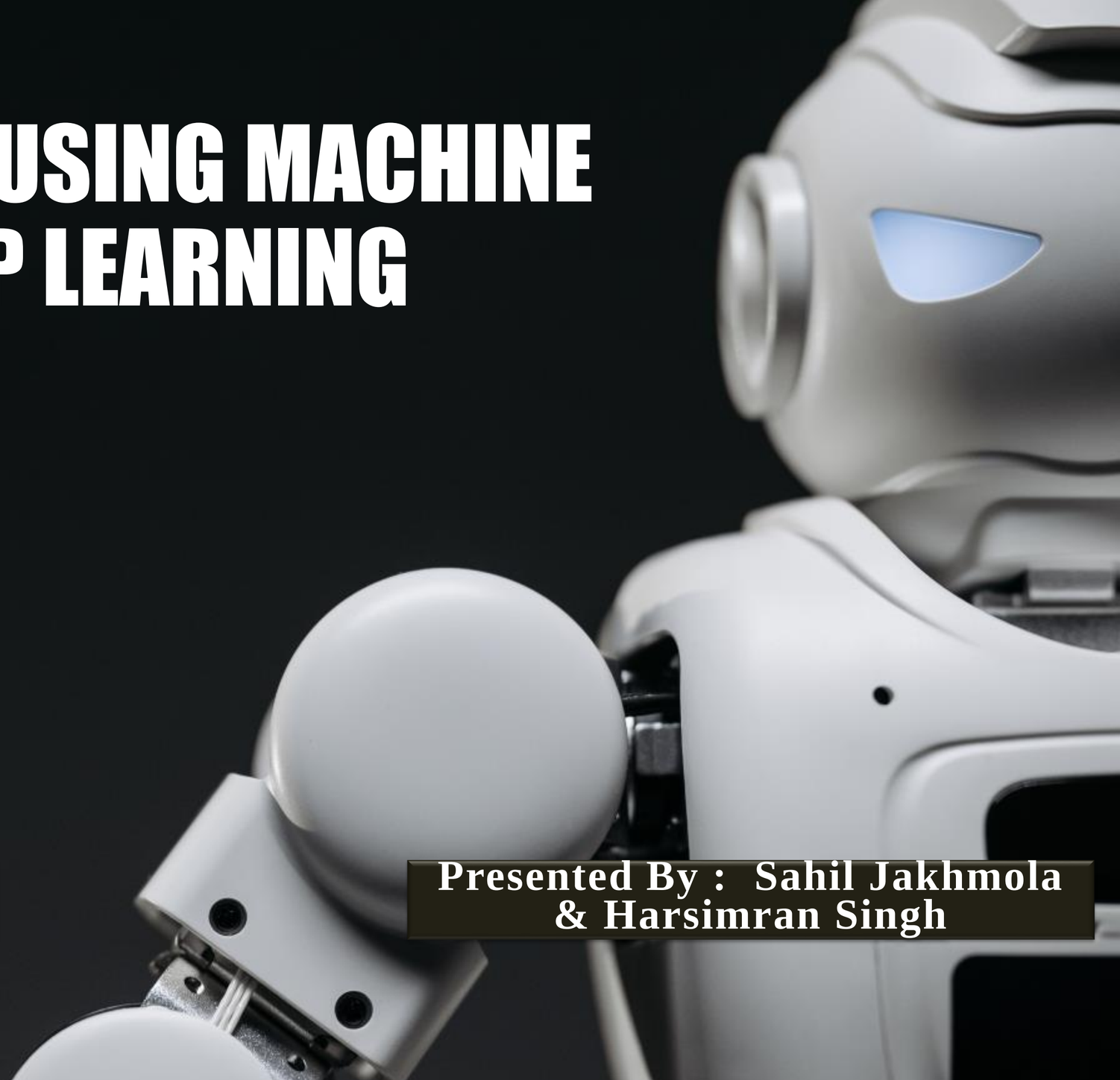
NIFTY PREDICTION USING MACHINE LEARNING AND DEEP LEARNING TECHNIQUES



Submitted to : Mayank Raghuwanshi

Presented By : Sahil Jakhmola & Harsimran Singh

WHAT IS MACHINE LEARNING AND DEEP LEARNING

- ML involves creating algorithms that enable computers to learn from and make predictions based on data.

- A subset of ML focused on neural networks with many layers (deep neural networks) that excel in processing large and complex datasets.

