INFO 609 -Final Project

“Predictive Modeling of University Student GPA Using Machine Learning Techniques”

(Research-Based Project)

Presented by
1.Vinothkumar Sureshkumar
2.Pradeep Jondhale
Student’s of MSISML,College Of Information Science.

Faculty Advisor:
Prof.Dr. Nitika Sharma

Project Mentor
Prof.Dr Betul Czerkawaski

THE UNIVERSITY OF ARIZONA

.Agenda/Contents

1.Project Overview & Motivation

Predictive Analytics in Education

- Traditional student outcome predictions relied on **manual data collection**, **siloed information**, and **slow intervention**.
- With the rise of **AI - machine learning and Deep learning** , institutions now analyze large volumes of **historical and real-time data** efficiently.

- **Predictive analytics** helps:
  - -Identify **at-risk students early**
  - -Forecast academic challenges
  - -Enable **personalized academic support**
  - -Improve **retention** and **overall student success**
- Universities use these insights to **optimize resources**, strengthen **enrollment strategies**, and design more **responsive, data-driven policies**.

- This project demonstrates **how predictive analytics converts raw data into actionable insights, supporting both individual student success and institutional decision-making**.

2. Research Problem/Objective

- How can machine learning techniques be used to accurately predict university students' cumulative GPA using academic, demographic, and enrollment features?

- Can an ANN model (Neural Network based Algorithm) accurately predict **Term GPA** from demographic and academic attributes?

- How do **ANN** performance metrics **compare** with **classical regression** methods?

- Which features most influence model output, and how can these insights support academic advising?

3. Dataset Description

- **Source:** University Student Academic Dataset
- **Total Records:** ~(66,429 * 15)
- **Target Variable:** *Cumulative GPA*
- **Key Predictor Categories:**
    1. **Demographics:** Age, Gender, First-Generation Flag
    2. **Academic Info:** Term GPA, Number of Classes Enrolled, Academic Level
    3. **Institutional Info:** College, Academic Career
    4. **Enrollment Status:** Full-time/Part-time

Key Predictor/Feature/Independent Variables Categories:

- Demographics: Age, Gender, First-Generation Flag
- Academic Info: Term GPA, Number of Classes Enrolled, Academic Level
- Institutional Info: College, Academic Career
- Enrollment Status: Full-time/Part-time

TermGPA

4.Data Preprocessing

- **Removed inconsistent records**
  –” 'FakeIdentifier'

- **Imputed nulls with mean (no significant outliers)**
  –” TermGPA”,” CumulativeGPA” (2-Variables)

- **One-hot encoding for Nominal categorical variables-**
  -'Gender’ (3 Levels) (2 variable),
  -'PrimaryMilitaryAffiliation’ (8 Levels), (7 Levels),
  - 'College’ -(21 Levels), (20 variable)
  -"UAFullTimePartTime“ -(2 levels), (1 variable)
  - 'FirstGenerationFlag’-(2 levels), (1 variable)

- Converted ordinal categorical data
  -"AcademicYear“(6 levels), (1 variable)
  -"AcademicLevelEndofTerm“(6 Levels)(1 Variable)
  -"AcademicCareer“(3 Levels), (1 variable)

- **Standardized numeric variables**
  –” Number of Classes Enrolled”, “Age”,” Units Passed included in GPA” ,”Units Passed not included in GPA”, ”Term GPA”,
  (5 Variables) =Total=41 Varaibles= 40 features+ 1 label

- **Train/test split:**
  - 80% train / 20% test

5. Feature Engineering

- **Transformation and encoding**:
  ✓ Scaling (StandardScaler)
  ✓ One-hot encoding for nominal features

- **Removed deature-**
  FakeIdentifier

- **Checked multicollinearity using:**
  ✓ Correlation Matrix
  .However,VIF is not applicable as **VIF requires purely numeric input.**

- **Used Ridge Regression to mitigate multicollinearity**

6.1. Modeling Techniques-Linear Regression(OLS)

- Model1- Classical Model -OLS-Linear Regression-

  -Baseline model for interpretability.

  -For a dependent variable y and multiple predictors $x_1, x_2, \ldots, x_p$:

$$\boldsymbol{y} = \boldsymbol{\beta_0} + \boldsymbol{\beta_1 x_1} + \boldsymbol{\beta_2 x_2} + \cdots + \boldsymbol{\beta_p x_p} + \boldsymbol{\varepsilon}$$

Where: y = outcome (dependent variable)

$\boldsymbol{\beta_0}$ = intercept ,

$\boldsymbol{\beta_1}, \boldsymbol{\beta_2}, \ldots, \boldsymbol{\beta_p}$ = regression coefficients

$\boldsymbol{x_1}, \boldsymbol{x_2}, \ldots, \boldsymbol{x_p}$ = predictor variables

$\boldsymbol{\varepsilon}$ = error term (unexplained variation)

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}$$

Where:

$\mathbf{y} = n \times 1$ vector of outcomes

$\mathbf{X} = n \times (p+1)$ design matrix (with first column of 1's for intercept)

$\boldsymbol{\beta} = (p+1) \times 1$ vector of coefficients

ε = error vector

- Assumptions Of OLS Linear Regression model:

  -The relationship between predictors and the dependent variable is **linear in parameters**.

  -The sample $(X_i, Y_i)$is drawn **randomly and independently** from the population.

  -Predictors must **not** be perfectly linearly dependent

  -No **correlation between errors and predictor , Error are iid ~ N(0,constant variance)**

  - $\hat{\boldsymbol{\beta}} \sim \boldsymbol{N}\left(\boldsymbol{\beta}, \boldsymbol{\sigma}^2 (\boldsymbol{X}'\boldsymbol{X})^{-1}\right)$

6.1. Modeling Techniques-Linear Regression(OLS)

- The Mean Squared Error to minimize: (Objective function)

$$\mathrm{MSE}(\boldsymbol{\beta}) = \frac{1}{n}\|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|^2$$

- Equivalent to minimizing:

$$\mathrm{SSE} = (\mathbf{y} - \mathbf{X}\boldsymbol{\beta})^T(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})$$

- Solution to Minimising MSE is,

$$\boldsymbol{\beta\hat{o}ls} = \left(\boldsymbol{X^T X}\right)^{-1}\boldsymbol{X^T y}$$

- **Problem with OLS.**

  If Some predictors are **highly correlated ,** The $X^T X$ matrix becomes **nearly singular .**
  Ordinary Least Squares (OLS) estimates become **unstable**, with VERY large variance and unreliable p-values.

6.1. Modeling Techniques-Linear Regression(OLS)

Before Fitting OLS regression model:

1) There found Multicollinearity in the predictor variables as below,

Predictor 1	Correlation coefficient	predictor 2
UnitsPassedincludedinGPA	0.714074	NumberofClassesEnrolled
AcademicLevelEndofTerm	-0.745911	AcademicCareer_Undergraduate
College_James E Rogers College of Law	0.997497	AcademicCareer_Law
UAFullTimePartTime_P	-0.712439	NumberofClassesEnrolled
AcademicCareer_Law	0.997497	College_James E Rogers College of Law
AcademicCareer_Undergraduate	-0.745911	AcademicLevelEndofTerm

- Variable to Drop
  -College_James E Rogers College of Law
  -UnitsPassedincludedinGPA
  -AcademicCareer_Undergraduate
  -UAFullTimePartTime_P

6.2 Modeling Techniques-Linear Regression(Ridge)

- Model 2- Bayesian Model –Ridge Regression.

