EMPLOYEE SALARY ANALYSIS AND PREDICTION USING MACHINE LEARNING ALGORITHMS

A Project Report submitted in the partial fulfillment of the

Requirements for the award of the degree

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

Submitted by

K Narasimha charyulu (20471A05L5)

A madhava rao (20471A05N6)

U Sai Kumar (20471A05N4)

Under the esteemed guidance of

M. Suneetha. M.Tech

Asst.Professor

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

NARASARAOPETA ENGINEERING COLLEGE: NARASARAOPETA

(AUTONOMOUS)

Accredited by NAAC with A+ Grade and NBA under Tier -1 Approved by AICTE, New Delhi, Permanently Affiliated to JNTUK, Kakinada

KOTAPPAKONDA ROAD, YALAMANDA VILLAGE, NARASARAOPET- 522601

2023-2024

NARASARAOPETA ENGINEERING COLLEGE

(AUTONOMOUS)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the project that is entitled "Employee Salary Analysis and Prediction Using Machine Learning Algorithms" is a bonafide work done by the team K.Narasimha charyulu (20471A5L5), A.Madhava rao (20471A05N6), U.`Sai Kumar (20471A05N4) in partial fulfillment of the requirements for the award of the degree of BACHELOR OF TECHNOLOGY in the Department of COMPUTER SCIENCE AND ENGINEERING during 2023-2024.

PROJECT GUIDE

M.Suneetha. M.Tech
Asst.professor

PROJECT CO-ORDINATOR

M.Sireesha, M.Tech., Ph.D.,
Assoc. Professor

HEAD OFTHE DEPARTMENT

Dr. S. N. Tirumala Rao, M.Tech., Ph.D.,
Professor & HoD

EXTERNAL EXAMINER

DECLARATION

We declare that this project work titled “EMPLOYEE SALARY ANALYSIS AND PREDICTION USING MACHINE LEARNING” is composed by our self that the work contain here is our own except where explicitly stated otherwise in the text and that this work has been submitted for any other degree or professional qualification except as specified.

K.Narasimha charyulu (20471A05L5)
A. Madhava rao (20471A05N6)
U. Sai Kumar (20471A05N4)

ACKNOWLEDGEMENT

We wish to express my thanks to carious personalities who are responsible for the completion of the project. We are extremely thankful to our beloved chairman sri **M.V.Koteswara Rao,B.Sc.,** who took keen interest in us in every effort throughout this course. We owe out sincere gratitude to our beloved principal **Dr.M.Sreenivasa Kumar,M.Tech., Ph.D., MISTE., FIE(I).,** for showing his kind attention and valuable guidance throughout the course.

We express our deep felt gratitude towards **Dr.S.N.Tirumala Rao, M.Tech.,Ph.D.,** HoD of CSE department and also to our guide **M. Suneetha, M.Tech.,Ph.D.,** of CSE department whose valuable guidance and unstinting encouragement enable us to accomplish our project successfully in time.

We extend our sincere thanks towards **Ms Sireesha.M, M.Tech.,** Associate professor & Project coordinator of the project for extending her encouragement. Their profound knowledge and willingness have been a constant source of inspiration for us throughout this project work.

We extend our sincere thanks to all other teaching and non-teaching staff to department for their cooperation and encouragement during our B.tech degree.

We have no words to acknowledge the warm affection, constant inspiration and encouragement that we received from our parents.

We affectionately acknowledge the encouragement received from our friends and those who involved in giving valuable suggestions had clarifying out doubts which had really helped us in successfully completing our project.

By

K Narasimha charyulu (20471A05L5)

A Madhava Rao (20471A05N6)

U Sai Kumar (20471A05N4)

NARASARAOPETA ENGINEERING COLLEGE
(AUTONOMOUS)

INSTITUTE VISION AND MISSION

INSTITUTION VISION

To emerge as a Centre of excellence in technical education with a blend of effective student centric teaching learning practices as well as research for the transformation of lives and community,

INSTITUTION MISSION

M1: Provide the best class infra-structure to explore the field of engineering and research

M2: Build a passionate and a determined team of faculty with student centric teaching, imbibing experiential, innovative skills

M3: Imbibe lifelong learning skills, entrepreneurial skills and ethical values in students for addressing societal problems

NARASARAOPETA ENGINEERING COLLEGE
(AUTONOMOUS)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

VISION OF THE DEPARTMENT

To become a center of excellence in nurturing the quality Computer Science & Engineering professionals embedded with software knowledge, aptitude for research and ethical values to cater to the needs of industry and society.

MISSION OF THE DEPARTMENT

The department of Computer Science and Engineering is committed to

M1: Could the students to become Software Professionals, Researchers and Entrepreneurs by providing advanced laboratories.

M2: Impart high quality professional training to get expertise in modern software tools and technologies to cater to the real time requirements of the Industry.

M3: Inculcate team work and lifelong learning among students with a sense of societal and ethical responsibilities.

NARASARAOPETA ENGINEERING COLLEGE
(AUTONOMOUS)

Program Specific Outcomes (PSO's)

PSO1: Apply mathematical and scientific skills in numerous areas of Computer Science and Engineering to design and develop software-based systems.

PSO2: Acquaint module knowledge on emerging trends of the modern era in Computer Science and Engineering

PSO3: Promote novel applications that meet the needs of entrepreneur, environmental and social issues.

NARASARAOPETA ENGINEERING COLLEGE
(AUTONOMOUS)

Program Educational Objectives (PEO's)

The graduates of the program are able to:

PEO1: Apply the knowledge of Mathematics, Science and Engineering fundamentals to identify and solve Computer Science and Engineering problems.

PEO2: Use various software tools and technologies to solve problems related to academia, industry and society.

PEO3: Work with ethical and moral values in the multi-disciplinary teams and can communicate effectively among team members with continuous learning.

PEO4: Pursue higher studies and develop their career in software industry.

NARASARAOPETA ENGINEERING COLLEGE
(AUTONOMOUS)

Program Outcomes

1. Engineering knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

2. Problem analysis: Identify, formulate, research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

3. Design/development of solutions: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

4. Conduct investigations of complex problems: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

5. Modern tool usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.

6. The engineer and society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

7. Environment and sustainability: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

8. Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

9. Individual and team work: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

10. Communication: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

11. Project management and finance: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

12. Life-long learning: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

NARASARAOPETA ENGINEERING COLLEGE
(AUTONOMOUS)

Project Course Outcomes (CO'S):

CO421.1: Analyze the System of Examinations and identify the problem.

CO421.2: Identify and classify the requirements.

CO421.3: Review the Related Literature

CO421.4: Design and Modularize the project

CO421.5: Construct, Integrate, Test and Implement the Project.

CO421.6: Prepare the project Documentation and present the Report using appropriate method.