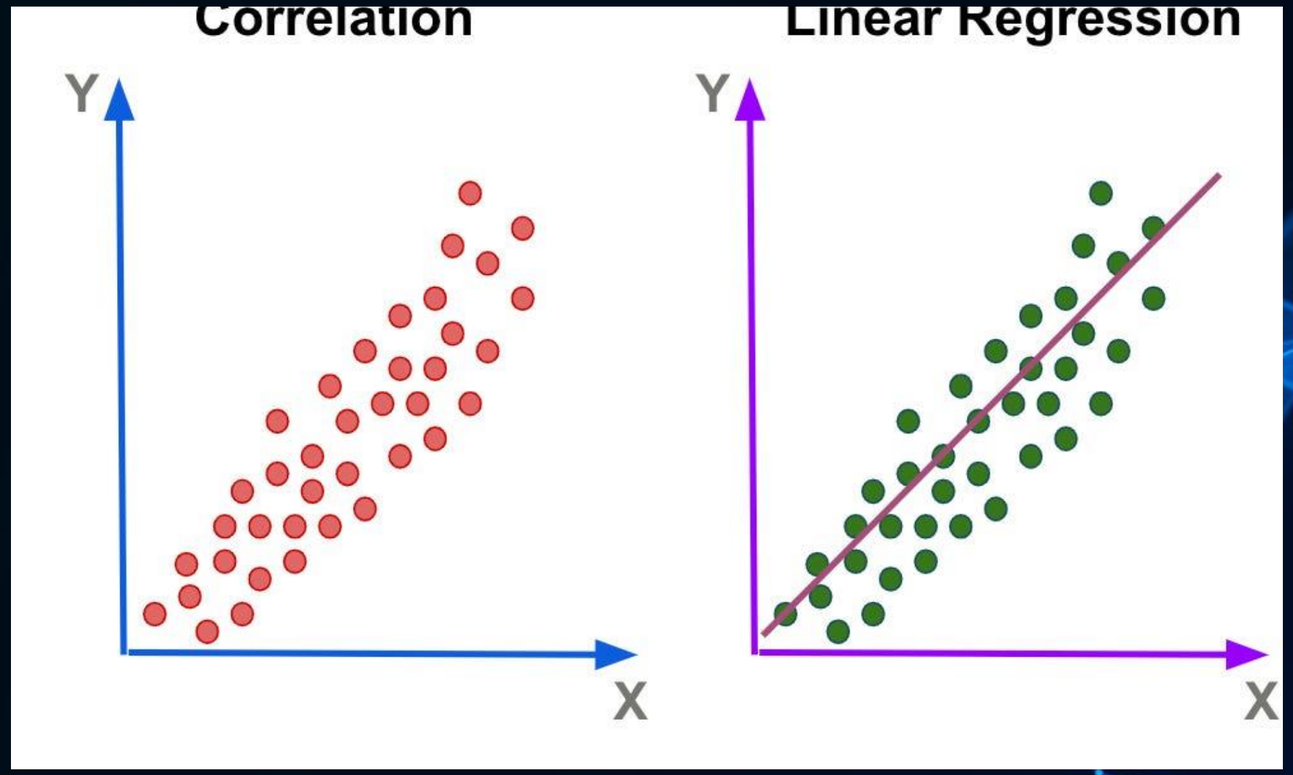
KEY CONCEPT USED IN PROJECT

- Linear Regression
- Decision Trees
- K-Nearest Neighbors
- Long Short-Term Memory
- Support Vector Regression

LINEAR REGRESSION

Linear regression is a type of supervised machine learning algorithm that computes the linear relationship between the dependent variable and one or more independent features by fitting a linear equation to observed data