  -model for regularization

  -For a dependent variable y and multiple predictors $x_1, x_2, \dots, x_p$:

$$\boldsymbol{y} = \boldsymbol{\beta_0} + \boldsymbol{\beta_1 x_1} + \boldsymbol{\beta_2 x_2} + \cdots + \boldsymbol{\beta_p x_p} + \boldsymbol{\varepsilon}$$

Where: y = outcome (dependent variable)

$\boldsymbol{\beta_0}$ = intercept ,

$\boldsymbol{\beta_1}, \boldsymbol{\beta_2}, \dots, \boldsymbol{\beta_p}$ = regression coefficients

$\boldsymbol{x_1}, \boldsymbol{x_2}, \dots, \boldsymbol{x_p}$ = predictor variables

$\boldsymbol{\varepsilon}$ = error term (unexplained variation)

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}$$

Where:

$\mathbf{y} = n \times 1$ vector of outcomes

$\mathbf{X} = n \times (p+1)$ design matrix (with first column of 1's for intercept)

$\boldsymbol{\beta} = (p+1) \times 1$ vector of coefficients

ε = error vector

6.2 Modeling Techniques-Linear Regression(Ridge)

1. **Assume we are doing linear regression:**

$$\mathbf{y} = \mathbf{X}\beta + \varepsilon, \varepsilon \sim \mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I})$$

2. **Likelihood Function (Gaussian)**

$$p(\mathbf{y} \mid \boldsymbol{\beta}, \mathbf{X}) = \mathcal{N}(\mathbf{y} \mid \mathbf{X}\boldsymbol{\beta}, \sigma^2\mathbf{I})$$

This gives the log-likelihood (up to constant):

$$\log p(\mathbf{y} \mid \boldsymbol{\beta}) = -\frac{1}{2\sigma^2}\|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|^2 + \text{const}$$

3. **Prior Distribution (Normal)**

Assume a zero-mean normal prior on β :

$$p(\boldsymbol{\beta}) = \mathcal{N}(\boldsymbol{\beta} \mid \mathbf{0}, \tau^2\mathbf{I})$$

Then log-prior is:

$$\log p(\beta) = -\frac{1}{2\tau^2}\|\beta\|^2 + \text{const}$$

3. **Posterior Distribution (via Bayes' Theorem)**

$$p(\boldsymbol{\beta} \mid \mathbf{y}, \mathbf{X}) \propto p(\mathbf{y} \mid \boldsymbol{\beta}) \cdot p(\boldsymbol{\beta})$$

$$\Rightarrow \log p(\boldsymbol{\beta} \mid \mathbf{y}) = -\frac{1}{2\sigma^2}\|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|^2 - \frac{1}{2\tau^2}\|\boldsymbol{\beta}\|^2 + \text{const}$$

Equivalently $\Rightarrow$ $\log p(\boldsymbol{\beta} \mid \mathbf{y}) = \frac{1}{2n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 + \lambda\sum_{j=1}^{p}\beta_j^2$ --Objective Function

$$\text{where}, \lambda = \sigma^2/\tau^2$$

This is equivalent to minimizing the negative log-posterior, i.e., MAP estimate:

6.2 Modeling Techniques-Linear Regression(Ridge)

- **Ridge estimates are:**

$$\hat{\boldsymbol{\beta}}_{\textbf{ridge}} = (\boldsymbol{X}^{\top}\boldsymbol{X} + \boldsymbol{\lambda I})^{-1}\boldsymbol{X}^{\top}\boldsymbol{y}$$

Where $\lambda > 0$ is the penalty parameter.

- **WHY THIS FIXES MULTICOLLINEARITY?**

1 -Because adding λ to diagonal $\Longrightarrow$ makes the matrix invertible

-$X^{\top}X + \lambda I$ is always invertible, even when $X^{\top}X$ is near singular.

- No exploding coefficients

-Numerically stable estimation.

2 – Because it shrinks correlated predictors

When predictors are correlated, OLS distributes weights arbitrarily.

Ridge shrinks them toward zero together, preventing instability.

6.3 Modeling Techniques-Linear Regression(Lasso)

- Model 3- Bayesian Model –**Lasso Regression**-

  -model for regularization

  -For a dependent variable y and multiple predictors $x_1, x_2, \dots, x_p$:

$$\boldsymbol{y} = \boldsymbol{\beta}_0 + \boldsymbol{\beta}_1\boldsymbol{x}_1 + \boldsymbol{\beta}_2\boldsymbol{x}_2 + \cdots + \boldsymbol{\beta}_p\boldsymbol{x}_p + \boldsymbol{\varepsilon}$$

Where:$y =$ outcome (dependent variable)

$\boldsymbol{\beta}_0$= intercept ,$\boldsymbol{\beta}_2, \dots, \boldsymbol{\beta}_p$ = regression coefficients

$\boldsymbol{x}_1, \boldsymbol{x}_2, \dots, \boldsymbol{x}_p$ = predictor variables

$\boldsymbol{\varepsilon}$ = error term (unexplained variation)

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}$$

Where: $\mathbf{y} = n \times 1$ vector of outcomes

$\mathbf{X} = n \times (p+1)$ design matrix (with first column of 1's for intercept)

$\boldsymbol{\beta} = (p+1) \times 1$ vector of coefficients

$\varepsilon =$ error vector

6.3 Modeling Techniques-Linear Regression(Lasso)

. **Assume we are doing linear regression:**

$$\mathbf{y} = \mathbf{X}\beta + \varepsilon, \varepsilon \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I})$$

2.**Likelihood Function (Gaussian)**

$$p(\mathbf{y} \mid \boldsymbol{\beta}, \mathbf{X}) = \mathcal{N}(\mathbf{y} \mid \mathbf{X}\boldsymbol{\beta}, \sigma^2 \mathbf{I})$$

This gives the log-likelihood (up to constant):

$$\log p(\mathbf{y} \mid \boldsymbol{\beta}) = -\frac{1}{2\sigma^2} \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|^2 + \text{const}$$

3.**Prior Distribution (Normal)**

Assume a zero-mean normal prior on β : $\beta_j \sim \text{Laplace}(0, b)$, $P(B) = \frac{1}{2b} \exp\left(. \rightarrow \frac{|\beta_j|}{b}\right)$

Then log-prior is: $\mathbf{log}\, \boldsymbol{p}(\boldsymbol{\beta}) = -\frac{\mathbf{1}}{\boldsymbol{b}} \parallel \boldsymbol{\beta} \parallel_{\mathbf{1}}$ **+const**

3. **Posterior Distribution (via Bayes' Theorem)**

$$\boldsymbol{arg\, \min_{\beta}}\{\parallel \boldsymbol{y} - \boldsymbol{X\beta} \parallel^{\mathbf{2}} + \boldsymbol{\lambda} \parallel \boldsymbol{\beta} \parallel_{\mathbf{1}}\}$$ --Objective Function

:

6.3 Modeling Techniques-Linear Regression(Lasso)

- **Lasso estimates are:**

$$\hat{\boldsymbol{\beta}}_{\mathbf{lasso}} = \arg\min_{\boldsymbol{\beta}}\{\| \mathbf{y} - \mathbf{X}\boldsymbol{\beta} \|^2 + \lambda \| \boldsymbol{\beta} \|_1\}$$

Where $\lambda > 0$ is the penalty parameter., $\boldsymbol{\lambda = \sigma^2 / b}$

- **WHY THIS FIXES MULTICOLLINEARITY?**

1 -Because adding λ to diagonal $\Longrightarrow$ makes the matrix invertible

-$X^\top X + \lambda I$ is always invertible, even when $X^\top X$ is near singular.

- No exploding coefficients

-Numerically stable estimation.