Course Outcomes – Program Outcomes mapping

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
C421.1		√											√		
C421.2	√		√		√								√		
C421.3				√		√	√	√					√		
C421.4			√			√	√	√					√	√	
C421.5					√	√	√	√	√	√	√	√	√	√	√
C421.6									√	√	√		√	√	

Course Outcomes – Program Outcome correlation

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
C421.1	2	3											2		
C421.2			2		3								2		
C421.3				2		2	3	3					2		
C421.4			2			1	1	2					3	2	
C421.5					3	3	3	2	3	2	2	1	3	2	1
C421.6									3	2	1		2	3	

Note: The values in the above table represent the level of correlation between CO's and PO's:

1. **Low level**

2. **Medium level**

3. **High level**

Project mapping with various courses of Curriculum with Attained PO's:

Name of the course from which principles are appliedin this project	Description of the device	Attained PO
C2204.2, C22L3.2	Gathering the requirements and defining the problem, plan to develop a < **Employee Salary Analysis and Prediction Using Machine Learning Algorithms .>**	PO1, PO3
CC421.1, C2204.3, C22L3.2	Each and every requirement is critically analyzed, the process model is identified and divided into <**Six modules>**	PO2, PO3
CC421.2, C2204.2, C22L3.3	Logical design is done by using the unified modelling language which involves individual team work	PO3, PO5, PO9
CC421.3, C2204.3, C22L3.2	Each and every module is tested, integrated, and evaluated in our project	PO1, PO5
CC421.4, C2204.4, C22L3.2	Documentation is done by all our four members in the form of a group	PO10
CC421.5, C2204.2, C22L3.3	Each and every phase of the work in group is presented periodically	PO10, PO11
C2202.2, C2203.3, C1206.3, C3204.3, C4110.2	< **Employee Salary Analysis leverages diverse data for accurate trait prediction. Through preprocessing and model training with algorithms like Random Forest and Ada Boost, we achieved high accuracy. The system is now operational and capable of real-time predictions>.**	PO4, PO7
C32SC4.3	**<The physical design includes hardware components like sensors, gsm module, software and Arduino.>**	PO5, PO6

ABSTRACT

This project aims to predict job salaries using various machine learning regression models. We introduce and explore regression models such as Linear Regression, Multiple Linear Regression, Lasso Regression, and Polynomial Regression. Our analysis focuses on evaluating these modelsbased on their R2 and RMSE values to identify the most accurate and reliable predictor of employee salaries. The ultimate goal is to employ the best-performing algorithm to predict salary outcomes effectively.

INDEX

1. INTRODUCTION TO MACHINE LEARNING

1.1 Introduction to Machine Learning:

Machine Learning is the field of study that gives computers the potential to learn without being explicitly programmed. ML is one of the most interesting technologies that one would have ever come across. As the name indicates, it gives the computer that makes it most similar to humans: The ability to learn. Machine learning is promptly being used today, perhaps in more places than one would expect.

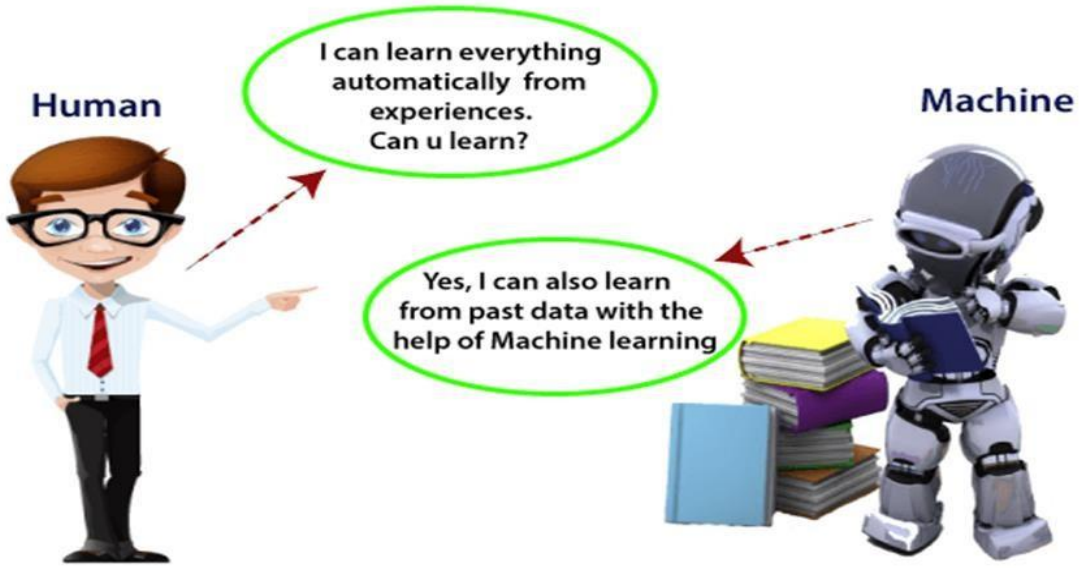
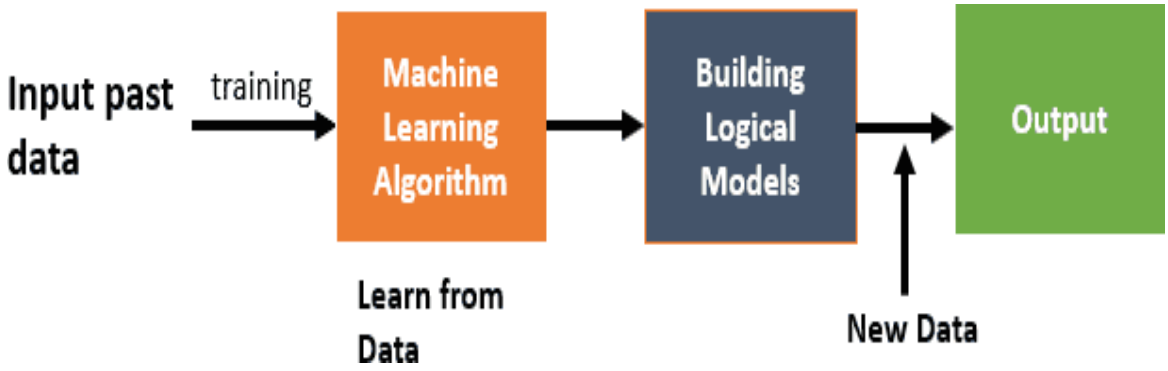


Fig 1.1.1: Machine Learning

Machine learning enables a machine to automatically learn from data, improve performance from experiences, and predict things without being explicitly programmed. A Machine Learning system learns from historical data, builds the prediction models, and whenever it receives new data, predicts the output for it. The accuracy of predicted output depends upon the amount of data, as the huge amount of data helps to build a better model which predicts the output more accurately. With the help of sample historical data, which is known as training data, machine learning algorithms build a mathematical model that helps in making predictions or decisions without being explicitly programmed. Machine learning brings computer science and statistics together for creating predictive models. Machine learning constructs or uses the algorithms that learn from historical data. The more we will provide the information, the higher will be the performance.



1.1.1 Classification of Machine Learning

At a broad level, machine learning can be classified into three types:

a. Supervised learning

b. Unsupervised learning

c. Reinforcement learning

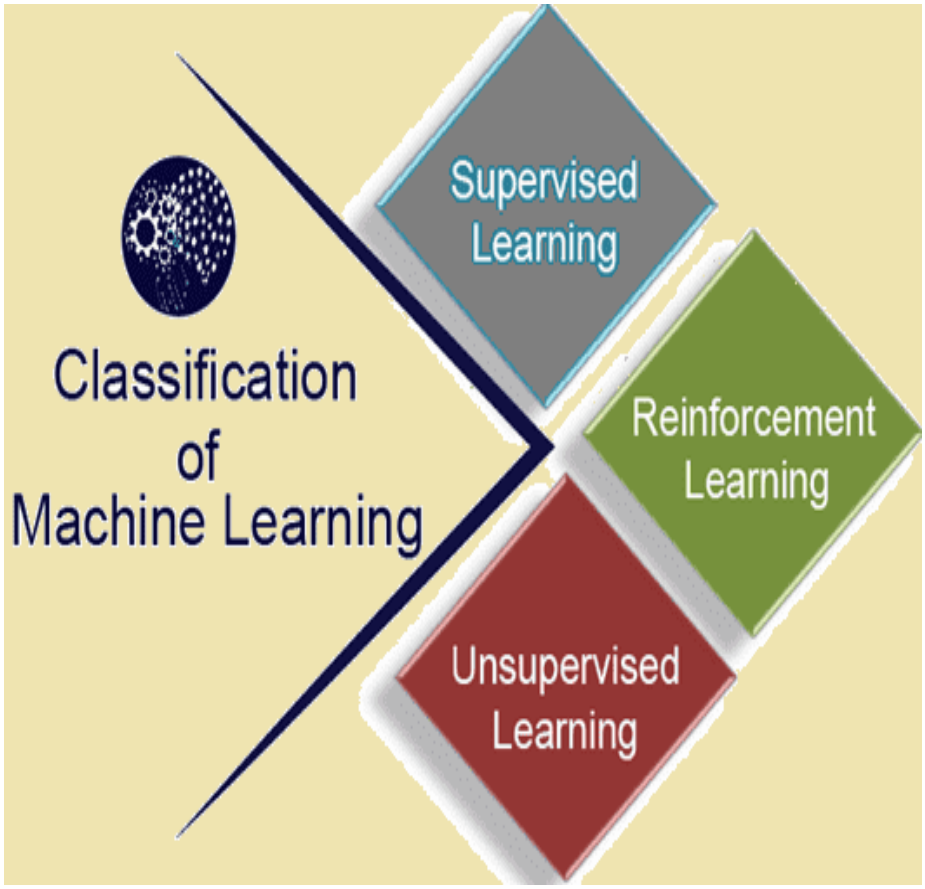


Fig 1.1.1.1: Types of Machine Learning

a) Supervised Learning

Supervised learning is a type of machine learning method in which we provide sample labelled data to the machine learning system in order to train it, and on that basis, it predicts the output. The system creates a model using labelled data to understand the datasets and learn about each data, once the training and processing are done then we test the model by providing a sample data to check whether it is predicting the exact output or not. Supervised learning can be grouped further in two categories of algorithms:

- Classification.
- Regression.

➢ Classification:

The Classification algorithm is a Supervised Learning technique that is used to identify the category of new observations on the basis of training data. In Classification, a program learns from the given dataset or observations and then classifies new observation into a number of classes or groups. Such as, Yes or No, 0 or 1, Spam or Not Spam, cat or dog, etc. Classes can be called as targets/labels or categories.

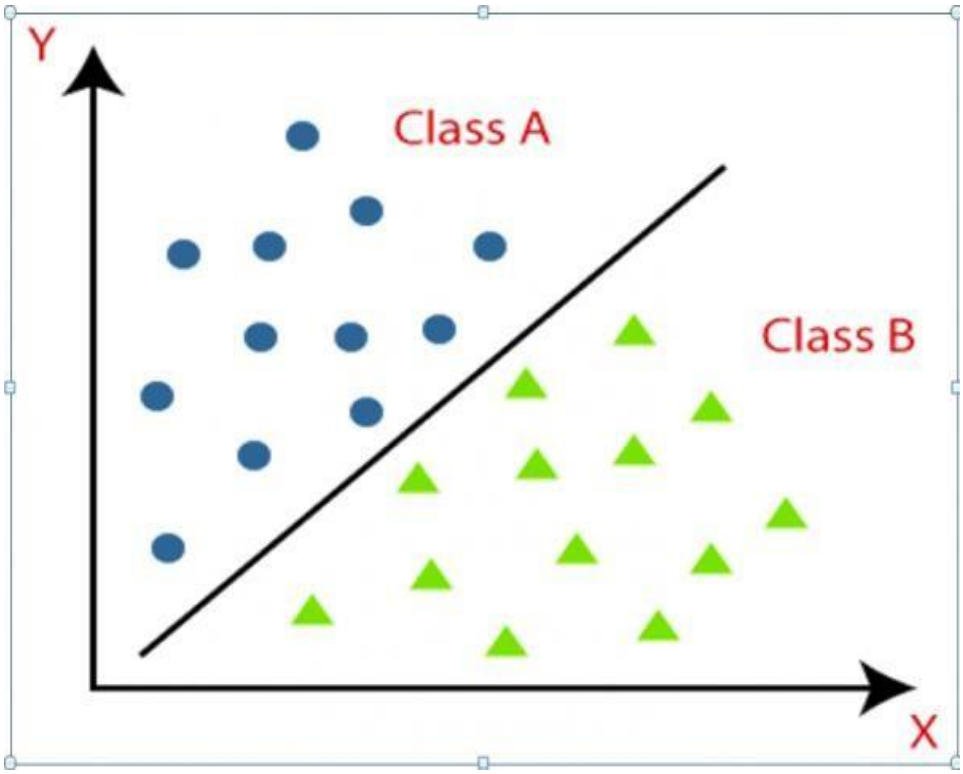


Fig 1.1.1.2: Classification

➢ Regression:

A regression problem is when the output variable is a real or continuous value, such as “salary” or “weight”. Many different models can be used, the simplest is the linear regression. It tries to fit data with the best hyper-plane which goes through the points.

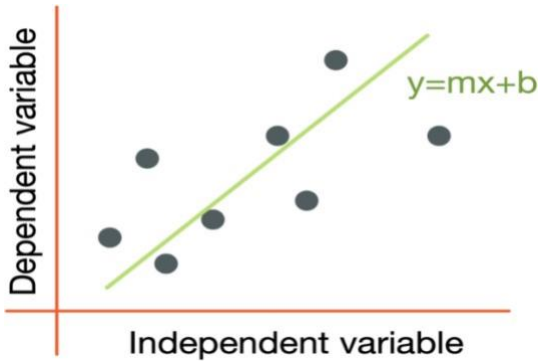


Fig 1.1.1.3: Regression

b) Unsupervised Learning

Unsupervised learning is a learning method in which a machine learns without any supervision. The training is provided to the machine with the set of data that has not been labelled, classified, or categorized, and the algorithm needs to act on that data without any supervision. The goal of unsupervised learning is to restructure the input data into new features or a group of objects with similar patterns. It can be further classifieds into two categories of algorithms:

- Clustering.
- Association.

c) Reinforcement Learning

Reinforcement learning is a feedback-based learning method, in which a learning agent gets a reward for each right action and gets a penalty for each wrong action. The agent learns automatically with these feedbacks and improves its performance. In reinforcement learning, the agent interacts with the environment and explores it. The goal of an agent is to get the most reward points, and hence, it improves its performance. The robotic dog, which automatically learns the movement of his arms, is an example of Reinforcement learning.

1.2 Introduction to Employee Salary Analysis And Prediction

This project aims to address a common challenge faced by college graduates when entering the job market: the difficulty of assessing job offers and negotiating salaries effectively. By utilizing various regression models on a dataset containing salary information, the goal is to create a predictive model that can estimate salaries for different job positions. This approach empowers graduates to evaluate job offers more objectively and negotiate for salaries that align better with their expectations and qualifications. Ultimately, this predictive tool can provide valuable insights to graduates, helping them make informed decisions and increase their chances of a successful job search.

1.3 Importance of Machine Learning in Employee Salary Analysis

The machine learning algorithms utilized in this project are pivotal in achieving accurate and dependable predictions of job salaries. Polynomial Regression plays a fundamental role in capturing intricate non-linear relationships between features and salaries, providing a basis for comparison with other models. Linear Regression serves as a foundational benchmark, offering insights into the individual impact of features on salary predictions. Multiple Linear Regression enhances accuracy by considering multiple features simultaneously, while Lasso Regression introduces regularization to combat overfitting and improve generalization. Furthermore, Bayesian Ridge Regression contributes robustness against multicollinearity and outliers, bolstering model stability and prediction accuracy, especially in noisy datasets. Together, these algorithms synergize to provide a comprehensive evaluation framework, enabling the selection of the most effective model for precise and reliable salary predictions.