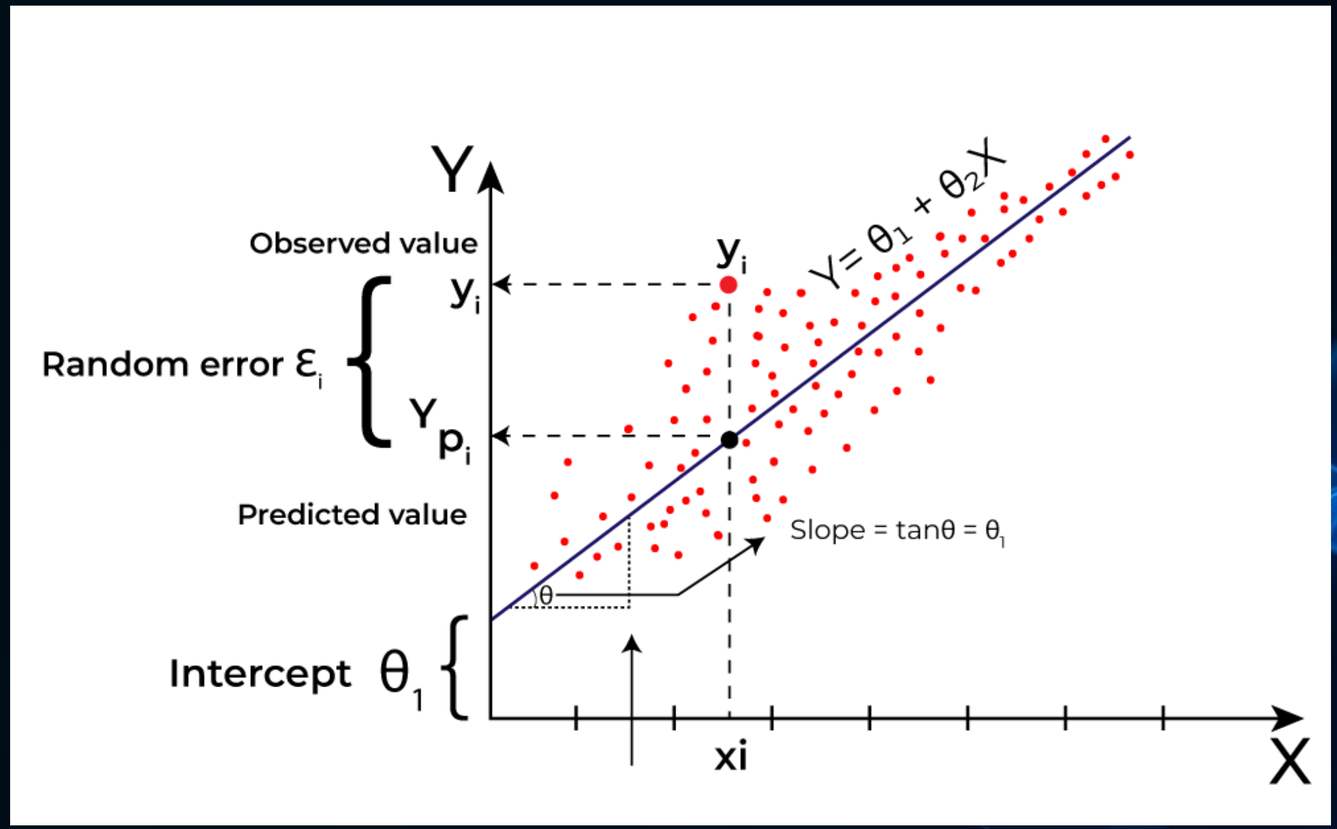
Linear regression analysis is used to predict the value of a variable based on the value of another variable. The variable you want to predict is called the dependent variable. The variable you are using to predict the other variable's value is called the independent variable.



Fit Line

Our primary objective while using linear regression is to locate the best-fit line, which implies that the error between the predicted and actual values should be kept to a minimum. There will be the least error in the best-fit line.

The best Fit Line equation provides a straight line that represents the relationship between the dependent and independent variables. The slope of the line indicates how much the dependent variable changes for a unit change in the independent variable.



K-NEAREST NEIGHBORS

The K-Nearest Neighbors (KNN) algorithm operates on the principle of similarity, where it predicts the label or value of a new data point by considering the labels or values of its K nearest neighbors in the training dataset.

STEPS

Step 1: Selecting the optimal value of K

- K represents the number of nearest neighbors that needs to be considered while making prediction.

Step 2: Calculating distance

- To measure the similarity between target and training data points, Euclidean distance is used. Distance is calculated between each of the data points in the dataset and target point.

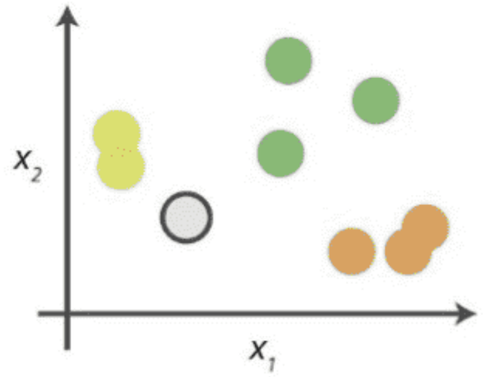
Step 3: Finding Nearest Neighbors

- The k data points with the smallest distances to the target point are the nearest neighbors.

Step 4: Voting for Classification or Taking Average for Regression

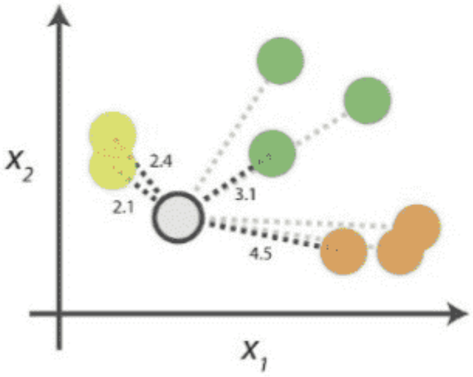
- In the classification problem, the class labels of K-nearest neighbors are determined by performing majority voting. The class with the most occurrences among the neighbors becomes the predicted class for the target data point.