2 – LASSO sets weights exactly to zero

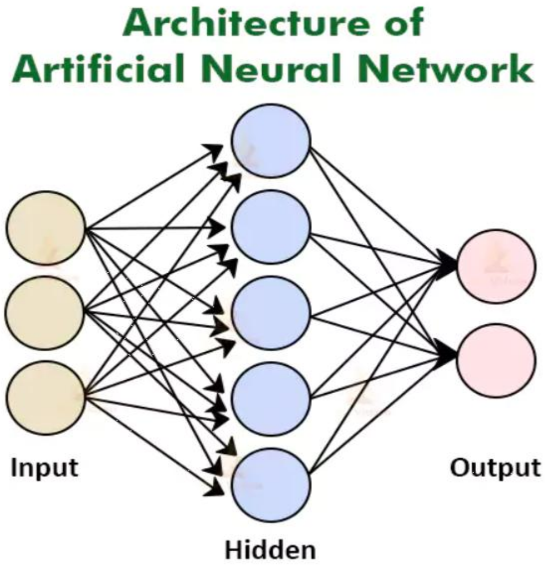
6.4 Modeling Techniques- (ANN)

- Universal Approximation Theorem (UAT)

  A feedforward neural network with at least one hidden layer, a finite number of neurons, and a suitable nonlinear activation function can approximate any continuous function on a closed and bounded interval to any desired degree of accuracy.

- Architecture of ANN(MLP) –Feedforward Network

  MLPs -universal function approximators.



6.4 Modeling Techniques- (ANN)

- Bias and Variance tradeoff Associated with –Model training and testing.

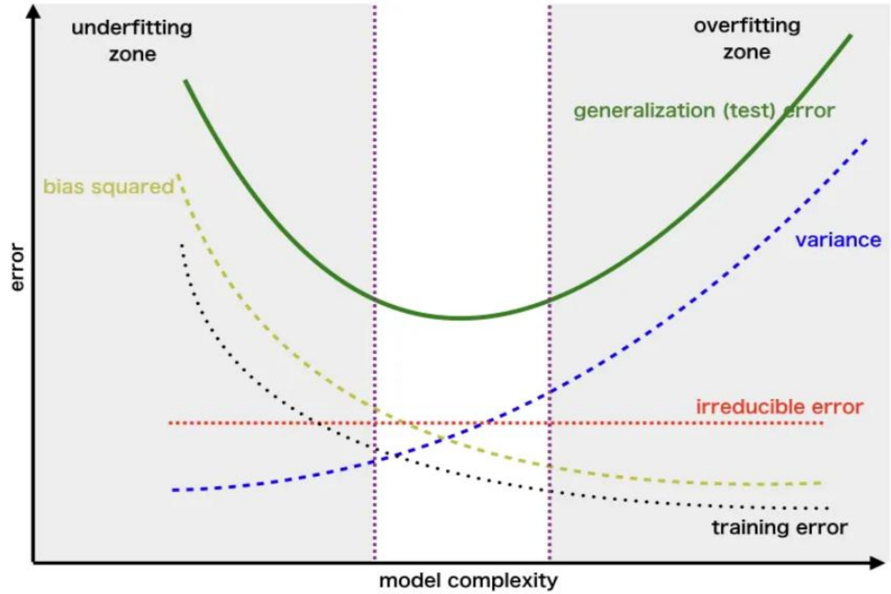
Model is :$\mathbf{y} = \boldsymbol{f}(\mathbf{X}) + \boldsymbol{\varepsilon}$ ~$\mathbf{N}(\boldsymbol{f}(X), \boldsymbol{\sigma}^2)$

Fitted/Estimated : $\hat{\boldsymbol{f}}(\boldsymbol{x})$

Error: [y-$\hat{f}(x)$] ---R.V

MSE: $\mathbb{E}\left[(\boldsymbol{y} - \hat{\boldsymbol{f}}(\boldsymbol{x}))^2\right] = \mathbb{E}\left[(\boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{\varepsilon} - \hat{\boldsymbol{f}}(\boldsymbol{x}))^2\right]$

$\boldsymbol{\varepsilon}$ ~$\mathbf{N}(\boldsymbol{f}(X), \boldsymbol{\sigma}^2)$



$$\mathbb{E}\left[(\boldsymbol{y} - \hat{\boldsymbol{f}}(\boldsymbol{x}))^2\right] = \underbrace{(\mathbb{E}[\hat{\boldsymbol{f}}(\boldsymbol{x})] - \boldsymbol{f}(\boldsymbol{x}))^2}_{\boldsymbol{Blas^2}} + \underbrace{\mathbb{E}\left[(\hat{\boldsymbol{f}}(\boldsymbol{x}) - \mathbb{E}[\hat{\boldsymbol{f}}(\boldsymbol{x})])^2\right]}_{\textbf{Varlance}} + \underbrace{\boldsymbol{\sigma}^2}_{\textbf{Irreduelble Error}}$$

6.4 Modeling Techniques- (ANN)

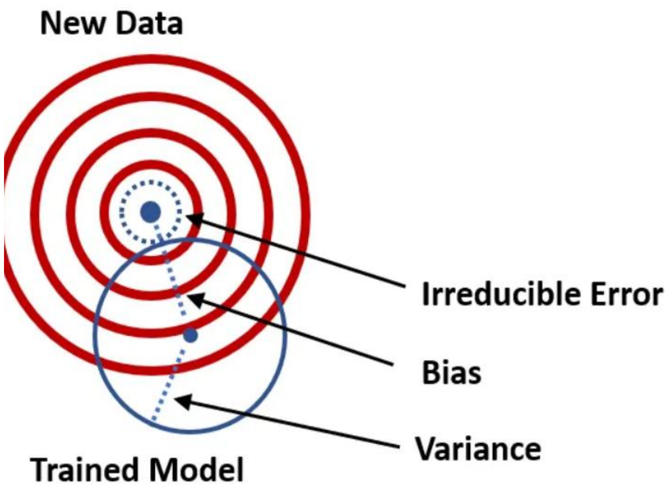
Behavior with Model Complexity

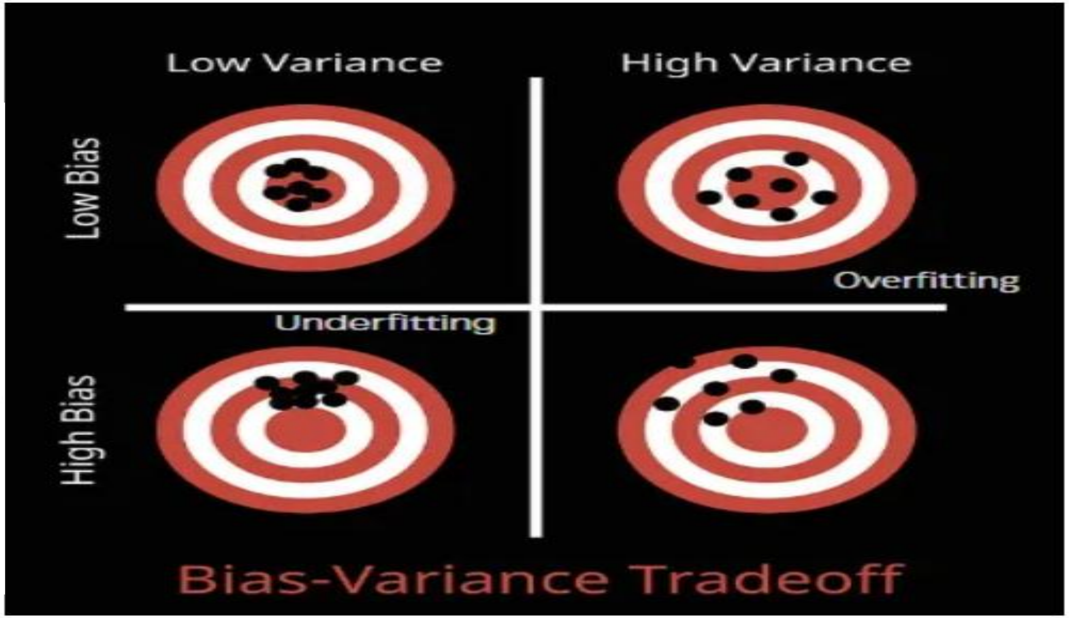
1.Low complexity models (e.g., linear regression on nonlinear data):

