BASAVARAJESWARI GROUP OF INSTITUTIONS

BALLARI INSTITUTE OF TECHNOLOGY & MANAGEMENT

NACC Accredited Institution*

(Recognized by Govt. of Karnataka, approved by AICTE, New Delhi & Affiliated to Visvesvaraya Technological University, Belagavi)

"JnanaGangotri" Campus, No.873/2, Ballari-Hospet Road, Allipur, Ballar1-583 104 (Karnataka) (India)

Ph: 08392 – 237100 / 237190, Fax: 08392 – 237197

DEPARTMENT OF CSE-ARTIFICIAL INTELLIGENCE

A Mini-Project Report On

"SALARY PREDICTION USING ANN MODEL"

A report submitted in partial fulfillment of the requirements for the

NEURAL NETWORK AND DEEP LEARNING

Submitted By

M MARULESH **USN: 3BR22CA027**

Under the Guidance of

Prof. Pavan Kumar
And Mr. Vijay Kumar

Dept of CSE (Artificial Intelligence),
BITM, Ballari

Visvesvaraya Technological University

Belagavi, Karnataka 2025-2026

BASAVARAJESWARI GROUP OF INSTITUTIONS

BALLARI INSTITUTE OF TECHNOLOGY & MANAGEMENT

NACC Accredited Institution*
(Recognized by Govt. of Karnataka, approved by AICTE, New Delhi & Affiliated to Visvesvaraya Technological University, Belagavi)
"JnanaGangotri" Campus, No.873/2, Ballari-Hospet Road, Allipur, Ballari-583 104 (Karnataka) (India)
Ph: 08392 – 237100 / 237190, Fax: 08392 – 237197

CERTIFICATE

Certified that the mini project work entitled "Salary prediction Using ANN Model" carried out by M Marulesh bearing USN 3BR22CA027 A Bonafide students of Ballari Institute of Technology and Management in partial fulfillment for the award of Bachelor of Engineering in CSE (Artificial Intelligence) of the Visvesvaraya Technological University, Belgaum during the year 2025 - 2026. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the report deposited in the departmental library. The project report has been approved as it satisfies the academic requirements in respect of the project work prescribed for the said Degree.

12/12/25

Signature of Lab Co-Ordinator's
Prof. Pavan Kumar and Mr.Vijay Kumar

Signature of HOD
Dr. Yeresime Suresh

ABSTRACT

Predicting employee salaries has become an essential task in modern data-driven human resource management, enabling organizations to design fair compensation structures, identify market trends, and support efficient recruitment strategies. Traditional statistical methods often struggle to model the complex, nonlinear relationships between various factors influencing salary levels. With advancements in artificial intelligence, Artificial Neural Networks (ANNs) have emerged as powerful predictive tools capable of learning intricate patterns from data. This project focuses on developing an ANN-based regression model to predict an individual's salary using a dataset containing features such as years of experience, age, education level, job title, and skill ratings. The dataset undergoes preprocessing steps including handling missing values, feature scaling, and encoding categorical variables to ensure high-quality input for the neural network. The ANN is designed with multiple dense layers, ReLU activation functions, dropout regularization to prevent overfitting, and a linear activation in the output layer suitable for continuous salary prediction.

After model construction, the ANN is trained with optimized hyperparameters and evaluated using regression metrics such as Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and the coefficient of determination (R^2 score). Training and validation loss curves further illustrate the learning behaviour of the model and highlight its performance.

The results indicate that the ANN effectively captures the nonlinear dependencies among input features, enabling accurate salary predictions for diverse employee profiles. This study demonstrates the potential of deep learning techniques in HR analytics and highlights their capability to support data-driven decision-making in workforce planning. With additional data sources and further optimization, the model can be scaled for real-world industry applications such as automated salary recommendation systems, employee evaluation platforms, and strategic HR forecasting.

.

ACKNOWLEDGEMENT

The satisfactions that accompany the successful completion of our mini project on SALRAY PREDICTION USING ANN MODEL would be incomplete without the mention of people who made it possible, whose noble gesture, affection, guidance, encouragement and support crowned my efforts with success. It is our privilege to express our gratitude and respect to all those who inspired us in the completion of our mini-project.