0. Look at the data



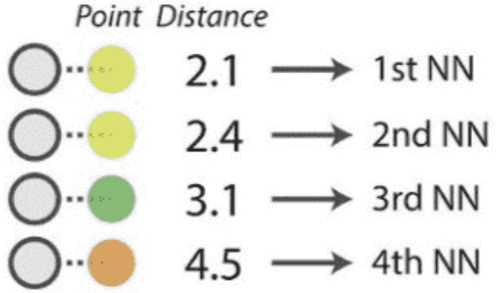
Say you want to classify the grey point into a class. Here, there are three potential classes - lime green, green and orange.

1. Calculate distances



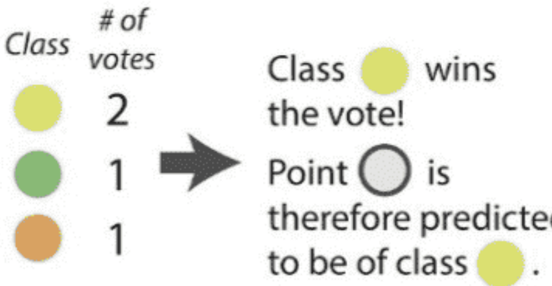
Start by calculating the distances betwee the grey point and all other points.

2. Find neighbours



Next, find the nearest neighbours by ranking points by increasing distance. The nearest neighbours (NNs) of the grey point are the ones closest in dataspace.

3. Vote on labels



Vote on the predicted class labels based on the classes of the k nearest neigh-bours. Here, the labels were predicted based on the k=3 nearest neighbours.

DECISION TREE

A decision tree is a flowchart-like structure used to make decisions or predictions. It consists of nodes representing decisions or tests on attributes, branches representing the outcome of these decisions, and leaf nodes representing final outcomes or predictions. Each internal node corresponds to a test on an attribute, each branch corresponds to the result of the test, and each leaf node corresponds to a class label or a continuous value.

Structure of a Decision Tree

Root Node

Represents the entire dataset and the initial decision to be made.

Internal Nodes

Represent decisions or tests on attributes. Each internal node has one or more branches.

Branches

Represent the outcome of a decision or test, leading to another node.

Leaf Nodes

Represent the final decision or prediction. No further splits occur at these nodes.

HOW DECISION TREES WORK?

1).Selecting the Best Attribute

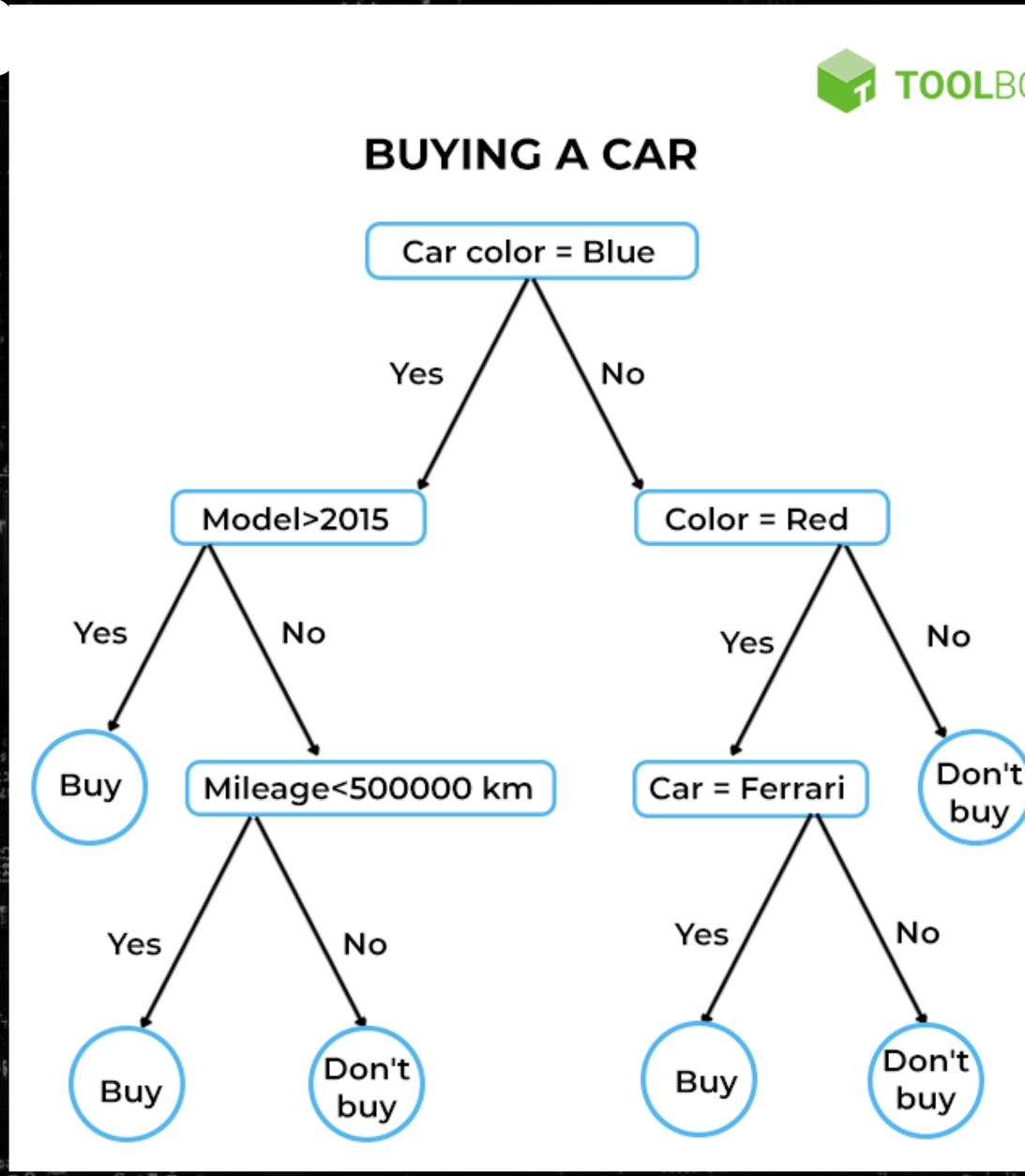
Using a metric like entropy, or information gain, the best attribute to split the data is selected.