- High bias
- Low variance

2.High complexity models (e.g., high-degree polynomial, deep neural networks):

- Low bias
- High variance



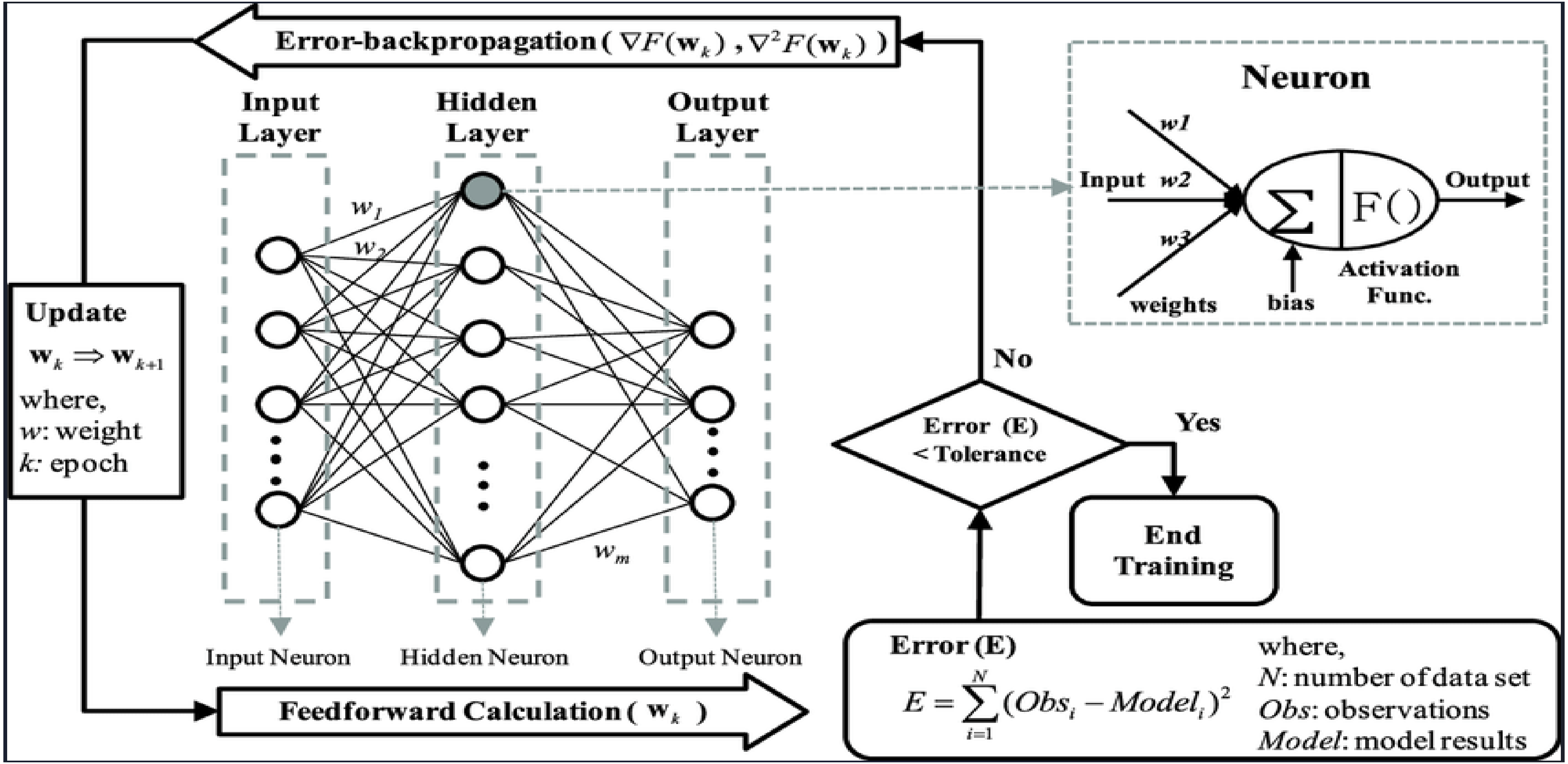


6.4 Modeling Techniques- (ANN Architecture)

- Terminologies/Components of ANN-

1. **input Layer:** Receives raw features (numeric or encoded).
2. **Weights & Biases:** Parameters learned during training.
3. **Activation Functions:** Introduce non-linearity (ReLU, Sigmoid, Tanh).
4. **Hidden Layers:** Transform inputs into higher-level representations.
5. **Output Layer:** Produces final prediction (regression/classification).
6. **Loss Function:** Measures model error (MSE, Cross-Entropy).
7. **Optimizer:** Adjusts weights (SGD, Adam).
8. **Feedforward & Backpropagation:** Forward prediction and gradient-based learning.

6.4 Modeling Techniques- (ANN Architecture)



6.4 Modeling Techniques- (ANN Architecture)

- Training Cycle:

1. **Initialization**-
   Weights and Biases Randomly. Define learning rate (η), activation functions, loss function, optimizer, and number of epochs.
2. **Forward Propagation-**
   Input data moves layer-by-layer through the network
   $$x \rightarrow W_1 x + b_1 \rightarrow a_1 \rightarrow W_2 a_1 + b_2 \rightarrow a_2 \rightarrow \cdots \rightarrow \hat{y}$$
   Each neuron performs:
   $$z = W^T x + b \; ; a = f(z)$$
   Where: **W** = weights, **b** = biases ; **f** = activation function (ReLU, Sigmoid, LeakyReLU etc.)
   a = activation (output of a neuron) ;This produces the **predicted output** $\hat{y}$.
3. **Compute Loss**: For regression: Loss $= \frac{1}{n} \sum (y - \hat{y})^2$
4. **Backpropagation**: Compute gradients of the loss with respect to each weight using the chain rule:
   $$\mathrm{w_{ji}} \leftarrow \mathrm{w_{ji}} + \Delta \mathrm{w_{ji}}$$
   Where , $\Delta w_{ji} = -\eta \frac{\partial E_d}{\partial w_{ji}}$, $\boldsymbol{Ed}$ is the error on training example $\mathbf{d}$.
5. **Repeat for all Training Batches** : The above forward + backprop + weight update is repeated for every batch:

One *full cycle over the entire dataset* = 1 *epoch*

6.4 Modeling Techniques- (ANN Architecture)

6. **Repeat for Many Epochs**

Continue for defined epochs (e.g., 50, 100).Weights stabilize and the model converges.