I am extremely grateful to my Lab Coordinators **Prof. Pavan Kumar and Mr. Vijay Kumar** for their noble gesture, support co- ordination and valuable suggestions given in completing the mini-project. I also thank **Dr. YERISIME SURESH,** H.O.D. Department of CSE(AI), for his co-ordination and valuable suggestions given in completing the mini-project. We also thank Principal, Management and non-teaching staff for their co-ordination and valuable suggestions given to us in completing the Mini project.

Name	USN
M MARULESH	3BR22CA027

TABLE OF CONTENTS

1. INTRODUCTION

Salary prediction is an important task in modern workforce analytics, helping organizations understand compensation trends, maintain pay equity, and make informed hiring decisions. An employee's salary is typically influenced by several factors such as education, skills, years of experience, job role, and industry standards. Understanding how these factors collectively determine wages is essential for designing fair compensation systems, reducing employee dissatisfaction, and improving human resource planning.Traditional salary estimation methods rely heavily on manual analysis, expert judgment, and linear statistical models. Although useful, these approaches often struggle to capture the nonlinear relationships and interactions among multiple features. Manual evaluation can also be time-consuming, inconsistent, and prone to human bias—especially when dealing with large datasets or diverse employee profiles. This has created a growing need for automated, intelligent prediction models that can provide accurate and scalable estimations.

With the advancement of artificial intelligence and machine learning, predictive modelling has become a powerful tool for analyzing employment-related data. Among these techniques, **Artificial Neural Networks (ANNs)** have shown exceptional capability in learning complex relationships and modelling nonlinear patterns. Inspired by the functioning of the human brain, ANNs can analyze high-dimensional datasets, identify hidden trends, and generate precise predictions. This makes them particularly suited for salary prediction tasks where multiple features interact in non-obvious ways.This project focuses on building an ANN-based regression model to predict employee salaries using features such as years of experience, age, education level, job role, and professional skills. By training the model on real-world or publicly available salary datasets, the ANN learns how different attributes contribute to salary outcomes. The process includes data preprocessing, encoding of categorical variables, feature scaling, model construction, training, evaluation, and result visualization.

The primary objective of this project is to design, implement, and assess a deep learning model capable of accurately predicting continuous salary values. Performance is evaluated using metrics such as Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and R^2 score. Visualizations of training and validation loss further provide insights into how well the model learns and generalizes.

1.1 Problem Statement

To design and develop an Artificial Neural Network (ANN) model that can accurately predict employee salaries based on key input features such as years of experience, education level, job role, and skills.
The model aims to learn patterns from the salary dataset and generate reliable predictions, helping organizations make data-driven decisions in recruitment and compensation management.

1.2 Scope of the project

The scope of this project includes the development, implementation, and evaluation of an ANN model designed to predict employee salaries based on relevant features. The project covers all major stages of a machine learning pipeline, including data preprocessing, feature encoding, normalization, model construction, training, and performance evaluation using regression metrics such as MAE, MSE, RMSE, and R^2 score.
It also includes visual analysis through training and validation loss graphs to better understand the behaviour and learning dynamics of the model. While this project is limited to the dataset used, the methodology can be extended to larger and more diverse datasets, integrated into HR management systems, and adapted for real-time salary recommendation platforms. Such predictive models can support HR professionals, help establish fair compensation structures, and enhance decision-making in talent management and recruitment.

1.3 Objectives

- To build an ANN model for accurate salary prediction.
- To preprocess, encode, and normalize the dataset for improved model performance.
- To evaluate the model using regression metrics such as MAE, MSE, RMSE, and R^2 score.
- To visualize training and validation behavior through loss curves and performance graphs.

2. LITERATURE SURVEY

[1] **Sharma et al. (2023)** explored salary prediction using machine learning regression models, including Linear Regression, Random Forest, and Gradient Boosting. Their study showed that ensemble techniques outperform linear models due to their ability to capture nonlinear dependencies in salary-related features. They emphasized the importance of feature engineering and data preprocessing for improving prediction accuracy.

[2] **Patel and Mehta (2023)** developed a deep learning–based salary prediction system using a multi-layer Artificial Neural Network. The results demonstrated that ANNs provide higher accuracy than traditional regression methods, especially when working with complex datasets containing categorical and numerical features. The authors highlighted that proper encoding and normalization significantly enhance model performance.

