PART 1

1. Regression Models

Regression Models is a process of determining a relationship between 1 or more independent variables and one dependent (output variable).

Ex - predicting the height of a person given the age of the person

- Predicting the price of the car given the car model ,year of manufacturing , mileage , engine capacity, etc.

- Based on the type of functions used to represent the relationship between the dependent (output variable) and independent variables. The regression models are categorised into four types.
  1. simple linear regression
  2. multiple linear equation
  3. polynomial regression and
  4. Logistic regression

1. **Linear Regression:**
   When there is only 1 independent variable x. if the relationship between X independent variable and y dependent variable is modelled by the relation
   y = a + bx
   then the regression model is called SLR model.

   - **Advantages:** Simple, interpretable, computationally efficient.
   - **Disadvantages:** Poor performance on non-linear data, sensitive to outliers.
   - **Applications:** Predicting house prices, sales forecasting, stock price prediction.

- when there are multiple independent variables like X1, X2 ,………Xn . if the relationship between Independent variable X and dependent variable y is modded by the relation

  y = a0+a1*x1+a2*x2+……..+an*xn

  then the regression model is called MLR model.

2. **Polynomial Regression:**
   When there is only 1 independent variable x. if the relationship between X independent variable and dependent variable y is modelled by the relation

   y = a0+a1*x+a2*x^2+……..+an*x^n

- For some **(+ve) integer n>1** , then it will become PR. Else if **n = 1** then it is SLR model.
- **Assumptions:** Same as linear regression but allows for non-linearity.
- **Advantages:** Can model non-linear relationships.

- **Disadvantages:** Prone to overfitting with high-degree polynomials.
- **Applications:** Growth curves, physics simulations, financial modelling.

3. **<u>Support Vector Regression (SVR):</u>**
A type of support vector machine (SVM) that is used for regression tasks. It tries to find a function that best predicts the continuous output value for a given input value. SVR can use both linear and non-linear kernels.
    - We solve supervised regression problem
    - Ex: when we want to find price of house.

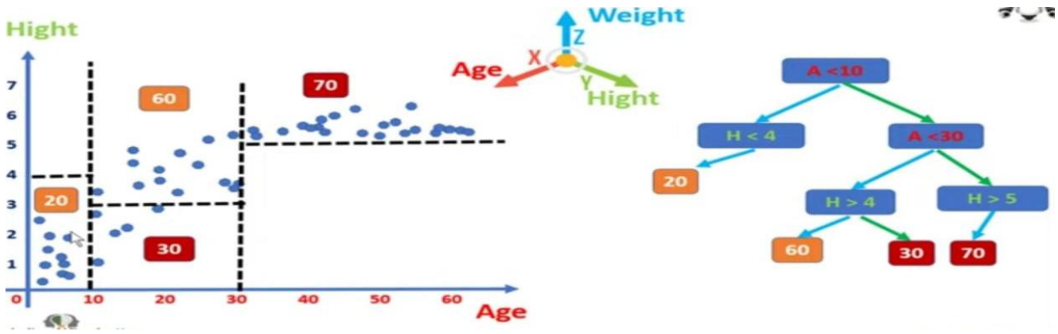
- Important parameters are:
    - Hyperplane [w^T .x +b=0]
    - Margine [w^T .x +b= -E] , [w^T .x +b= E]
    - Threshold value(E0)
- So The main focuses or target of SVR is to fit the feature / instances inside the margine line using E.
- **Principle:** Uses a margin of tolerance (epsilon) around a hyperplane to fit data.
- **Assumptions:** Works well with high-dimensional data, assumes data is noise-free.
- **Advantages:** Effective in high-dimensional spaces, robust to outliers.
- **Disadvantages:** Computationally expensive, sensitive to hyperparameters.
- **Applications:** Time series forecasting, financial modeling, biological data analysis.