7. **Evaluate on Validation/Testing Data**

After training, evaluate using metrics:

-RMSE, MAE, R^2 for regression

-Accuracy, AUC, Precision, Recall for classification

Hence- ANN Training Involves:

Initialize weights and biases
Forward propagation: Compute outputs layer-by-layer
Compute loss using true vs predicted
Backpropagation: Calculate gradients
Update weights via gradient descent / optimizer
Repeat for all batches
Complete one epoch, repeat for many epochs
Evaluate model on test data

Optimization mechanism:

Derivation of Back Propagation Algorithm

$$w_{ji} \leftarrow w_{ji} + \Delta w_{ji}$$

Where $\Delta w_{ji} = -\eta \frac{\partial E_d}{\partial w_{ji}}$; $\boldsymbol{Ed}$ is the error on training example $\mathbf{d}$,

$$SSE = ErrorSS = E_d(\vec{w}) \equiv \frac{1}{2} \sum_{k \in \text{outputs}} (t_k - o_k)^2$$

Here outputs is the set of output units in the network, *Objective function to be minimised*: $E_d(\vec{w})$

$\mathbf{t_k}$ is the target value of unit k for training example $\mathbf{d}$, and

$\mathbf{o_k}$ is the output of unit k given training example $\mathbf{d}$.

Notations Used:

x_{ji}, w_{ji} = input and weights to unit j from the output of unit i

$net_j = \sum_i w_{ji} X_{ji}$ (the weighted sum of inputs to unit j)

o_j = the output computed by unit j using activation function $\sigma(net_j)$.

t_j = the target output for unit j

σ = the sigmoid function

outputs = {the set of units in the final layer of the network}

Downstream(**j** =**{All Units connected to j's tail}-** the set of units whose immediate inputs include the output of unit **j**.

6.4 Modeling Techniques- (ANN Architecture)

To derive a convenient expression for $\frac{\partial \boldsymbol{E_d}}{\partial \boldsymbol{net_j}}$ -We consider two cases in turn:

Case 1 , where unit j is an output unit for the network,

$\boldsymbol{o_j} = \sigma\left(\boldsymbol{net_j} = \sum_{\boldsymbol{i}} \boldsymbol{w_{ji} X_{ji}}\right)$ *and* $\boldsymbol{net_j} = f\left(w_{ji}\right)$

$$\frac{\partial E_d}{\partial w_{ji}} = \frac{\partial E_d}{\partial net_j}\frac{\partial net_j}{\partial w_{ji}} = \frac{\partial E_d}{\partial net_j} * \boldsymbol{X_{ji}} \qquad \text{-----(A)}$$

now , we need , $\frac{\partial E_d}{\partial net_j} = \frac{\partial E_d}{\partial oj} * \frac{\partial \text{oj}}{\partial net_j} \qquad --(\text{B})$

$$\frac{\partial E_d}{\partial o_j} = \frac{\partial}{\partial o_j}\frac{1}{2}\sum_{k\in \text{ outputs}} (t_k - o_k)^2 = \frac{\partial}{\partial o_j}\frac{1}{2}\left(t_j - o_j\right)^2$$

$$= \frac{1}{2}2\left(t_j - o_j\right)\frac{\partial\left(t_j - o_j\right)}{\partial o_j} = -\left(t_j - o_j\right) \qquad --(C)$$

For Sigmoid Function, $\frac{\partial \sigma(x)}{\partial x} = [1 - \sigma(x)][\sigma(x)]$, $\boldsymbol{o_j} = \sigma\left(\boldsymbol{net_j} = \sum_{\boldsymbol{i}} \boldsymbol{w_{ji} X_{ji}}\right)$

it implies that,

$$\frac{\partial o_j}{\partial(\text{ net }_j)} = \frac{\partial \sigma(\text{ net }_j)}{\partial(\text{ net }_j)} = \sigma(\text{ net }_j)\left(1 - \sigma(\text{ net }_j)\right) = o_j\left(1 - o_j\right) -----(D)$$

So, $$\frac{\partial E_d}{\partial net_j} = \frac{\partial E_d}{\partial o_j}\frac{\partial o_j}{\partial net_j} = -\left(t_j - o_j\right) * o_j\left(1 - o_j\right)$$

$$\frac{\partial E_d}{\partial w_{ji}} = \frac{\partial E_d}{\partial net_j} * \boldsymbol{X_{ji}} = -\left(t_j - o_j\right) * o_j\left(1 - o_j\right) * \boldsymbol{X_{ji}}$$

6.4 Modeling Techniques- (ANN Architecture)

To derive a convenient expression for $\frac{\partial E_d}{\partial net_j}$ -We consider two cases in turn:

Case 1 , where unit j is an output unit for the network,

$$o_j = \sigma\left(net_j = \sum_i w_{ji} X_{ji}\right) \text{ and } net_j = f(w_{ji}) = \sum_i w_{ji} X_{ji}$$

$$\frac{\partial E_d}{\partial w_{ji}} = \frac{\partial E_d}{\partial net_j}\frac{\partial net_j}{\partial w_{ji}} = \frac{\partial E_d}{\partial net_j} * X_{ji} \qquad \text{-----(A)}$$

now , we need , $\frac{\partial E_d}{\partial net_j} = \frac{\partial E_d}{\partial Oj} * \frac{\partial Oj}{\partial net_j} \qquad --(\mathbf{B})$

$$\frac{\partial E_d}{\partial o_j} = \frac{\partial}{\partial o_j}\frac{1}{2}\sum_{k\in \text{ outputs}}(t_k - o_k)^2 = \frac{\partial}{\partial o_j}\frac{1}{2}(t_j - o_j)^2$$

$$\frac{\partial E_d}{\partial o_j} = \frac{1}{2}2(t_j - o_j)(-1) = -(t_j - o_j) \qquad --(C)$$

For Sigmoid Function, $\frac{\partial \sigma(x)}{\partial x} = [1-\sigma(x)][\sigma(x)]$, $o_j = \sigma\left(net_j = \sum_i w_{ji} X_{ji}\right)$

$$\Rightarrow \qquad \frac{\partial o_j}{\partial(\text{ net }_j)} = \frac{\partial \sigma(\text{ net }_j)}{\partial(\text{ net }_j)} = \sigma(\text{ net }_j)\left(1-\sigma(\text{ net }_j)\right) = o_j(1-o_j) \text{ -----}(D)$$

So, $\frac{\partial E_d}{\partial net_j} = \frac{\partial E_d}{\partial o_j}\frac{\partial o_j}{\partial net_j} = -(t_j - o_j) * o_j(1-o_j) = -\delta_j \qquad \text{-------(B)}$

$$\frac{\partial E_d}{\partial w_{ji}} = \frac{\partial E_d}{\partial net_j} * X_{ji} = -(t_j - o_j) * o_j(1-o_j) * X_{ji} \text{-------(A)}$$

Let, $\delta_j = (t_j - o_j)o_j(1-o_j)$,So,

$$\frac{\partial E_d}{\partial w_{ji}} = \delta_j * X_{ji} \quad \text{and} \quad \Delta w_{ji} = -\eta\frac{\partial E_d}{\partial w_{ji}} = \eta * \delta_{j*} X_{ji} \text{ -----}(\mathbf{E})$$

6.4 Modeling Techniques- (ANN Architecture)

- **Case 2, where unit j is an internal unit of the network.**

$$\frac{\partial E_d}{\partial w_{ji}} = \frac{\partial E_d}{\partial net_j}\frac{\partial net_j}{\partial w_{ji}}$$