[3] **Zhang et al. (2022)** conducted a comparative analysis of various machine learning algorithms for compensation forecasting across different industries. They concluded that tree-based models and neural networks perform best due to their ability to model nonlinear relationships. Their study stressed the need for interpretable models to support HR decision-making.

[4] **Khan & Ali (2022)** evaluated several regression techniques, including Support Vector Regression (SVR) and Random Forest Regressor, for predicting employee salaries. Their findings indicated that Random Forest achieves superior performance, especially when the dataset contains diverse and high-dimensional feature sets.

[5] **Fernandes et al. (2023)** examined deep learning methods for wage estimation and demonstrated that ANN architectures can effectively learn patterns from experience, education, and job role data. However, they noted that larger datasets and hyperparameter tuning are essential for avoiding overfitting and improving generalization.

[6] **Reddy (2024)** proposed a hybrid machine learning framework combining XGBoost with neural network layers for enhanced salary prediction. The study reported significant improvements in prediction accuracy, emphasizing that hybrid models can leverage both feature selection and deep learning advantages.

3. SYSTEM REQUIREMENTS

The system requirements for developing the salary prediction model include both software and hardware components essential for performing data preprocessing, model training, and evaluation efficiently. The software environment is built using Python along with widely used data science and deep learning libraries. TensorFlow/Keras is used for constructing the ANN model, Pandas and NumPy handle data manipulation, Scikit-learn supports preprocessing and regression evaluation metrics, and Matplotlib or Seaborn is used for visualizing training behavior and results. The project is implemented using an interactive development environment such as Jupyter Notebook, Google Colab, or VS Code, which allows smooth execution and debugging of the model.

On the hardware side, the project can run effectively on a standard personal computer with a minimum of **4 GB RAM**, although **8 GB RAM** is recommended for faster preprocessing and model training. A **dual-core or multi-core processor** is sufficient to handle computations involved in salary prediction tasks. While GPU support is optional, it can significantly accelerate the training process, especially when working with larger datasets or deeper neural networks. Overall, the system requirements are minimal, ensuring that the project can be executed on most modern computing devices.

To implement the salary prediction system successfully, a stable computing environment capable of managing machine learning workflows is required. Python serves as the primary programming language due to its flexibility and the availability of robust libraries for deep learning and regression modelling. Essential tools include TensorFlow for building ANN architectures, Scikit-learn for preprocessing and evaluation, and Pandas for efficient dataset handling

3.1 Software Requirements

Python 3.8 or above
• TensorFlow / Keras (for building and training the ANN model)
• NumPy (for numerical computations)
• Pandas (for data loading and preprocessing)
• Scikit-learn (for encoding, scaling, and evaluation metrics)
• Matplotlib (for visualization of graphs)
• Google Colab (development environment)

3.2 Hardware Requirements

- Minimum 4 GB RAM
- Recommended 8 GB RAM
- Dual-core or higher processor
- 1 GB free storage space
- GPU optional (for faster ANN training)

3.1 Functional Requirements

• The system must load and preprocess the salary dataset.

• It must handle missing values, encode categorical features, and normalize numerical inputs.

• The system must build an ANN model for salary prediction (regression).

• It must train the ANN model using the prepared training data.

• The system must evaluate model performance using MAE, MSE, RMSE, and R^2 score.

• It must generate training and validation loss graphs for analysis.

• The system must predict salary values for new input data.

3.2 Non-Functional Requirements

• The system should provide accurate and reliable salary predictions.

• It should offer clear, interpretable, and user-friendly output.

• The system must execute efficiently on basic hardware configurations.

• It should remain stable even when data contains noise or imperfect values.

• The system must be easy to maintain, modify, and extend.

• Results should be clearly interpretable through graphs and regression metrics.

4 DESCRIPTION OF MODULES

The Artificial Neural Network–based salary prediction system is organized into several modules, each responsible for a specific step in the machine learning workflow. These modules work together to ensure efficient data preprocessing, model construction, training, evaluation, and visualization.

4.1 Data Preprocessing Module

This module loads the salary dataset and prepares it for ANN training. It handles missing values, encodes categorical features such as job role or education level, and normalizes numerical features like experience and age. These preprocessing steps ensure that the data is clean, consistent, and suitable for a regression-based neural network model.