4. **<u>Decision Tree Regression:</u>**
    - Decision tree are versatile machine learning algorithm that can perform both classification and regression task and even multi output tasks.
    - Scikit-learn use is the classification and regression tree to train decision tree.
    - In **decision tree regression** we have to split the data , by using MSE .
    - we always try to minimise the mean square error to split the data until unless we get the leaf node.
    - In Decision tree regression after applying MSE the target value we get we take the average of those all values and this average value will be average of all the value in the instances in that particularly node.

Mathematical Formula

CART cost function for Regression

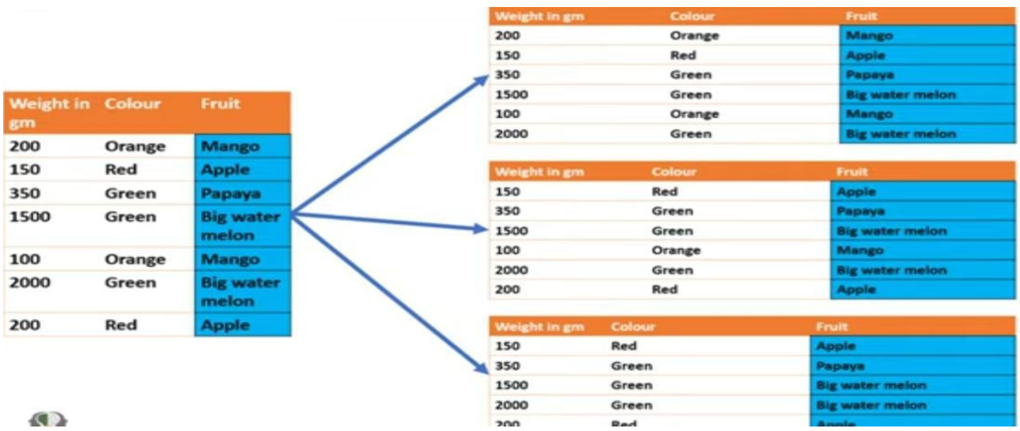
$$J(k, t_k) = \frac{m_{\text{left}}}{m}\text{MSE}_{\text{left}} + \frac{m_{\text{right}}}{m}\text{MSE}_{\text{right}} \quad \text{where} \begin{cases} \text{MSE}_{\text{node}} = \sum_{i \in \text{node}} \left(\hat{y}_{\text{node}} - y^{(i)}\right)^2 \\ \hat{y}_{\text{node}} = \frac{1}{m_{\text{node}}} \sum_{i \in \text{node}} y^{(i)} \end{cases}$$

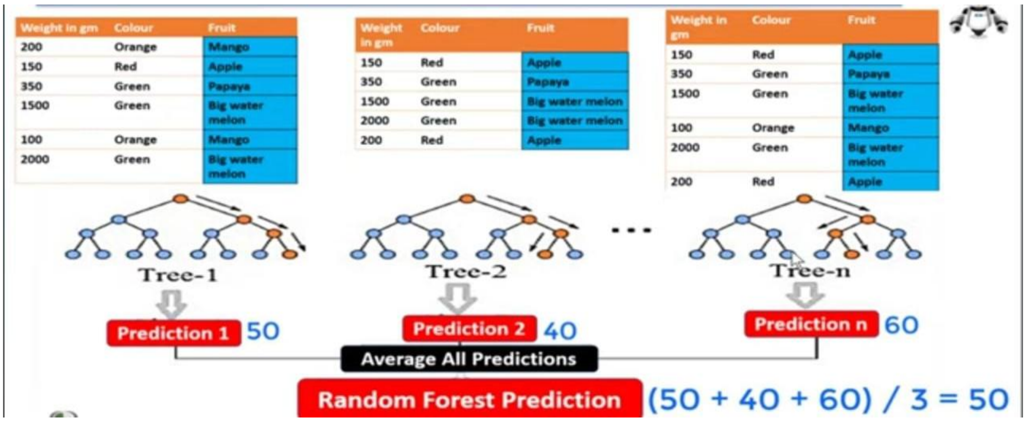


- **Principle:** Splits data into branches based on feature values to make predictions.
- **Assumptions:** No strict assumptions about data distribution.
- **Advantages:** Easy to interpret, handles non-linear data well.
- **Disadvantages:** Prone to overfitting, sensitive to small changes in data.
- **Applications:** Customer segmentation, price prediction, healthcare diagnostics.