1.4 Implementation of machine learning using Python

Python is a popular programming language. It was created in 1991 by Guido van Rossum. It is used for:

- Web Development
- Software Development
- Mathematics
- System Analysis

The most recent major version of Python is Python 3. However, Python 2, although not being updated with anything other than security updates, is still quite popular. 8 It is possible to write Python in an Integrated Development Environment, such as Thonny, Pycharm, Net beans or Eclipse, Anaconda which are particularly useful when managing larger collections of Python files. Python was designed for its readability. Python uses new lines to complete a command, as opposed to other programming languages which often use semicolons or parentheses. Python relies on indentation, using whitespace, to define scope; such as the scope of loops, functions and classes. Other programming languages often use curly-brackets for this purpose. In the older days, people used to perform Machine Learning tasks manually by coding all the algorithms and mathematical and statistical formula. This made the process time consuming, tedious and inefficient. But in the modern days, it is become very much easy and efficient compared to the olden days by various python libraries, frameworks, and modules. Today, Python is one of the most popular programming languages for this task and it has replaced many languages in the industry, one of the reason is its vast collection of libraries.

Python libraries that used in Machine Learning are:

- Numpy
- Scipy
- Scikit-learn
- Theano
- TensorFlow
- Keras
- PyTorch
- Pandas
- Matplotlib

NumPy is a very popular python library for large multi-dimensional array and matrix processing, with the help of a large collection of high-level mathematical functions. It is very useful for fundamental scientific computations in Machine Learning. It is particularly useful for linear algebra, Fourier transform, and random number capabilities. High-end libraries like TensorFlow uses NumPy internally for manipulation of Tensors.

SciPy is a very popular library among Machine Learning enthusiasts as it contains different modules for optimization, linear algebra, integration and statistics. There is a difference between the SciPy library and the SciPy stack. The SciPy is one of the core packages that make up the SciPy stack. SciPy is also very useful for image manipulation.

Skikit-learn is one of the most popular Machine Learning libraries for classical Machine Learning algorithms. It is built on top of two basic Python libraries, NumPy and SciPy. Scikit-learn supports most of the supervised and unsupervised learning algorithms. Scikitlearn can also be used for data-mining and data-analysis, which makes it a great tool who is starting out with Machine Learning.

Theano is a popular python library that is used to define, evaluate and optimize mathematical expressions involving multi-dimensional arrays in an efficient manner. It is achieved by optimizing the utilization of CPU and GPU. It is extensively used for unit-testing and self verification to detect and diagnose different types of errors. Theano is a very powerful library that has been used in large-scale computationally intensive scientific projects for a longtime but is simple and approachable enough to be used by individuals for their own projects.

TensorFlow is a very popular open-source library for high performance numerical computation developed by the Google Brain team in Google. As the name suggests, Tensor flow is a framework that involves defining and running computations involving tensors. It can train and run deep neural networks that can be used to develop several AI applications. TensorFlow is widely used in the field of deep learning research and application.

Keras is a very popular Machine Learning library for Python. It is a high-level neural networks API capable of running on top of TensorFlow, CNTK, or Theano. It can run seamlessly on both CPU and GPU. Keras makes it really for ML beginners to build and design a Neural Network. One of the best thing about Keras is that it allows for easy and fast prototyping.

2. REQUIREMENTS SPECIFICATION

2.1 SOFTWARE REQUIREMENTS:

- Browser : Google Chrome
- Operating System : Windows7,8,10
- Language : Python
- Platform : Google Colab, jupyter

2.2 HARDWARE REQUIREMENTS:

- Processor : Intel® core i5
- RAM : 6GB
- OS type : 64-bit operating system, x64 processor

3.LITERATURE SURVEY

3.1 LITERATURE SURVEY

Several studies have explored the prediction of salaries using various data mining and machine learning techniques. In [1], the authors utilized regression modeling with salary as the dependent variable and other factors as independent variables. This approach highlights the use of regression in addressing salary prediction problems.

In contrast, [2] focused on a salary prediction method based on data mining techniques, specifically employing a 10-fold cross-validation experiment using graduate student data. The study found that decision tree (J48) outperformed KNN and Naive Bayes in terms of accuracy, indicating the efficacy of decision tree models in salary prediction tasks.

Another study, [3], proposed a prediction model using Decision Tree technique with seven features. Additionally, the system not only predicted salaries but also identified the 3 highest salaries among graduate students with similar attributes. The experiment conducted on a substantial dataset of 13,541 records revealed an accuracy of 41.39%, demonstrating the model's predictive capabilities.

Deep learning techniques were explored in [4] for constructing a salary prediction model specifically tailored for Thailand's labor workforce. By utilizing five months' worth of personal profile data from a job search website, the Deep learning model achieved strong performance with an RMSE of 0.774 x 10^4 and a runtime of only 17 seconds, showcasing its effectiveness in handling regression tasks for salary prediction.

Furthermore, [5] delved into the importance of features contributing to salary forecasts, highlighting attributes such as experience, work security, and specific job positions as significant contributors to workers' compensation levels based on their analysis of the job market.

Finally, [6] introduced a novel approach combining initial feature vectors with log-based feature reduction techniques. This included semantic features indicating negated meaning, universal quantification, trigrams of part of speech, and specific noun phrases, showcasing a comprehensive feature engineering strategy for salary prediction tasks.

Overall, these studies collectively contribute to the understanding of salary prediction models, feature importance, and the efficacy of various machine learning and data mining techniques in addressing salary-related challenges in different contexts