2).Internal Nodes

Represent decisions or tests on attributes. Each internal node has one or more branches.

3).Repeating the Process:

The process is repeated recursively for each subset, creating a new internal node or leaf node until a stopping criterion is met



SUPPORT VECTOR MACHINE

Support Vector Machine" (SVM) is a supervised machine learning algorithm which can be used for both classification or regression challenges. However, it is mostly used in classification problems.

1

HYPERPLANE

A decision boundary that divides the feature space into two regions, each corresponding to a different class.

2

SUPPORT VECTORS

The data points closest to the hyperplane that influence its position.

3

MARGIN

The distance between the hyperplane and the nearest data points (support vectors).

1

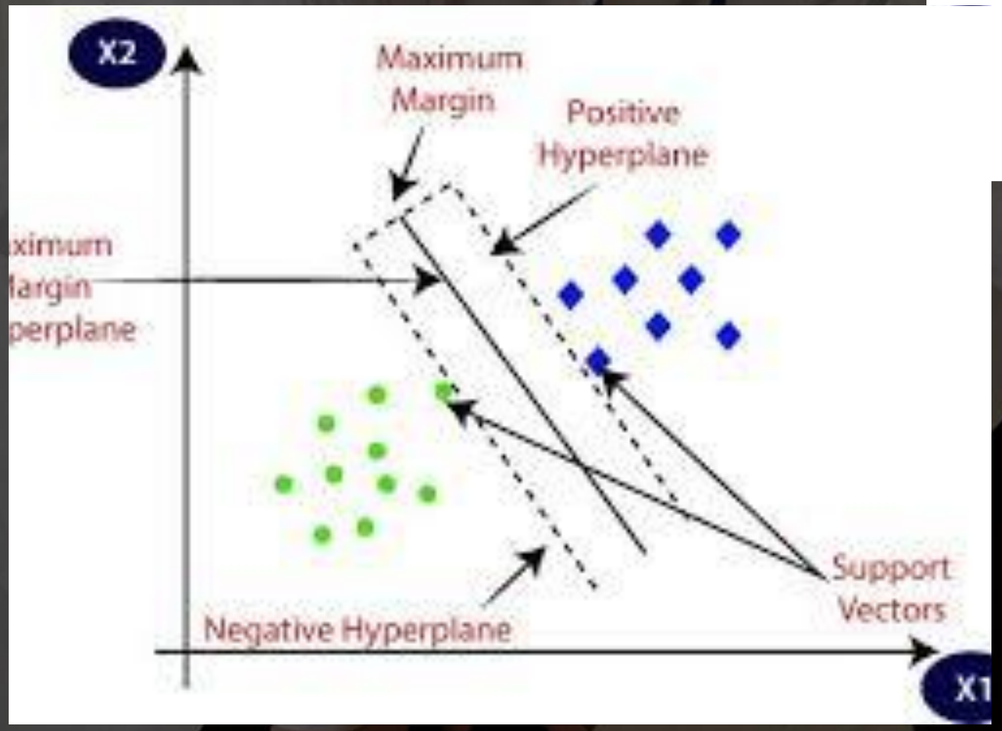
FEATURE SPACE AND HYPERPLANE

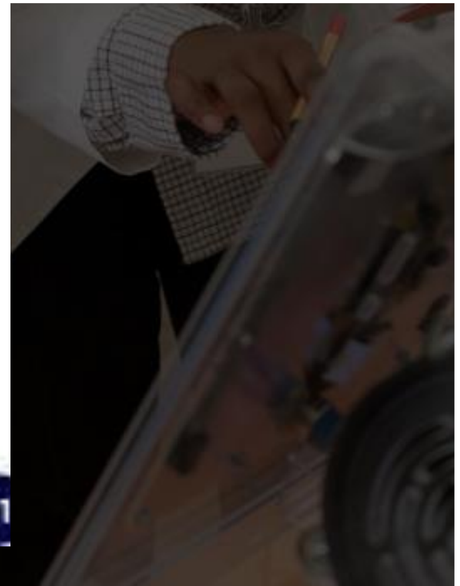
- In a given feature space, SVM tries to find a hyperplane that best separates the data points into different classes. For two-dimensional data, this is a line; for three-dimensional data, it is a plane; and for higher dimensions, it is a hyperplane.

2

SUPPORT VECTORS

The data points that are closest to the hyperplane are called support vectors. These points are critical as they define the position and orientation of the hyperplane.