5. **<u>Random Forest Regression</u>**

- **Principle:** Uses an ensemble of decision trees to improve accuracy and reduce overfitting.
- **Assumptions:** No strict assumptions about data distribution.
- **Advantages:** Reduces overfitting, handles missing values well, works with large datasets.
- **Disadvantages:** Computationally expensive, less interpretable.
- **Suitable Applications:** Weather prediction, fraud detection, medical diagnosis.
- Random forest or random decision for rest are ensemble learning method for classification and regression by making multiple decision tree using random samples from training data.





6. **Ridge Regression:**

- **Principle:** Linear regression with L2 regularization to reduce multicollinearity effects.
- **Assumptions:** Same as linear regression, but less sensitive to multicollinearity.
- **Advantages:** Reduces overfitting, improves generalization.
- **Disadvantages:** Less interpretable coefficients, not effective for sparse data.
- **Applications:** Economic forecasting, text analysis, medical research.

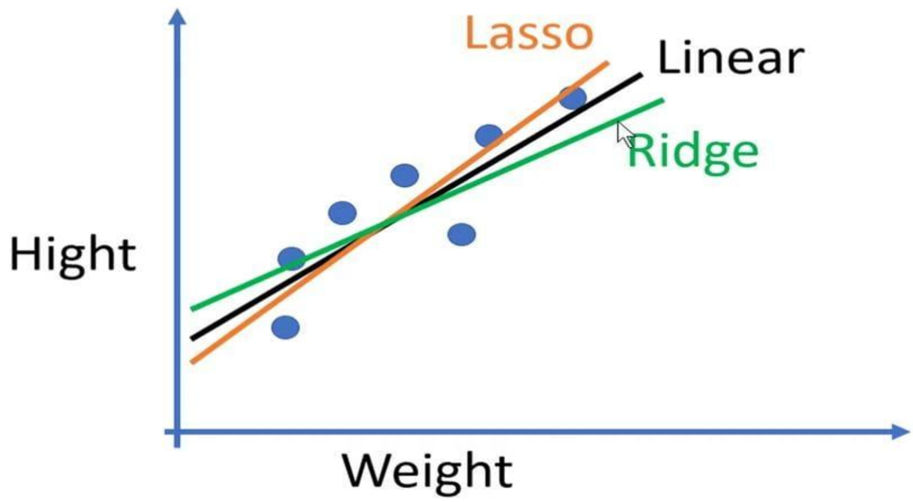
$$\text{Ridge} = \sum_{i=1}^{M}(y_i - \hat{y}_i)^2 = \sum_{i=1}^{M}\left(y_i - \sum_{j=0}^{p} w_j \times x_{ij}\right)^2 + \lambda \sum_{j=0}^{p} w_j^2$$

7. **Lasso Regression**

- **Principle:** Linear regression with L1 regularization to shrink irrelevant feature coefficients to zero.
- **Assumptions:** Similar to Ridge Regression but promotes feature selection.
- **Advantages:** Performs feature selection, reduces overfitting.
- **Disadvantages:** Can eliminate important features, sensitive to lambda hyperparameter.
- **Applications:** Gene selection in bioinformatics, credit risk assessment, high-dimensional datasets.

$$\text{Lasso} = \sum_{i=1}^{M}(y_i - \hat{y}_i)^2 = \sum_{i=1}^{M}\left(y_i - \sum_{j=0}^{p} w_j \times x_{ij}\right)^2 + \lambda \sum_{j=0}^{p} |w_j|$$

- When a linear regression fit the model its leads to over fitting in order to generalize the linear regression, we use the technique of regularisation such as L1 and L2 is regulation.