4 SYSTEM ANALYSIS

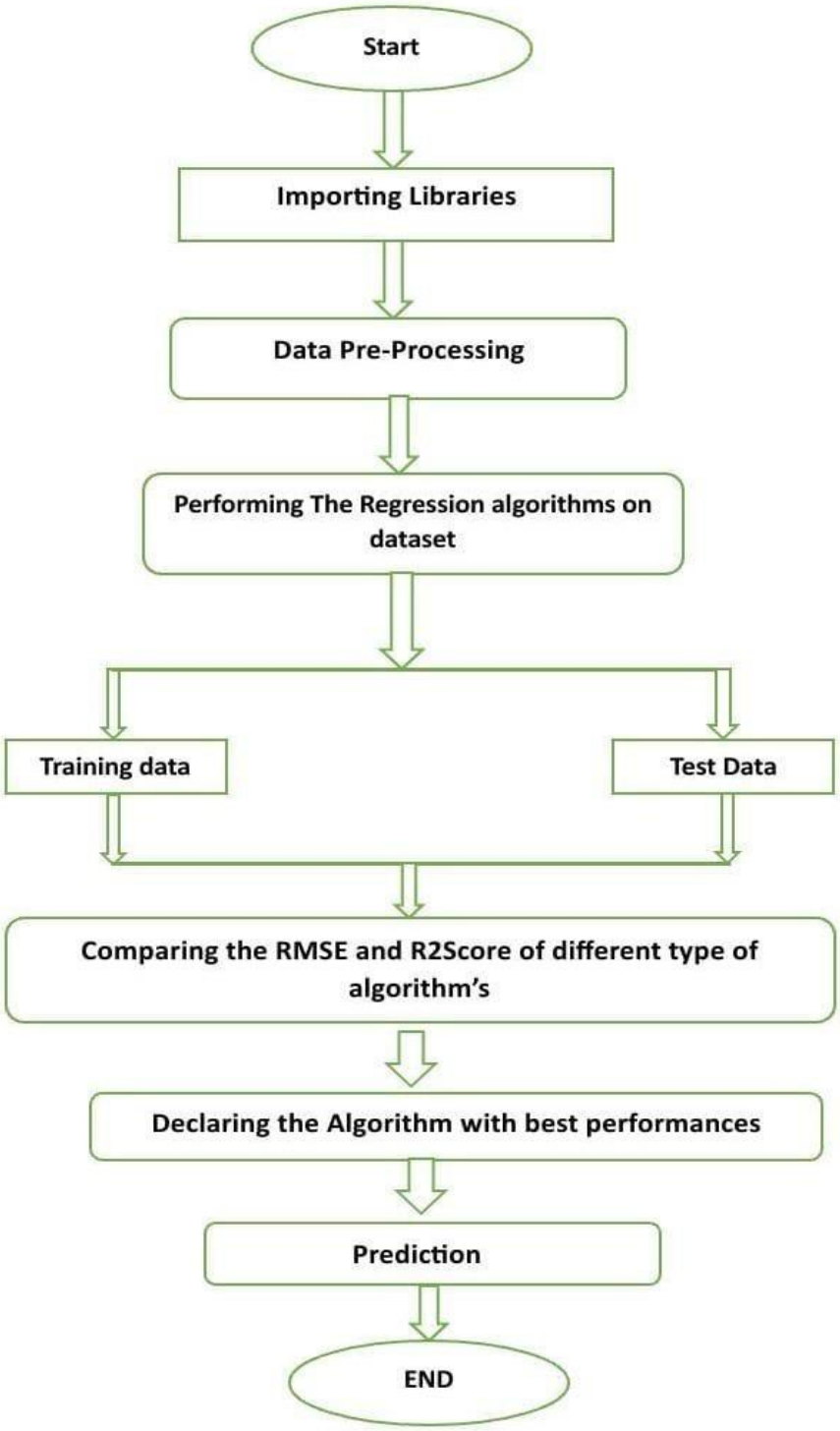
4.1 EXISTING SYSTEM

The current approach in salary prediction involves utilizing Polynomial Regression as the primary algorithm. Polynomial Regression is a popular choice due to its ability to capture non-linear relationships in the data. However, there are limitations to this approach, particularly when dealing with noisy or complex datasets. While Polynomial Regression can provide decent predictions, it may not always yield the best results in terms of accuracy and robustness.

4.2 Proposed system:-

In the proposed system, we aim to enhance the accuracy and reliability of salary prediction by incorporating Bayesian Ridge Regression alongside other regression algorithms. Bayesian Ridge Regression offers advantages such as regularization and handling multicollinearity, which can lead to more stable and accurate predictions, especially in scenarios with noisy or correlated features. By introducing Bayesian Ridge Regression and conducting a comparative analysis with other regression models like Linear Regression, Multiple Linear Regression, and Lasso Regression, we seek to identify the best-performing algorithm for salary prediction. This approach allows for a more comprehensive evaluation and selection of the model with superior performance, ultimately improving the overall quality of salary predictions in the system.

4.3 System Architecture:



4.4 Scope of the project

The scope of this project encompasses a comprehensive exploration of different regression models, including Linear Regression, Multiple Linear Regression, Lasso Regression, and Polynomial Regression. Through detailed analysis using metrics like R2 and RMSE, we aim to determine the most suitable model for predicting job salaries accurately. The project involves developing, implementing, and fine-tuning the selected regression model to ensure precise salary predictions. Additionally, we will evaluate the model's performance, compare it with other regression techniques, and assess its effectiveness in real-world salary prediction scenarios.

4.5 Analysis

1. Sample code number: id number
2.Gender: Male/Female
3. Age: Numbers
4. Salary: Numbers
5. PhD:0/1

4.6 Dataset

TABLE 1. THE ATTRIBUTIONS AND FEATURES OF THE DATASET

	Salary	Gender	Age	PhD
0	140.0	1	47	1
1	30.0	0	65	1
2	35.1	0	56	0
3	30.0	1	23	0
4	80.0	0	53	1

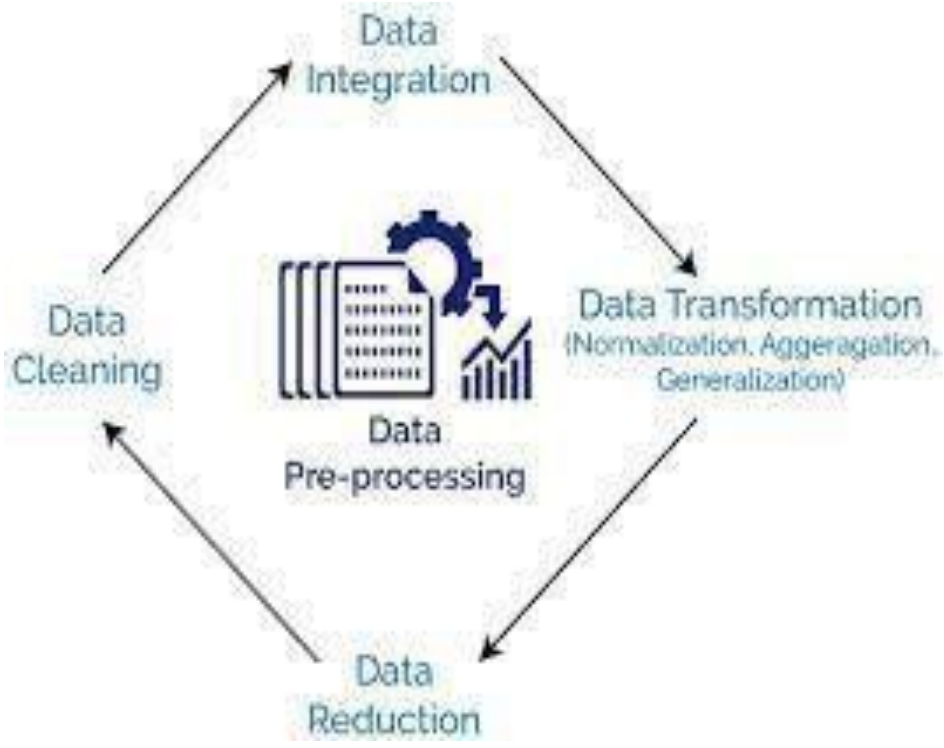
Fig:4.6.1 Dataset

4.7 Data Pre-processing

Before feeding data to an algorithm we have to apply transformations to our data which is referred as pre-processing. By performing pre-processing the raw data which is not feasible for analysis is converted into clean data. In-order to achieve better results using a model in Machine Learning, data format has to be in a proper manner. The data should be in a particular format for different algorithms. For example, if we consider Random Forest algorithm it does not support null values. So that those null values have to be managed using raw data.

Pre-processing refers to the transformations applied to our data before feeding it to the algorithm[7]. Data Pre-processing is a technique that is used to convert the raw data into a clean data set. In other words, whenever the data is gathered from different sources it is collected in raw format which is not feasible for the analysis.

Preprocessing is the first step while creating the machine learning model. It is the process of converting raw dataset into cleaned dataset. Raw data contains noise, missing values, duplicate values which is not suitable for machine learning model. So, preprocessing is required for cleaning the data and making it suitable for machine learning model.



4.7.1 Data Preprocessing

4.8 FEATURE EXTRATION: Feature extraction plays a crucial role in the success of regression projects, especially when dealing with complex datasets. In this project, feature extraction involves identifying and creating meaningful input variables that capture the essence of the data and improve the predictive power of regression models.

One of the key aspects of feature extraction is selecting relevant features that have a significant impact on the target variable (e.g., housing prices, stock market trends). This process often starts with exploratory data analysis (EDA) to understand the relationships between different features and the target variable. Correlation analysis, statistical tests, and domain knowledge are utilized to identify potential features for extraction.

In addition to selecting features, feature engineering techniques are applied to transform and enhance the existing features. This may include scaling numerical features to a common range, encoding categorical variables into numerical representations, creating interaction terms between features, and deriving new features through mathematical transformations or domain-specific knowledge.

Overall, effective feature extraction in this project aims to improve the model's predictive performance, reduce noise, handle multicollinearity, and enhance the interpretability of regression results. It involves a combination of data exploration, feature engineering, and dimensionality reduction techniques tailored to the characteristics of the dataset and the objectives of the regression analysis.

4.9. CORELATION:

Correlation is the statistical measure of the relationship between two variables. There are different types of correlation coefficients like Pearson coefficient (linear) and Spearman coefficient (non-linear) which capture different degrees of probabilistic dependence but not necessarily causation. The correlation coefficient, or Pearson’s, is calculated using a least-squares measure of the error between an estimating line and the actual data values, normalized by the square root of their variances. The coefficients range in value from -1 (perfect inverse correlation) to 1 (perfect direct correlation), with zero being no correlation.