4.2 ANN Model Building Module

This module constructs the Artificial Neural Network architecture used for salary prediction. It includes an input layer, multiple hidden layers with ReLU activation functions, and an output layer with a linear activation for continuous salary prediction. The model is compiled using the Adam optimizer and Mean Squared Error (MSE) loss function, which is appropriate for regression tasks..

4.3 Model Training Module

Once the ANN architecture is defined, this module trains the model using the preprocessed dataset. It specifies training parameters such as the number of epochs, batch size, and validation split. During training, the module tracks training and validation loss to monitor the learning process and detect signs of overfitting or underfitting.

4.4 Model Evaluation Module

This module assesses the performance of the trained salary prediction model. It uses regression metrics such as Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and R^2 score to measure prediction accuracy. The module also interprets these metrics to determine how well the model generalizes to unseen data.

4.5 Visualization Module

This module generates visual results to help users understand the model’s behaviour and performance. It produces training vs. validation loss graphs, error distribution plots, and predicted vs. actual salary comparison charts. These visualizations improve interpretability and provide insights into the effectiveness of the model.

4.6 Prediction Module

The final module applies the trained ANN model to new input data to generate salary predictions. It takes user-provided attributes such as experience, education level, job role, and skills, processes them through the trained model, and outputs the estimated salary value. This module ensures fast, automated, and reliable predictions suitable for HR analytics and decision-support systems.

4.7Data Splitting Module

This module divides the salary dataset into training and testing sets to ensure proper model evaluation. An 80:20 split is used, where 80% of the data trains the ANN model and 20% is reserved for testing. For datasets containing categorical or diverse features, the split ensures that the data distribution is preserved, preventing bias during evaluation. This module ensures the model is evaluated fairly and accurately on unseen salary data, improving the reliability of performance metrics.

4.8 Feature Scaling Module

This module performs normalization of all numerical input features using techniques such as StandardScaler. Attributes like years of experience, age, and skill ratings can vary significantly in scale, and unscaled values may negatively affect ANN training. By converting the features into a standard normal distribution, the module enhances model stability, speeds up convergence, and improves overall training efficiency. Feature scaling ensures that no single feature dominates the learning process

4.9 Output Interpretation Module

This module handles the interpretation and display of the final salary prediction results. It converts the raw predicted values into meaningful outputs that users can easily understand. The module may also provide additional details such as prediction error, confidence insights, or comparison with actual salary trends. This ensures that HR professionals or end users can clearly interpret the model’s output for real-world decision-making.

5 IMPLEMENTATION

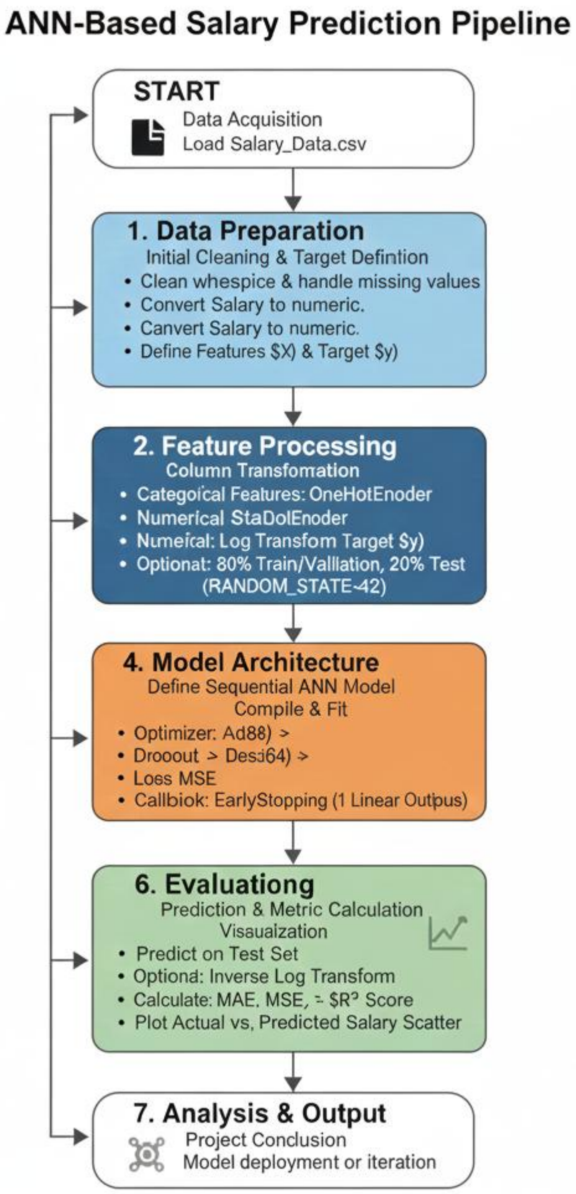
The implementation of the salary prediction system is carried out using Python and an ANN-based regression model. The salary dataset is first loaded into a Pandas DataFrame, and the input features—such as experience, age, education level, job role, and skills—are separated from the target salary value. Categorical features are encoded, and numerical features are standardized to ensure consistent scaling.