3

MAXIMIZING THE MARGIN:

VM aims to maximize the margin between the hyperplane and the support vectors. The margin is the distance between the hyperplane and the nearest data points from each class. A larger margin implies better generalization of the classifier.

LONG SHORT TERM MEMORY

Long Short Term Memory networks – usually just called “LSTMs” – are a special kind of RNN, capable of learning long-term dependencies. They were introduced by Hochreiter & Schmidhuber (1997), and were refined and popularized by many people in following work. They work tremendously well on a large variety of problems, and are now widely used.

LSTMs introduce a concept of memory cells and gates to control the flow of information. This allows them to capture and maintain information over extended periods, making them highly effective for sequential data.

Forget Gate

This gate determines which information from the previous cell state should be discarded. It takes the previous state and the current input and outputs a value between 0 and 1 for each number in the cell state . A value of 0 means "completely forget" and a value of 1 means "completely keep".

Input Gate

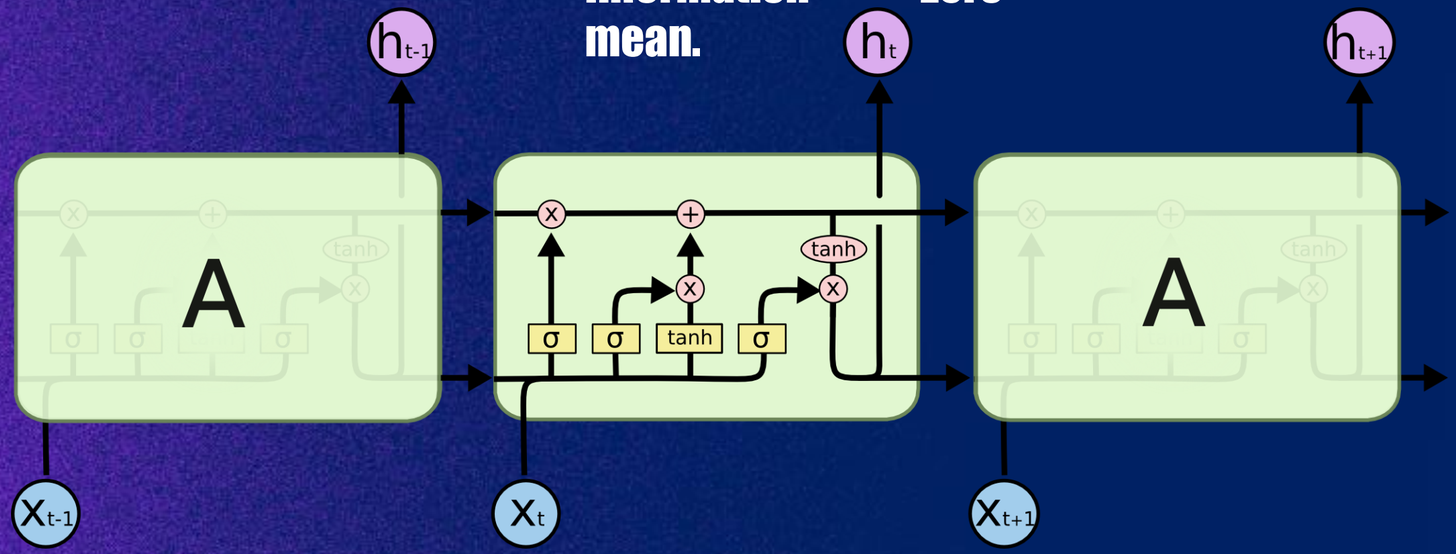
This gate decides which values from the input will update the cell state.