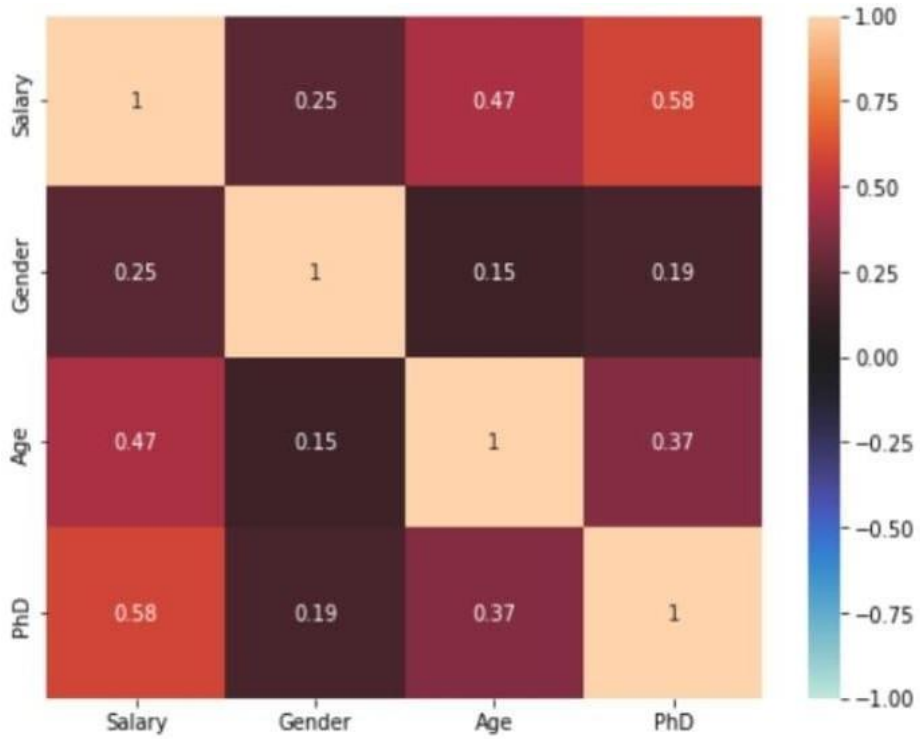


Figure:4.9.1 Correlation for Employee Salary Analysis And Prediction

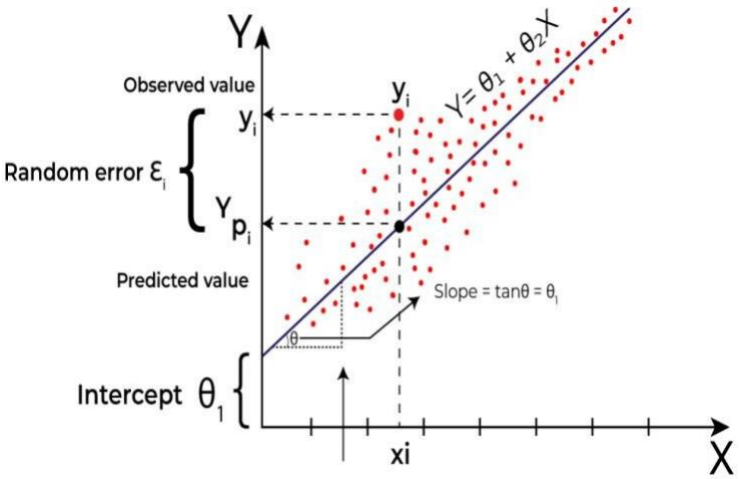
4.10 REGRESSION:

Regression analysis is a statistical method to model the relationship between a dependent (target) and independent (predictor) variables with one or more independent variables. More specifically, Regression analysis helps us to understand how the value of the dependent variable is changing corresponding to an independent variable when other independent variables are held fixed. It predicts continuous/real values such as **temperature, age, salary, price,** etc.. Some of regression algorithms are:

a. Linear regression
b. Multi Linear Regression
c. Ridge Regression
d. Lasso Regression
e. Elastic-net Regression
f. Polynomial Regression
g. Bayesian Ridge Regression

a. Linear Regression :

Machine Learning is a branch of Artificial intelligence that focuses on the development of algorithms and statistical models that can learn from and make predictions on data. **Linear regression** is also a type of machine-learning algorithm more specifically a **supervised machine-learning algorithm** that learns from the labelled datasets and maps the data points to the most optimized linear functions. which can be used for prediction on new datasets.



4.10.1 Linear Regression

Here Y is called a dependent or target variable and X is called an independent variable also known as the predictor of Y. There are many types of functions or modules that can be used for regression. A linear function is the simplest type of function. Here, X may be a single feature or multiple features representing the problem. Linear regression performs the task to predict a dependent variable value (y) based on a given independent variable (x)). Hence, the name is Linear Regression. In the figure above, X (input) is the work experience and Y (output) is the salary of a person. The regression line is the best-fit line for our model.

b. Multiple Linear Regression:

Multiple Linear Regression attempts to model the relationship between two or more features and a response by fitting a linear equation to observed data. The steps to perform multiple linear Regression are almost similar to that of simple linear Regression. The Difference Liesin the evaluation. We can use it to find out which factor has the highest impact on the predictedoutput and how different variables relate to each other.

c. Ridge Regression:

Ridge Regression, also known as Tikhonov regularization, is a technique used to analyze data afflicted by multicollinearity, a phenomenon where independent variables in a linear regression model are highly correlated. When multicollinearity is present, the estimated regression coefficients can become unstable, and the analysis overly sensitive to small variations in the data. This is where Ridge Regression comes in.

Unlike ordinary linear regression, which seeks to minimize the sum of the squares of the residuals (the difference between the observed and predicted values by the model), Ridge Regression adds a penalty term to the calculation. This penalty term is proportional to the square of the model coefficients, also known as the L2 norm of the coefficients. Mathematically, this translates to minimizing the following cost function:

$$J(\theta) = MSE(\theta) + \lambda \sum_{i=1}^{n} \theta_i^2$$

Where:

- $J(\theta)$ is the Ridge cost function to be minimized.
- MSE(θ) is the mean squared error.
- λ is the regularization hyperparameter.
- θi are the model coefficients, with i ranging from 1 to n, the number of independent variables.

d. Lasso Regression:

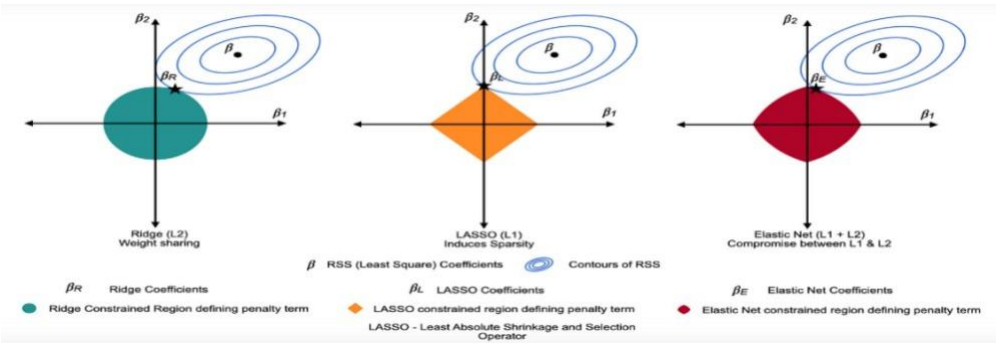
Lasso regression is another regularization technique to reduce the complexity of the model. It stands for **Least Absolute and Selection Operator.** It is similar to the Ridge Regression except that the penalty term contains only the absolute weights instead of a square of weights. Since it takes absolute values, hence, it can shrink the slope to 0, whereas Ridge Regression can only shrink it near to 0.It is also called as **L1 regularization.** The equation for the cost function of Lasso regression will be:

$$\sum_{i=1}^{M}(y_i - y'_i)^2 = \sum_{i=1}^{M}\left(y_i - \sum_{j=0}^{n}\beta_j * x_{ij}\right)^2 + \lambda\sum_{j=0}^{n}|\beta_j|$$