The dataset is then split into training and testing sets using an 80–20 ratio. StandardScaler is applied to normalize all numerical inputs, improving the learning efficiency of the neural network. An ANN model is built using TensorFlow/Keras, consisting of an input layer, dense hidden layers with ReLU activation, a dropout layer to reduce overfitting, and a final output layer with linear activation for continuous salary prediction.

The model is compiled using the Adam optimizer and Mean Squared Error (MSE) as the loss function. It is trained for several epochs with a validation split of 0.2, allowing the network to learn the relationships between input features and salary levels. After training, the model is evaluated on the test set using regression metrics such as MAE, MSE, RMSE, and R^2 score to measure prediction accuracy.

Visualizations such as training vs. validation loss curves and predicted vs. actual salary plots are generated to interpret the model's performance. These graphs help identify learning trends and assess the model’s ability to generalize. Overall, this implementation process—from data preprocessing to visualization—successfully develops a functional ANN model capable of accurate salary prediction.

6 SYSTEM ARCHITECTURE



Input

This stage loads the Salary Dataset (CSV) into the system. It involves reading the file into a Pandas DataFrame and examining its structure and basic statistics. Typical tasks include viewing the first few rows, checking the number of samples and features, identifying numerical and categorical columns, and inspecting data types. The dataset is also checked for issues such as missing values, inconsistent formatting, or outliers. This step helps determine what information is available, such as **years of experience, age, gender, education level, job title, skill ratings**, and the target variable **Salary**, and whether the dataset needs cleaning before model development.

Preprocessing

Preprocessing prepares the raw salary data so the ANN can learn patterns effectively and produce accurate predictions.

• Handle missing or inconsistent values

Identify NaN, empty fields, or improperly formatted values (e.g., commas in salary) and apply a suitable strategy—remove rows, fill with mean/median, or clean and convert them as needed.

• Encode categorical variables

Convert non-numeric fields such as *Gender*, *Education Level*, or *Job Title* into numerical format using OneHotEncoder so the ANN can process them correctly.

• Standardize numerical features

Apply StandardScaler to features like *experience*, *age*, and skill-related columns. This ensures all features share a common scale, preventing large-valued features from dominating learning.

• Train–test split

Divide the dataset into training and testing sets, typically 80:20. This ensures the model is trained on one portion of the data and evaluated on unseen samples to measure generalization performance.

• Data formatting

Ensure all arrays are in suitable formats (float32 / int32) for efficient processing by TensorFlow/Keras.

Preprocessing is a critical stage, as it directly influences model convergence, stability, and final prediction accuracy.

ANN Model Construction

This stage defines the neural network architecture and the compilation settings used for salary prediction.

• Input layer:

Sized according to the number of preprocessed input features generated after encoding and scaling.

• Hidden layers:

Example architecture:
Dense(128, ReLU) → Dropout(0.2) → Dense(64, ReLU) → Dropout(0.15) → Dense(32, ReLU)

These layers learn complex nonlinear relationships between features such as experience, education, and job role. ReLU activation supports efficient gradient flow and helps the model learn deeper patterns.

• Dropout:

Randomly disables a fraction of neurons during training to reduce overfitting and improve the model's generalization ability—especially important when the dataset is limited.

• Output layer:

Dense(1, linear) — produces a continuous salary value, as this is a regression problem. A linear activation is appropriate because the output is not a probability but a real number.

