing, and time series analysis. Neural networks are particularly effective in handling complex patterns and relationships in data.

Here's a high-level overview of how a neural network works:

Architecture: A neural network consists of an interconnected network of nodes called neurons organized in layers. The three main types of layers are the input layer, hidden layers, and output layer. The input layer receives the input data, the hidden layers process the data through a series of transformations, and the output layer produces the final predictions or outputs.

Weights and biases: Each connection between neurons in the network is associated with a weight, which represents the strength or importance of that connection. Additionally, each neuron has an associated bias, which allows for the flexibility of the network to capture complex relationships.

Activation functions: Activation functions introduce non-linearities to the output of each neuron. Common activation functions include the sigmoid, tanh, and ReLU (Rectified Linear Unit). These functions help the network model complex relationships and make it capable of learning non-linear patterns.

Forward propagation: During the forward propagation phase, the input data is passed through the network layer by layer. The outputs of each layer are calculated using the weighted sum of inputs, applying the activation function, and passing the result to the next layer.

Loss function: A loss function measures the error or mismatch between the predicted outputs of the network and the actual labels. The choice of the loss function depends on the specific task. For example, in binary classification, the cross-entropy loss function is commonly used.

Backpropagation: Backpropagation is the process of updating the weights and biases of the network based on the error calculated by the loss function. It propagates the error backward through the network, calculating gradients and adjusting the parameters using optimization algorithms like stochastic gradient descent (SGD), Adam, or RMSprop.

Training: The neural network is trained by iteratively feeding the training data through the network, performing forward propagation, calculating the loss, and updating the parameters using backpropagation. This process continues until the network converges to a state where the loss is minimized.

Prediction: Once the neural network is trained, it can be used to make predictions on new, unseen data. The forward propagation process is applied to the new data, and the output of the network provides the predicted values or probabilities for the given task.

Neural networks can be deep, meaning they have multiple hidden layers, resulting in deep learning models. Deep neural networks have demonstrated outstanding performance in various domains, but they require large amounts of data and computational resources for training. Additionally, techniques like regularization, dropout, and batch normalization are commonly employed to improve the generalization and stability of the models.

1) accuracy is 0.6662189364433289

2) accuracy is 0.7476600408554077

Interpretability

The random forest has highest accuracy whereas neural network has the lowest accuracy for the given data.

CONCLUSION

In conclusion, machine learning algorithms like decision trees provide a strong method to forecast revenuelevels based on provided attributes. By taking advantage of decision trees' interpretability, we can learn moreabout the major variables affecting income and offer useful recommendations for people, organizations, andpolicymakers who are trying to comprehend and analyze income disparities and make decisions.