Some of the features in this technique are completely neglected for model evaluation. Hence, the Lasso regression can help us to reduce the overfitting in the model as well as the feature selection

e . Elastic-net Regression:

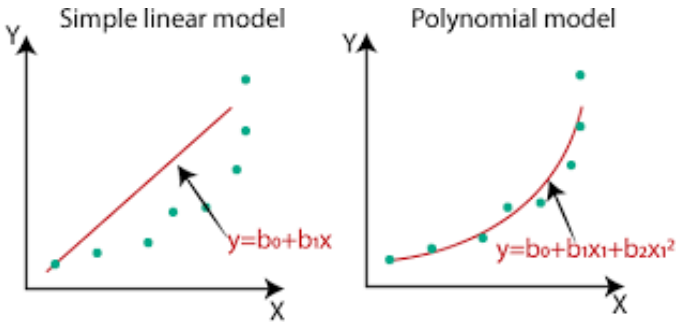
Elastic Net Regression is a powerful machine learning algorithm that combines the features of both Lasso and Ridge Regression. It is a regularized regression technique that is used to deal with the problems of multicollinearity and overfitting, which are common in high-dimensional datasets. This algorithm works by adding a penalty term to the standard least-squares objective function. In this blog, we will dive into the details of Elastic Net Regression, its advantages, and its applications.



4.10.2 Elastic-net Regression

f. Polynomial Regression:

Polynomial Linear Regression is a type of regression analysis in which the relationship between the independent variable and the dependent variable is modeled as an n-th degree polynomial function. Polynomial regression allows for a more complex relationship between the variables to be captured, beyond the linear relationship in Simple and Multiple Linear Regression.



4.10.3 Polynomial Regression

g. Bayesian Ridge Regression:

Bayesian regression allows a natural mechanism to survive insufficient data or poorly distributed data by formulating linear regression using probability distributors rather than point estimates. The output or response 'y' is assumed to drawn from a probability distribution rather than estimated as a single value.

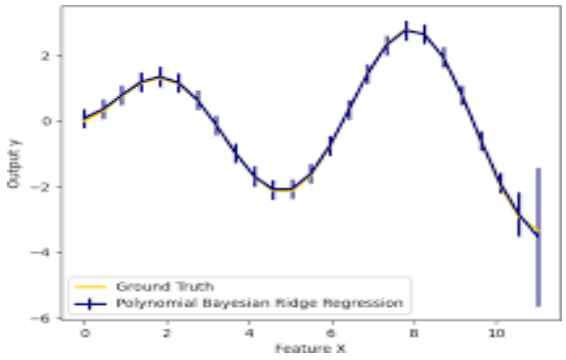
Mathematically, to obtain a fully probabilistic model the response y is assumed to be Gaussian distributed around Xw*X*as follows

$p(y|X,w,\alpha)=N(y|Xw,\alpha)$

One of the most useful type of Bayesian regression is Bayesian Ridge regression which estimates a probabilistic model of the regression problem. Here the prior for the coefficient w is given by spherical Gaussian as follows −

$p(w|\lambda)=N(w|0,\lambda-1Ip)$

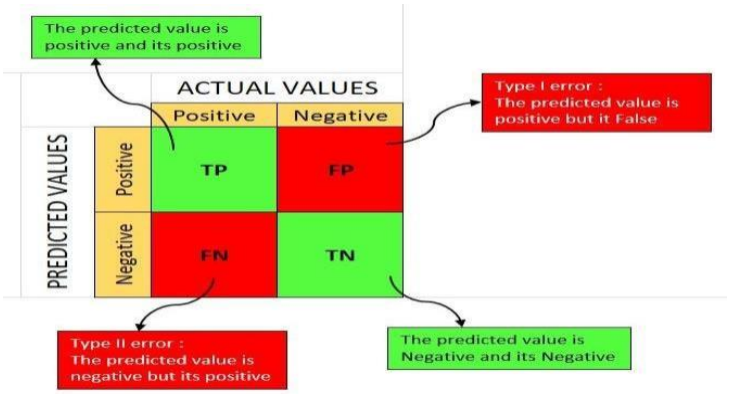
This resulting model is called Bayesian Ridge Regression and in scikit- learn **sklearn.linear_model.BeyesianRidge** module is used for Bayesian Ridge Regression.



4.10.4 Bayesian Ridge Regression

4.11 CONFUSION MATRIX

The confusion matrix is a matrix used to determine the performance of the classification models for a given set of test data. It can only be determined if the true values for test data are known. The matrix itself can be easily understood, but the related terminologies may be confusing. Since it shows the errors in the model performance in the form of a matrix, hence also known as an **error matrix**. Some features of Confusion matrix are given below:



4.11.1 Confusion Matrix

A true positive (tp) is a result where the model predicts the positive class correctly. Similarly, a true negative (tn) is an outcome where the model correctly predicts the negative class. A false positive (fp) is an outcome where the model incorrectly predicts the positive class. And a false negative (fn) is an outcome where the model incorrectly predicts the negative class.

Sensitivity or Recall or hit rate or true positive rate (TPR)

It is the proportion of individuals who actually have the disease were identified as having the disease. TPR = tp / (tp + fn)

Specificity, selectivity or true negative rate (TNR)

It is the proportion of individuals who actually do not have the disease were identified as not having the disease. TNR = tn / (tn + fp) =1-FPR

Precision or positive predictive value (PPV)

If the test result is positive what is the probability that the patient actually has the disease.

PPV = tp / (tp + fp)

Negative predictive value (NPV)

If the test result is negative what is the probability that the patient does not have disease.

NPV = tn / (tn + fn)

Miss rate or false negative rate (FNR)

It is the proportion of the individuals with a known positive condition for which the test result is negative.

FNR = fn / (fp + tn)

Fall-out or false positive rate (FPR)

It is the proportion of all the people who do not have the disease who will be identified as having the disease.

FPR = fp/ (fp + tn)

False discovery rate (FDR)

It is the proportion of all the people identified as having the disease who do not have the disease.

FDR = fp / fp + tp

False omission rate (FOR)

It is the proportion of the individuals with a negative test result for which the true condition is positive.

FOR = fn / (fn + tn)

Accuracy

The accuracy reflects the total proportion of individuals that are correctly classified.