A comparative table summarizing the key features of each model.

Model	Principle	Assumptions	Advantages	Disadvantages	Suitable Applications
Linear Regression	Finds best-fit line	Linearity, normality	Simple, interpretable	Poor on non-linear data	Sales forecasting, stock prices
Polynomial Regression	Adds polynomial terms	Non-linearity allowed	Models non-linear data	Overfits with high-degree	Growth curves, physics models
SVR	Margin-based regression	Works with high dimensions	Handles outliers, good accuracy	Computationally expensive	Time series, finance
Decision Tree Regression	Tree-based splits	No assumptions	Interpretable, non-linear data	Overfits, sensitive to changes	Price prediction, healthcare
Random Forest Regression	Ensemble of trees	No assumptions	Reduces overfitting, robust	Computationally heavy	Weather prediction, fraud detection

Ridge Regression	Linear regression + L2 penalty	Reduces multicollinearity	Improves generalization	Harder to interpret coefficients	Economic forecasting, medical research
Lasso Regression	Linear regression + L1 penalty	Feature selection	Selects important features	May remove key features	Credit risk, high-dimensional datasets

PART 2

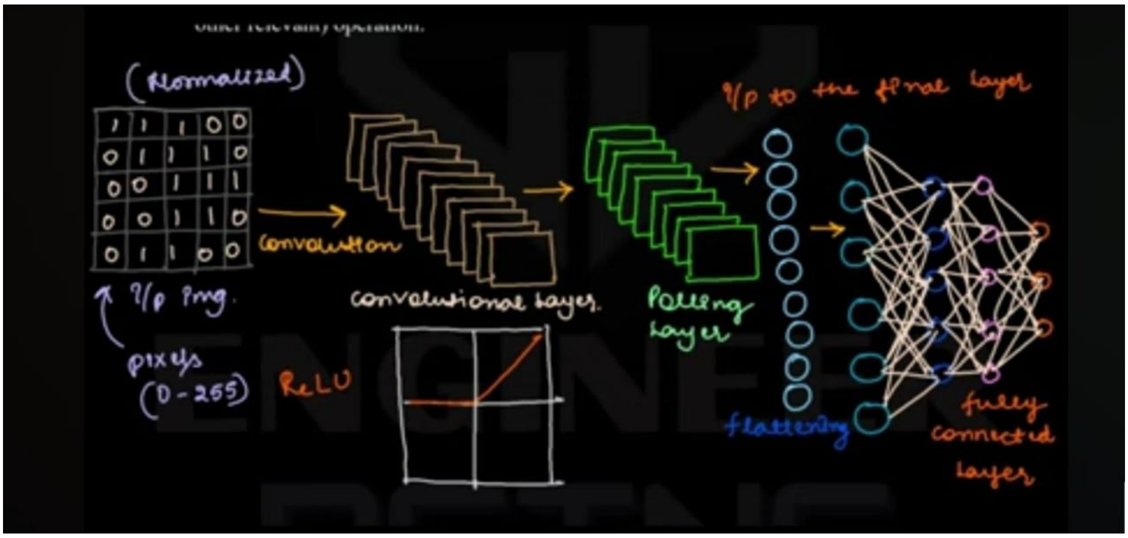
1. **Convolutional Neural Networks (CNNs):** CNNs are designed for image processing and classification tasks.
   - They consist of several key components:

   1.<u>Convolutional Layers:</u>
   - These layers apply filters (kernels) to extract features such as edges, textures, and patterns.
   - Each filter slides over the input image and creates a feature map.

   2. <u>Pooling Layers:</u>
   - Used to reduce the spatial dimensions of feature maps, helping to retain important features while reducing computational load.
   - Max Pooling is commonly used, which selects the highest value from a windowed region.

   3. <u>Activation Functions:</u>
   - Non-linear functions applied to introduce complexity in the network.
   - The ReLU (Rectified Linear Unit) function is widely used to avoid vanishing gradients and improve training.