• Compile settings:

Optimizer: Adam (adaptive learning, fast convergence)

Loss: Mean Squared Error (MSE), suitable for regression

Metrics: Mean Absolute Error (MAE) for interpretability

Hyperparameters such as layer sizes, dropout rates, batch size, and learning rate are chosen based on experimentation and can be tuned to improve model performance.

The goal of this stage is to build a model expressive enough to learn salary patterns while being regularized to prevent overfitting and maintain prediction stability.

Training

Training is the stage where the neural network learns by updating its weights to minimize prediction error.

• Fit the model:

Train the ANN for a fixed number of epochs (e.g., 50–100) with a chosen batch size (e.g., 32).

A **validation_split** (e.g., 0.2) is used to track validation loss during training.

• Monitor:

Record **training and validation loss** from the history object.

Watch for signs of overfitting, such as training loss decreasing while validation loss stops improving or increases.

• Callbacks (optional):

- **EarlyStopping:** stops training when validation loss no longer improves.
- **ModelCheckpoint:** saves the best model weights.
- **ReduceLROnPlateau:** reduces learning rate when improvements slow down.

• Hyperparameter tuning:

Adjust epochs, batch size, learning rate, hidden layer sizes, dropout rates, and activation functions to improve accuracy and reduce error.

Training transforms the model's initial weights into a learned function capable of predicting salary based on experience, education, job role, and other features.

Visualization and Prediction

Loss vs Epochs: shows how training and validation loss change over time and indicates model stability.

Actual vs Predicted Plot: compares real salaries with predicted values to assess performance visually.

Error Distribution Plot: shows how far predictions deviate from actual values.

Residual Plot: helps identify patterns or inconsistencies in prediction errors.

These graphs give a clear understanding of the model's learning behavior and accuracy.

Prediction:

The trained ANN is used to predict salaries for the test set or new user inputs. The model outputs a continuous salary value, which can be directly interpreted or compared against expected salary ranges.

7 CODE IMPLEMENTATION

Algorithm: Salary Prediction using Artificial Neural Network (ANN)

Input: Salary Dataset

Output: Predicted Salary Value and Performance Metrics (MAE, MSE, RMSE, R^2)

1. Start

2. Load Dataset

   2.1 Load the salary dataset from the CSV file.

   2.2 Separate the dataset into:

- **Feature matrix X** (all input columns such as experience, education, job title, age, etc.)
- **Target vector y** (Salary column)

3. Preprocess Data

   3.1 Clean and convert data types; remove or handle missing values.

   3.2 One-hot encode categorical features (e.g., Gender, Education Level, Job Title).

   3.3 Normalize numerical features using StandardScaler.

   3.4 Split data into training and testing sets using train_test_split with:

   - **test_size = 0.2**
   - **random_state = 42**

   3.5 Ensure X is float32 and y is float32 for TensorFlow compatibility

4. Build ANN Model

   4.1 Initialize a Sequential model.

   4.2 Add input layer with shape = number of processed features.

   4.3 Add first hidden layer: **Dense(128)** with **ReLU** activation.

   4.4 Add Dropout(0.2) to reduce overfitting.

   4.5 Add second hidden layer: **Dense(64)** with ReLU activation.

   4.6 Add Dropout(0.15).

   4.7 Add third hidden layer: **Dense(32)** with ReLU activation.

   4.8 Add output layer: **Dense(1)** with **Linear activation** for regression output.

5. Compile Model

5.1 Set optimizer = **Adam**.

5.2 Set loss function = **Mean Squared Error (MSE)**.

5.3 Set evaluation metric = **Mean Absolute Error (MAE)**.

6. Train Model

6.1 Train the model using X_train and y_train with:

- **Epochs = 100** (or early stopping will stop earlier)
- **Batch size = 8 or 32**
- **Validation split = 0.1**