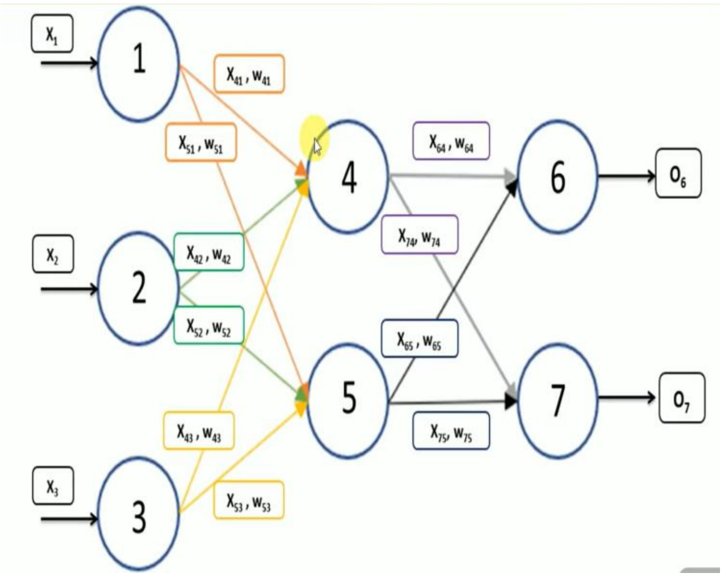
$o_j = X_{Kj}$ (X_{Kj} = Input k<-j =output of Neuron j (o_j))

=> $\frac{\partial net_k}{\partial o_j} = \frac{\partial}{\partial o_j}\left(\sum_i w_{ki}X_{ki}\right) = \frac{\partial}{\partial o_j}\left(\sum_i w_{ki}o_i\right) = w_{Kj} --(\mathrm{E})$

$$o_j = \sigma\left(net_j = \sum_i w_{ji}X_{ji}\right)$$

$$\frac{\partial o_j}{\partial(\text{ net }_j)} = \frac{\partial\sigma(\text{ net }_j)}{\partial(\text{ net }_j)} = \sigma(\text{ net }_j)\left(1-\sigma(\text{ net }_j)\right) = o_j(1-o_j)$$

-



- From case I ,when j is output unit ,there is no downstream.

$$\frac{\partial E_d}{\partial net_j} = \frac{\partial E_d}{\partial o_j}\frac{\partial o_j}{\partial net_j} = -(t_j - o_j) * o_j(1-o_j) = -\delta_j \quad \text{----(B)}$$

So,When j *is hiden unit*,there is downstream.

$$\frac{\partial E_d}{\partial net_j} = \sum_{k\in \text{ Downstream }'j}\frac{\partial E_d}{\partial net_k}\frac{\partial net_k}{\partial net_j} = \sum_{k\in \text{ D}'(j)}\left[-\delta_k\frac{\partial net_k}{\partial net_j} = -\delta_k\frac{\partial net_k}{\partial o_j}\frac{\partial o_j}{\partial net_j} = \delta_k w_{kj}\frac{\partial o_j}{\partial net_j}\right]$$

$$\frac{\partial E_d}{\partial net_j} = \sum_{k\in \text{ Downstream }(j)} -\delta_k w_{kj} o_j(1-o_j) \quad \text{----(F)}$$

On Substitution in equation we get, $\frac{\partial E_d}{\partial w_{ji}} = \frac{\partial E_d}{\partial net_j}\frac{\partial net_j}{\partial w_{ji}} = \frac{\partial E_d}{\partial net_j} * \frac{\partial(\sum_i w_{ji}X_{ji})}{\partial w_{ji}} = o_j(1-o_j)\sum_{k\in \text{ Downstream }(j)}\delta_k w_{kj}x_{ji}$

- **Hence,** $\Delta w_{ji} = -\eta\frac{\partial E_d}{\partial net_j}x_{ji} = \Delta w_{ji} = \eta o_j(1-o_j)\sum_{k\in \text{ Downstream }(j)}\delta_k w_{kj}x_{ji}$

6.4 Modeling Techniques- (ANN Architecture)

$\boldsymbol{P}[-\boldsymbol{z}_{\alpha/2} \leq \boldsymbol{N}(\mathbf{0},\mathbf{1}) \leq \boldsymbol{z}_{\alpha/2}] = \mathbf{1} - \boldsymbol{\alpha}$; $\boldsymbol{P}[-z_{\frac{\alpha}{2}} \frac{\hat{\tau}}{\sqrt{n}} \leq [\text{Loss} - \text{E}(\text{Loss})] \leq z_{\alpha/2} \frac{\hat{\tau}}{\sqrt{n}}] = \mathbf{1} - \boldsymbol{\alpha}$

For Given level of error metric in the gradient estimate ε and significance level α.

s.t $\boldsymbol{P}[-\varepsilon \leq \boldsymbol{N}(\mathbf{0},\mathbf{1}) \leq \varepsilon] = \mathbf{1} - \boldsymbol{\alpha}$; we require:

$$z_{\alpha/2} \cdot \frac{\hat{\tau}}{\sqrt{n}} < \varepsilon$$

Solving for Number of Epochs n,

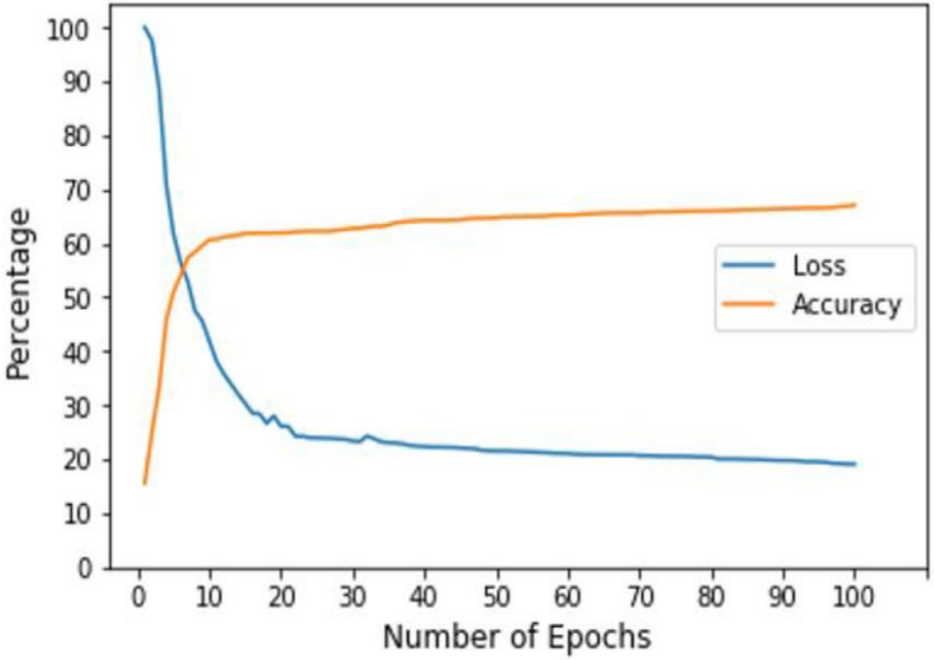
Squaring both sides leads to the lower bound on the number of epochs:

$$n > \frac{z_{\alpha/2}^2 \cdot \hat{\tau}^2}{\varepsilon^2}$$

n =Number of Epochs

For our Model= R_squared=71.85%

n used were at least =100



7. Model Comparison & Evaluation)

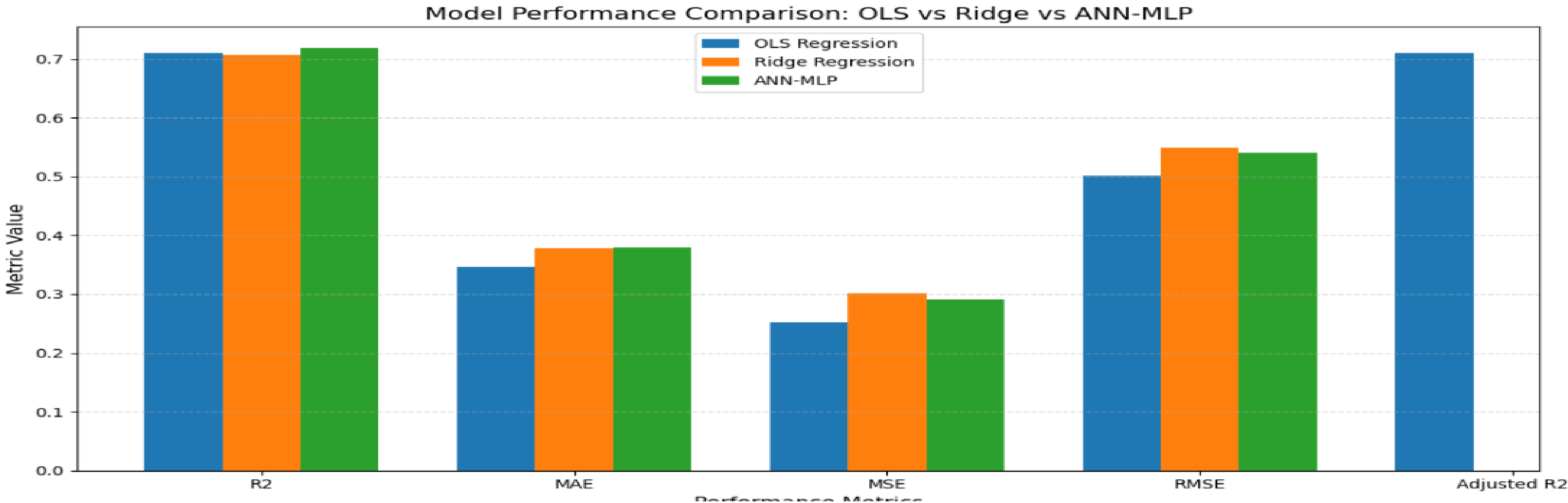
Models-1. Ridge Regression
2.OLS Linear Regression
3.Feedforward –ANN (MLP)

	Ridge Regression Output					
	Feature	**Coefficient**	**Std Error**	**t-Value**	**Lower 95% CI**	**Upper 95% CI**
1	TermGPA	0.783354613	0.00255523	306.5692829	0.778346338	0.788362889
2	Intercept	0.167987915	0.02160827	7.774241674	0.125635472	0.210340358
3	AcademicLevelEndofTerm	0.104537336	0.00321499	32.5156332	0.098235927	0.110838745
4	College_College of Nursing	0.071596165	0.02202623	3.250495778	0.028424518	0.114767812
5	College_Undergraduate Education	0.051445371	0.02268665	2.267649549	0.006979293	0.09591145
6	College_Coll of Ag Life & Env Sci	0.047293361	0.02117242	2.233724945	0.005795191	0.088791532
7	UnitsPassednotincludedinGPA	0.022453461	0.00247184	9.083709061	0.017608631	0.027298291
8	Age	0.009205319	0.00261016	3.526726067	0.004089377	0.01432126
9	AcademicYear	0.007722043	0.00243666	3.169113384	0.002946168	0.012497917
	Feature	**Coefficient**	**Std Error**	**t-Value**	**Lower 95% CI**	**Upper 95% CI**
10	College_Colleges of Letters Arts &	-0.45510578	0.03920746	-11.6076327	-0.531952819	-0.378258736
11	PrimaryMilitaryAffiliation_Child Dependent	-0.18321572	0.02403263	-7.62362462	-0.230319924	-0.136111512
12	College_College of Humanities	-0.18306474	0.0217865	-8.40266987	-0.225766513	-0.140362974
13	AcademicCareer_Law	-0.15379352	0.02340568	-6.57077895	-0.199668903	-0.107918147
14	College_Graduate College	-0.14719668	0.02754366	-5.34412204	-0.201182554	-0.093210811
15	PrimaryMilitaryAffiliation_No Military Affiliation	-0.1338414	0.01041802	-12.8471053	-0.15426083	-0.113421967
16	College_College of Science	-0.12318487	0.02082229	-5.91601007	-0.163996778	-0.082372958
17	PrimaryMilitaryAffiliation_Veteran	-0.12062966	0.01341483	-8.99226283	-0.146922864	-0.09433645
18	PrimaryMilitaryAffiliation_Unknown Military Affiliation	-0.11971578	0.05276489	-2.26885288	-0.223135544	-0.01629602
19	PrimaryMilitaryAffiliation_Guard Reserve	-0.09931297	0.02683743	-3.70053922	-0.151914621	-0.046711312
20	PrimaryMilitaryAffiliation_Spouse Dependent	-0.08589272	0.02643719	-3.24893493	-0.137709912	-0.034075538
21	PrimaryMilitaryAffiliation_Other Dependent	-0.08393854	0.01328204	-6.31970429	-0.109971475	-0.057905606
22	College_College of Social & Behav Sci	-0.05244689	0.02002987	-2.61843429	-0.091705651	-0.013188136
23	FirstGenerationFlag_Y	-0.04214167	0.00538323	-7.82832208	-0.052692864	-0.03159048
24	NumberofClassesEnrolled	-0.01802168	0.00256306	-7.03131346	-0.02304531	-0.012998057

OLS Regression Output

	Predictor Variable	Coefficient	Std Error	t-value	Lower CI	Upper CI
1	TermGPA	0.667240429	0.001948872	342.3726096	0.663420637	0.671060221
2	AcademicLevelEndofTerm	0.071543898	0.00196452	36.41800511	0.067693436	0.07539436
3	College_College of Nursing	0.057974952	0.01854496	3.126183621	0.021626803	0.0943231
4	College_Undergraduate Education	0.040959533	0.019095743	2.144956246	0.00353185	0.078387216
5	College_Coll of Ag Life & Env Sci	0.03869261	0.017827674	2.170367835	0.003750344	0.073634877
6	UnitsPassednotincludedinGPA	0.02983654	0.002965575	10.06096169	0.024024008	0.035649072
7	AcademicYear	0.004665882	0.001238714	3.766715194	0.002238001	0.007093762
8	Age	0.00092843	0.000238821	3.88756103	0.000460341	0.001396519
	Predictor Variable	Coefficient	Std Error	t-value	Lower CI	Upper CI
9	const	-8.421920168	2.506263634	-3.360348869	-13.33420046	-3.509639872
10	College_Colleges of Letters Arts & Sci	-0.41249482	0.032691932	-12.6176337	-0.476571053	-0.348418587
11	PrimaryMilitaryAffiliation_Child Dependent	-0.18535261	0.020023734	-9.25664586	-0.224599156	-0.146106064
12	College_College of Humanities	-0.170401491	0.018379216	-9.271423176	-0.206424781	-0.134378201
13	AcademicCareer_Law	-0.153957023	0.019660483	-7.830785504	-0.192491597	-0.115422449
14	College_Graduate College	-0.133838003	0.023179827	-5.773899939	-0.179270497	-0.088405508
15	PrimaryMilitaryAffiliation_Unknown Military Affiliation	-0.133339688	0.046395985	-2.873948839	-0.224275886	-0.042403491
16	PrimaryMilitaryAffiliation_No Military Affiliation	-0.122700028	0.008726935	-14.0599221	-0.139804833	-0.105595223
17	College_College of Science	-0.120359785	0.017537144	-6.863135055	-0.154732612	-0.085986959
18	PrimaryMilitaryAffiliation_Guard Reserve	-0.118184362	0.022405707	-5.274743728	-0.16209958	-0.074269145
19	PrimaryMilitaryAffiliation_Veteran	-0.10940313	0.01126446	-9.712239267	-0.131481488	-0.087324773
20	PrimaryMilitaryAffiliation_Other Dependent	-0.075596089	0.01111173	-6.803269353	-0.097375095	-0.053817083
21	PrimaryMilitaryAffiliation_Spouse Dependent	-0.073651256	0.022625222	-3.255272278	-0.117996723	-0.02930579
22	College_College of Social & Behav Sci	-0.052348839	0.016882474	-3.100780022	-0.085438513	-0.019259165
23	FirstGenerationFlag_Y	-0.042242094	0.004506299	-9.374010381	-0.051074447	-0.033409742
24	NumberofClassesEnrolled	-0.011938809	0.001555494	-7.675252938	-0.01498758	-0.008890039