   4. <u>Fully Connected Layers:</u>
   - After convolutional and pooling layers, fully connected layers process extracted features to make predictions.
   - The final layer often uses a softmax activation function for multi-class classification.



2. **<u>Long Short-Term Memory Networks (LSTMs):</u>**

   - LSTMs are a type of Recurrent Neural Network (RNN) designed to handle **sequential data** efficiently.

- Unlike traditional RNNs, LSTMs solve the problem of **vanishing gradients**, making them suitable for long-range dependencies in data, such as **time-series forecasting, speech recognition, and text generation**.

key Components **of LSTM Architecture:** consists of multiple gates to control the flow of information.

1. <u>Memory Cell</u>

- The **core component** of an LSTM network that retains information over multiple time steps.
- Helps store and update long-term dependencies in the sequence.

- **2. <u>Gates in LSTM Cells</u>**

- Forget Gate
- Input Gate
- Output Gate
- Cell State

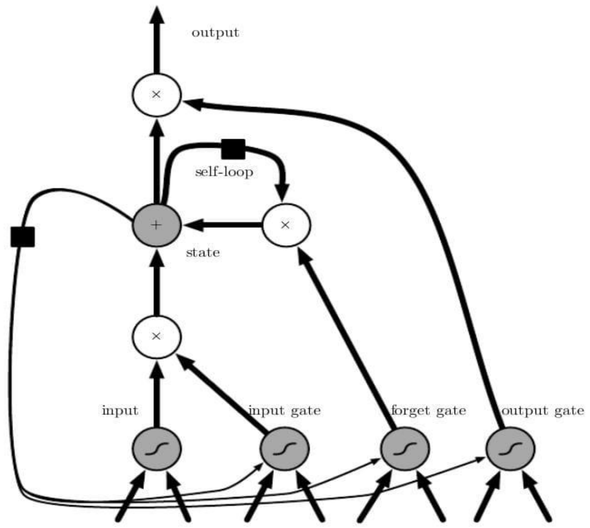


Figure 10.16: Block diagram of the LSTM recurrent network "cell." Cells are connected

* LSTM :

$$f_i^{(t)} = \sigma\left(b_i^f + \sum_j U_{i,j}^f x_j^{(t)} + \sum_j W_{i,j}^f h^{(t-1)}\right),$$

where, $x^{(t)}$ is the current I/P vector & $h^{(t)}$ is the current hidden layer vector containing the O/P of all the LSTM cells & b^f, U^f, W^f are respectively biases, I/P weights & recurrent weights for the forget gates.

→ The LSTM cell internal state is thus updated as follows, but with a conditional self-loop weight $f_i^{(t)}$:

$$s_i^{(t)} = f_i^{(t)} s_i^{(t-1)} + g_i^{(t)} \sigma\left(b_i + \sum_j U_{i,j} x_j^{(t)} + \sum_j W_{i,j} h_j^{(t-1)}\right),$$

where b, U & W respectively denote the biases, I/P weight & recurrent weights into the LSTM cell.

⇒ The External I/P gate unit $g_i^{(t)}$ is computed similarly to the forget gate but with its own parameters

$$g_i^{(t)} = \sigma\left(b_i^g + \sum_j U_{i,j}^g x_j^{(t)} + \sum_j W_{i,j}^g h_j^{(t-1)}\right)$$

⇒ The O/P $h_i^{(t)}$ of the LSTM cell can also be shut off, via the O/P gate $q_i^{(t)}$ which also uses a sigmoid unit for gating.

$$h_i^{(t)} = \tanh\left(s_i^{(t)}\right) q_i^{(t)}$$

$$q_i^{(t)} = \sigma\left(b_i^o + \sum_j U_{i,j}^o x_j^{(t)} + \sum_j W_{i,j}^o h_j^{(t-1)}\right)$$

which has parameters b^o, U^o, W^o for its biases, I/P weight & recurrent weights respectively. //