6.2 Store training history (training and validation loss).

7. Test Model

7.1 Use the trained model to predict salary values for X_test.

7.2 Flatten predictions for easier evaluation.

8. Evaluate Performance

8.1 Compute **MAE** using mean_absolute_error(y_test, y_pred).

8.2 Compute **MSE and RMSE**.

8.3 Compute **R^2 score** to measure prediction quality.

8.4 Display sample predictions vs. actual salaries.

9. Visualize Results

9.1 Plot **training vs. validation loss** across epochs.

9.2 Plot **Actual vs. Predicted Salary** scatter plot.

9.3 Plot **Error Distribution** to analyze prediction deviations.

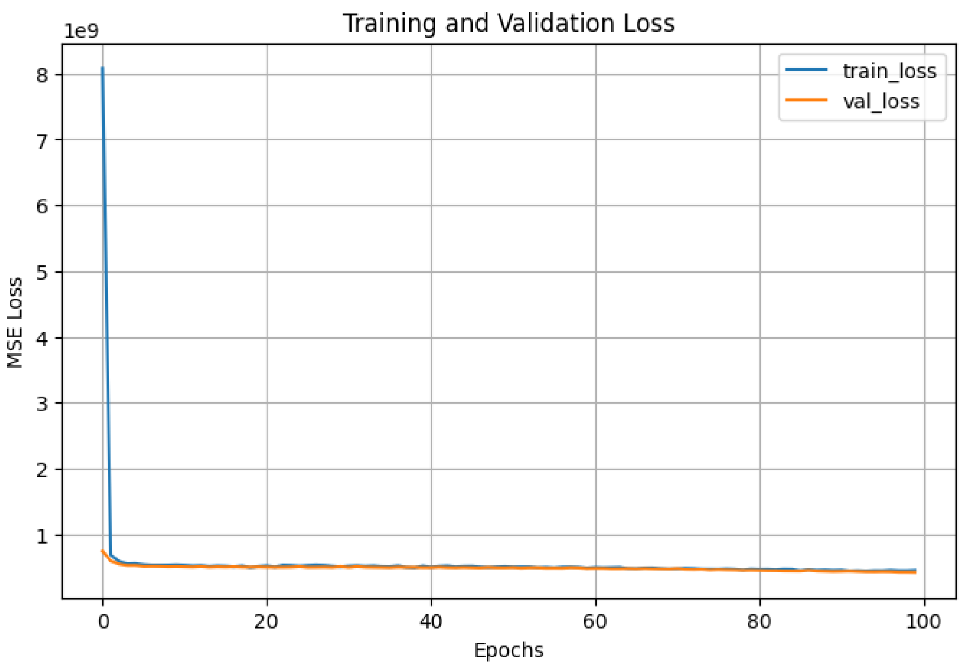
9.4 Plot **Residual Plot** to check consistency of errors.