Input Modulation Gate

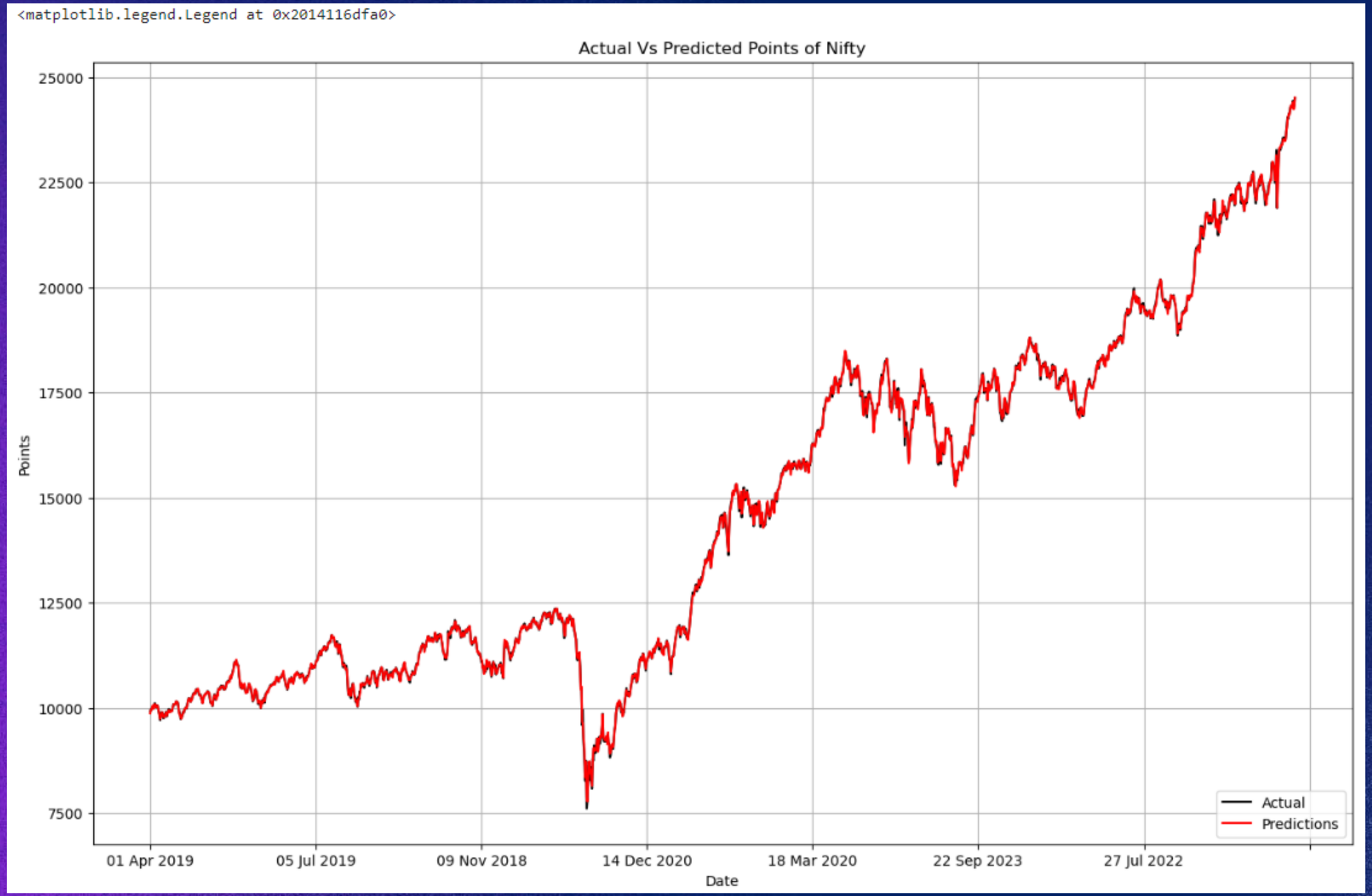
It is used to modulate the information that the Input gate will write onto the Internal State Cell by adding non-linearity to the information and making the information Zero-mean.