Comparision of Models Based On Performance metrics.							
OLS-Regression			RIDGE REGRESSION			ANN-MLP =Feedforword Network	
R^2 Score:	0.710469		R^2 (test):	0.707256597		R^2 (test):	0.7185
MAE:	0.346291		MAE (test):	0.377366253		MAE (test):	0.3791
MSE:	0.251616		MSE (test):	0.300893569		MSE (test):	0.2914
RMSE:	0.501613		RMSE (test):	0.548537664		RMSE (test):	0.5398
Adjusted R^2:	0.710304						



8. Key Results, Insights & Discussion

- **Model Comparison & Interpretability**
  1. **Linear Regression models (OLS & Ridge)** are more interpretable and less complex.
  2. **Neural Networks (ANN–MLP)** are inherently more complex but generally provide higher predictive power.
- However, in our dataset—which contains **noise, high dimensionality, and many sparse categorical predictors**—the ANN did **not significantly outperform** OLS or Ridge.
  This indicates that model complexity alone does not guarantee better performance when data quality issues exist.

8. Key Results, Insights & Discussion

Key Predictors Requiring Attention

Based on OLS, Ridge, and ANN feature influence, the following predictors show **large negative coefficients**, meaning they reduce predicted Cumulative GPA and warrant institutional focus:

- **Colleges:**
  1. Colleges of Letters Arts & Sciences
  2. College of Humanities
  3. Graduate College
  4. College of Science
- **Primary Military Affiliation variables**
- **First-Generation Student Flag**
- **Number of Classes Enrolled**

These predictors may signal student groups needing **additional academic support, resources, or redesigned interventions**.

8. Key Results, Insights & Discussion

Data Characteristics & Modeling Challenges

- The dataset includes a **high proportion of nominal (categorical) predictors** with many unique categories.
- After one-hot encoding, this leads to **data sparsity**, which:
  1. slows down ANN training,
  2. increases variance,
  3. worsens the **curse of dimensionality**,
  4. and amplifies **noise** in the dataset.

As a result, ANN performance remained similar to OLS/Ridge rather than exceeding it.

.9. Insights & Discussion Limitations

Role of Predictive Analytics

- While **correlation is not causation**, predictive analytics helps:
    1. Identify influential predictors,
    2. Guide institutional attention and resource allocation,
    3. Validate expert opinions using data-driven evidence, and
    4. Support **proactive, timely intervention strategies** for student success.

Predictive Analytics therefore serves as a practical tool to **prioritize at-risk groups**, support policy design, and improve educational outcomes.

.9. Insights & Discussion Limitations

- Limitations-

1.Data Quality Issues (Noise, Missingness, Inconsistent Records)

-Predictive models are only as strong as the data fed into them.

-Educational datasets often contain **noisy, inconsistent, or incomplete records**, leading to biased or unstable predictions.

-Missing values and measurement errors can distort relationships between predictors and GPA.

2. High Dimensionality & Data Sparsity

-Including many **high-cardinality nominal predictors** (e.g., Colleges, Military Affiliation, AcademicCareer) increases the number of dummy variables.

-This leads to **sparsity**, slowing model training (especially ANN) and increasing the risk of overfitting.

-Sparse data also weakens statistical power for detecting significant relationships.

.9. Insights & Discussion Limitations

3. Multicollinearity

-Predictors such as **AcademicCareer ⇔ AcademicLevelEndofTerm**, **UnitsPassedincludedinGPA ⇔ NumberofClassesEnrolled** show high correlation.

-Multicollinearity:

1.Inflates standard errors,
2.Makes individual p-values unreliable,
3.Makes model coefficients unstable to small data changes.

Even Ridge Regression does not eliminate all interpretability issues.

4. Model Interpretability vs Accuracy Trade-off

-Linear Regression is **interpretable**, but may undershoot complex nonlinear relationships.

-ANN can model nonlinear structure but suffers from:

1.lack of transparency (black-box),
2.high sensitivity to tuning parameters,
3.difficulty explaining the effect of each predictor.

For institutional decision-making (education domain), high interpretability is crucial.For the same reason ANN cshould be used to validate the results of mainstream Linear Regression Models.

.9. Insights & Discussion Limitations

5. Assumption Violations

Traditional models rely on:

- linearity,
- homoscedasticity,
- independence of errors,
- normally distributed residuals.

-Educational datasets often violate these assumptions due to:

- heterogeneous student groups,(Mostly because of Protected Variables)
- different grading standards,
- non-linear academic progression.

This affects model stability and predictive accuracy.

.9. Insights & Discussion Limitations

6. Temporal and Contextual Changes

-Student performance patterns change over time (policy changes, curriculum shifts, online learning expansion).

-Predictive models become **stale** if not re-trained or recalibrated regularly.

7. Ethical and Fairness Concerns

-Models may unintentionally encode bias against:

1. first-generation students,
2. military-affiliated groups,
3. certain colleges or majors.

Poorly controlled predictive systems can lead to unfair academic decisions (Example-College Admission) or misclassification of at-risk students.(Type 2 Risk)

.9. Insights & Discussion Limitations

8. Overfitting Risks (Especially in ANN)

-ANN tends to overfit when:

-data is noisy,

-too many parameters vs observations,

-categorical variables are sparse.

Despite regularization and dropout, performance may not significantly surpass simpler models.

9. Limited Causal Inference

-Predictive analytics identifies patterns, **not causal relationships**.

-Even significant predictors cannot guarantee intervention success;
e.g., increasing *UnitsPassed* may not *cause* higher GPA.

10. Deployment Challenges

-Real-world deployment requires:

-continuous monitoring,

-updating feature pipelines,

-integration with existing student information systems.

Resource requirements may be unrealistic for smaller institutions.

10. Recommendations

1. Use Predictive Models Carefully

-Predictive Analytics is statistical models based on **the Law of Large Numbers (LLN).**

-Individual admissions should NOT rely solely on models—human academic judgment is essential.

2. Improve Student Success, Not Selection

-Use analytics to enhance **student learning, support, and progression**.

-Goal: Help students succeed academically, personally, and professionally.

3. Analytics Should Support, Not Replace Humans

Predictive insights must complement:

-Advisor judgement

-Faculty interventions

-Holistic student evaluation

4. Maintain Human Oversight

-Human review is essential to avoid **blind, mechanical, or biased decisions**.

-Predictive Analytics should guide decisions—not make them independently.

11. Conclusion & Future Work

Conclusion

- Predictive analytics is helpful but **should not replace human judgment**.
- Statistical models rely on **population-level patterns**, not individual-level decision-making.
- Use predictive insights to **support**:
  - -academic advising,
  - -faculty intervention,
  - -holistic student evaluation.
- **Human oversight remains essential** to avoid mechanical or unfair decisions.

Future Work

- Improve data quality to reduce **noise** and **sparsity**.
- Explore more advanced ML models with better handling of high-dimensional categorical data.
- Integrate real-time student engagement data for more accurate predictions.
- Develop explainable AI tools to help educators understand model recommendations like
- Use SHAP to ensure university predictive models remain fair, unbiased, and explainable by detecting any unintended influence from sensitive predictors.
- Continuously monitor model performance to avoid bias and ensure fairness.

Thank You!