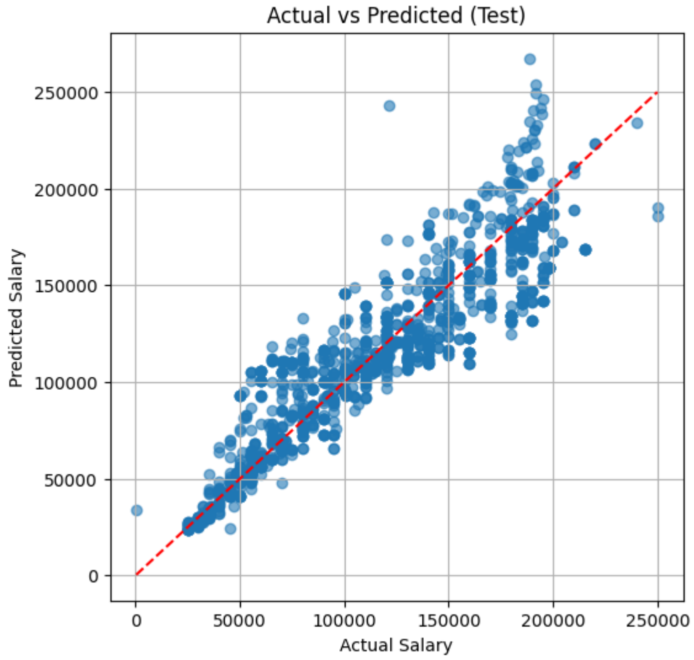
10. End

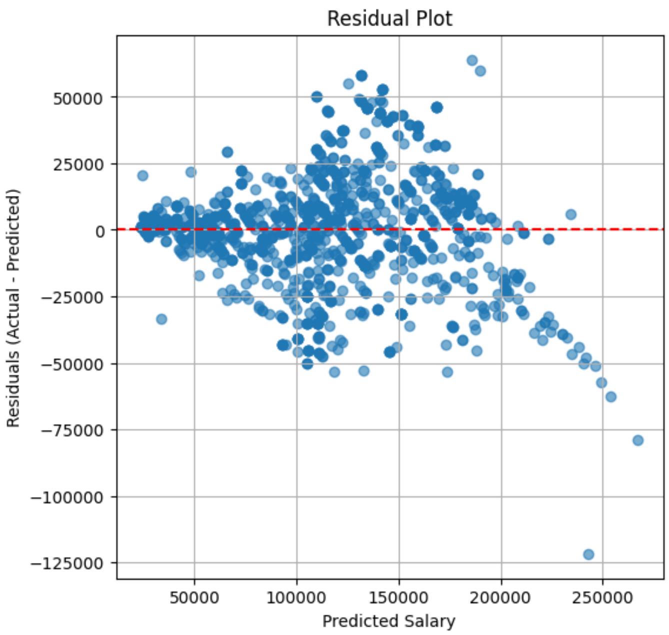
8 RESULT

```
Test MAE: 14335.17
Test MSE: 384188861.58
Test R2 : 0.8608

--- Sample Test Predictions ---
     Actual  Predicted
0  122000.0   115429.0
1   65000.0    70727.0
2  160000.0   162362.0
3  136986.0   125734.0
4  110000.0   130360.0
5  185000.0   137827.0
6   85000.0    95515.0
7  120000.0   117941.0
8   80000.0    76760.0
9  130000.0   117244.0
```




Actual vs Predicted (Test)
Predicted Salary
Actual Salary
0
50000
100000
150000
200000
250000


Residual Plot
Residuals (Actual - Predicted)
Predicted Salary
50000
25000
0
−25000
−50000
−75000
−100000
−125000
100000
150000
200000
250000

9 CONCLUSION

The Artificial Neural Network–based salary prediction system developed in this project demonstrates the capability of deep learning techniques to analyze employee-related data and accurately estimate salary values. By using a structured salary dataset and applying systematic preprocessing, feature encoding, scaling, and ANN model training, the neural network was able o learn meaningful relationships between various factors such as experience, education level, job role, and age. The model produced reliable regression outputs, and evaluation metrics such as MAE, MSE, RMSE, and R^2 score validated its prediction performance.

Visualizations—including training and validation loss curves, actual vs. predicted salary plots, and error distribution graphs—provided deeper insights into the stability and effectiveness of the model. The project highlights that features such as years of experience, education level, and job role play a significant role in determining salary outcomes. Although the system is dataset-dependent and not intended for official HR decision-making, it clearly demonstrates the potential of machine learning models to support automated salary estimation and enhance data-driven workforce analytics.

Overall, the project successfully shows how ANN models can be applied in financial and HR domains, offering a strong foundation for future improvements such as larger datasets, optimization techniques, or integration into real-time salary recommendation platforms. This work reinforces the value of deep learning in predictive analytics and its ability to support informed decision-making in modern organizations.

10 REFERENCES

[1] Sharma et al. (2023). Analysis of machine learning regression models, highlighting the effectiveness of ensemble methods for salary prediction.

[2] Patel & Mehta (2023). Development of an ANN-based salary prediction system demonstrating improved accuracy through proper encoding and normalization.

[3] Zhang et al. (2022). Comparative study of compensation forecasting models across industries, emphasizing the importance of nonlinear learning techniques.

[4] Khan & Ali (2022). Evaluation of regression algorithms such as Support Vector Regression and Random Forest for employee salary estimation.

[5] Fernandes et al. (2023). Study of deep learning architectures for wage prediction, showing the ability of ANN models to learn complex salary patterns.

[6] Reddy (2024). Proposal of a hybrid XGBoost–Neural Network model for enhanced salary prediction accuracy.

[7] Gupta & Roy (2022). Comparison of Decision Tree, SVR, and ANN models for salary forecasting using experience-based datasets.

[8] Kaggle. (2024). Salary datasets used for machine learning–based salary prediction.

[9] TensorFlow Developers. (2015–2024). TensorFlow deep learning framework used for implementing ANN models.

[10] Scikit-Learn Developers. (2011–2024). Scikit-learn library used for preprocessing, encoding, scaling, and regression evaluation metrics.