Output Gate

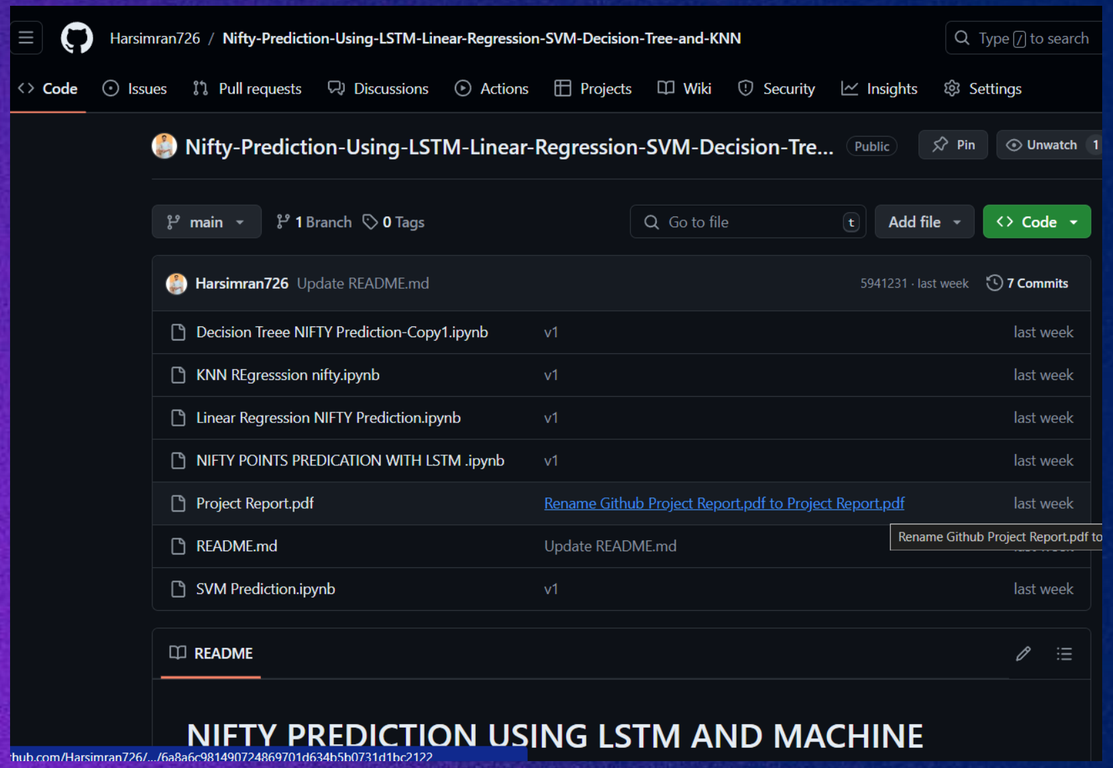
The output gate is like a filter. It takes the information from the LSTM's memory and carefully selects what to show to the outside world.



<matplotlib.legend.Legend at 0x2014116dfa0>



Algorithm	Strengths	Weaknesses	Suitable for
Linear Regression	Simple, interpretable, efficient	Assumes linear relationship, sensitive to outliers	Long-term trends
KNN Regression	Captures local patterns, non-parametric	Computationally expensive, sensitive to k	Short-term patterns, non-linear relationships
Decision Tree Regression	Interpretable, handles non-linearity	Prone to overfitting, unstable	Non-linear relationships, feature importance
Support Vector Regression (SVR)	Handles outliers, effective for non-linearity	Computationally expensive, sensitive to hyperparameters	Non-linear relationships, high-dimensional data
LSTM	Captures long-term dependencies, handles complex patterns	Complex, computationally intensive	Time series data, capturing trends and seasonality


Harsimran726 / Nifty-Prediction-Using-LSTM-Linear-Regression-SVM-Decision-Tree-and-KNN
Type / to search
Code
Issues
Pull requests
Discussions
Actions
Projects
Wiki
Security
Insights
Settings
Nifty-Prediction-Using-LSTM-Linear-Regression-SVM-Decision-Tre...
Public
Pin
Unwatch 1
main
1 Branch
0 Tags
Go to file
t
Add file
Code
Harsimran726 Update README.md
5941231 · last week
7 Commits
Decision Treee NIFTY Prediction-Copy1.ipynb v1 last week
KNN REgresssion nifty.ipynb v1 last week
Linear Regression NIFTY Prediction.ipynb v1 last week
NIFTY POINTS PREDICATION WITH LSTM .ipynb v1 last week
Project Report.pdf Rename Github Project Report.pdf to Project Report.pdf last week
Rename Github Project Report.pdf to
README.md Update README.md
SVM Prediction.ipynb v1 last week
README
NIFTY PREDICTION USING LSTM AND MACHINE

Github


THANK YOU