Accuracy = (tp + tn) / (tp + tn + fp + fn)

5. Implementation

5.1 Implementation Code

```
import math
import numpy as np
import pandas as pd
from sklearn.linear_model import BayesianRidge
import seaborn as sns
from IPython.display import display
from statsmodels.formula import api
from sklearn.feature_selection import RFE
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestRegressor
from statsmodels.stats.outliers_influence import variance_inflation_factor
from sklearn.decomposition import PCA
from sklearn.linear_model import Ridge
from sklearn.linear_model import Lasso
from sklearn.linear_model import ElasticNet
from sklearn.linear_model import LinearRegression
from sklearn.ensemble import RandomForestClassifier
from sklearn.preprocessing import PolynomialFeatures
from sklearn.metrics import r2_score, mean_absolute_error, mean_squared_error
import matplotlib.pyplot as plt
plt.rcParams['figure.figsize'] = [10,6]

import warnings
warnings.filterwarnings('ignore')
df = pd.read_csv('/content/Salary.csv')
display(df.head())

target = 'Salary'
features= [i for i in df.columns if i not in [target]]
print(features)
```

```
original_df = df.copy(deep=True)

df.info()
df.nunique().sort_values()



nu = df[features].nunique().sort_values()
nf = []; cf = []; nnf = 0; ncf = 0; #numerical & categorical features

for i in range(df[features].shape[1]):
    if nu.values[i]<=2:cf.append(nu.index[i])
    else: nf.append(nu.index[i])

print('Inference: The Datset has {} numerical & {} categorical
features.'.format(len(nf),len(cf)))

display(df.describe())

plt.figure(figsize=[8,4])
sns.distplot(df[target], color='g',hist_kws=dict(edgecolor="black", linewidth=2), bins=30)
plt.title('Target Variable Distribution - Median Value of Homes ($1Ms)')
plt.show()

#Visualising the numeric features

print('Numeric Features Distribution'.center(100))
plt.figure(figsize=[15,3])
plt.subplot(1,4,1)
sns.distplot(df[nf[0]],hist_kws=dict(edgecolor="black", linewidth=2), bins=10,
color=list(np.random.randint([255,255,255])/255))
plt.tight_layout()
plt.show()
plt.figure(figsize=[15,3])
```

```
plt.subplot(1,4,1)
df.boxplot(nf[0])
plt.tight_layout()
plt.show()

for feature in features:
    print('Histogram of ', feature)
    plt.hist(df[feature])
    plt.show()



df.boxplot(column=features[1])
plt.show()

df[features].corr()

g =  sns.pairplot(df)
plt.title('Pairplots for all the Feature')

plt.show()

sns.set()
sns.heatmap(df[features].corr(), annot=True, fmt='.1f')
plt.yticks(rotation = 0)

counter = 0
rs,cs = original_df.shape

df.drop_duplicates(inplace=True)

if df.shape==(rs,cs):
    print('No duplicates')
else:
    print(f' Number of duplicates dropped/fixed ---> {rs-df.shape[0]}')
```

```
nvc = pd.DataFrame(df.isnull().sum().sort_values(), columns=['Total Null Values'])
nvc['Percentage'] = round(nvc['Total Null Values']/df.shape[0],3)*100
print(nvc)

df3 = df.copy()
df1 = df3.copy()

#features1 = [i for i in features if i not in ['CHAS','RAD']]
features1 = nf

for i in features1:
    Q1 = df1[i].quantile(0.25)
    Q3 = df1[i].quantile(0.75)
    IQR = Q3 - Q1
    df1 = df1[df1[i] <= (Q3+(1.5*IQR))]
    df1 = df1[df1[i] >= (Q1-(1.5*IQR))]
    df1 = df1.reset_index(drop=True)
display(df1.head())
print('\n\033[1mInference:\033[0m\nBefore removal of outliers, The dataset had {}
samples.'.format(df3.shape[0]))
print('After removal of outliers, The dataset now has {} samples.'.format(df1.shape[0]))

m=[]
for i in df.columns.values:
    m.append(i.replace(' ','_'))

df.columns = m
X = df.drop([target],axis=1)
Y = df[target]
Train_X, Test_X, Train_Y, Test_Y = train_test_split(X, Y, train_size=0.8, test_size=0.2,
random_state=100)
Train_X.reset_index(drop=True,inplace=True)
```

```
print('Original set ---> ',X.shape,Y.shape,'\nTraining set --->
',Train_X.shape,Train_Y.shape,'\nTesting set  ---> ', Test_X.shape,'', Test_Y.shape)

std = StandardScaler()

print('\033[1mStandardardization on Training set'.center(120))
Train_X_std = std.fit_transform(Train_X)
Train_X_std = pd.DataFrame(Train_X_std, columns=X.columns)
display(Train_X_std.describe())

print('\n','\033[1mStandardardization on Testing set'.center(120))
Test_X_std = std.transform(Test_X)
Test_X_std = pd.DataFrame(Test_X_std, columns=X.columns)
display(Test_X_std.describe())

print('\033[1mCorrelation Matrix'.center(100))
plt.figure(figsize=[8,6])
sns.heatmap(df.corr(), annot=True, vmin=-1, vmax=1, center=0) #cmap='BuGn'
plt.show()

from sklearn.preprocessing import PolynomialFeatures
Trr=[]; Tss=[]; n=3
order=['ord-'+str(i) for i in range(2,n)]
DROP=[];b=[]
LR = LinearRegression()
LR.fit(Train_X_std, Train_Y)

pred1 = LR.predict(Train_X_std)
pred2 = LR.predict(Test_X_std)

Trr.append(np.sqrt(mean_squared_error(Train_Y, pred1)))
Tss.append(np.sqrt(mean_squared_error(Test_Y, pred2)))

for i in range(len(Train_X_std.columns)-1):
```

```
    vif = pd.DataFrame()
    X = Train_X_std.drop(DROP,axis=1)
    vif['Features'] = X.columns
    vif['VIF'] = [variance_inflation_factor(X.values, i) for i in range(X.shape[1])]
    vif['VIF'] = round(vif['VIF'], 2)
    vif = vif.sort_values(by = "VIF", ascending = False)
    vif.reset_index(drop=True, inplace=True)
    if vif.loc[0][1]>1:
        DROP.append(vif.loc[0][0])
        LR = LinearRegression()
        LR.fit(Train_X_std.drop(DROP,axis=1), Train_Y)

        pred1 = LR.predict(Train_X_std.drop(DROP,axis=1))
        pred2 = LR.predict(Test_X_std.drop(DROP,axis=1))

        Trr.append(np.sqrt(mean_squared_error(Train_Y, pred1)))
        Tss.append(np.sqrt(mean_squared_error(Test_Y, pred2)))
print('Dropped Features --> ',DROP)
plt.plot(Trr, label='Train RMSE')
plt.plot(Tss, label='Test RMSE')
plt.legend()
plt.grid()
plt.show()

from sklearn.preprocessing import PolynomialFeatures
Trr=[]; Tss=[]; n=3
order=['ord-'+str(i) for i in range(2,n)]
Trd = pd.DataFrame(np.zeros((10,n-2)), columns=order)
Tsd = pd.DataFrame(np.zeros((10,n-2)), columns=order)

m=df.shape[1]-1
for i in range(m):
    lm = LinearRegression()
    rfe = RFE(lm,n_features_to_select=Train_X_std.shape[1]-i)           # running RFE
```

```
    rfe = rfe.fit(Train_X_std, Train_Y)

    LR = LinearRegression()
    LR.fit(Train_X_std.loc[:,rfe.support_], Train_Y)
    pred1 = LR.predict(Train_X_std.loc[:,rfe.support_])
    pred2 = LR.predict(Test_X_std.loc[:,rfe.support_])

    Trr.append(np.sqrt(mean_squared_error(Train_Y, pred1)))
    Tss.append(np.sqrt(mean_squared_error(Test_Y, pred2)))
plt.plot(Trr, label='Train RMSE')
plt.plot(Tss, label='Test RMSE')
#plt.ylim([19.75,20.75])
plt.legend()
plt.grid()
plt.show()

from sklearn.decomposition import PCA
from sklearn.preprocessing import PolynomialFeatures
Trr=[]; Tss=[]; n=3
order=['ord-'+str(i) for i in range(2,n)]
Trd = pd.DataFrame(np.zeros((10,n-2)), columns=order)
Tsd = pd.DataFrame(np.zeros((10,n-2)), columns=order)
m=df.shape[1]-1

for i in range(m):
    pca = PCA(n_components=Train_X_std.shape[1]-i)
    Train_X_std_pca = pca.fit_transform(Train_X_std)
    Test_X_std_pca = pca.fit_transform(Test_X_std)

    LR = LinearRegression()
    LR.fit(Train_X_std_pca, Train_Y)

    pred1 = LR.predict(Train_X_std_pca)
    pred2 = LR.predict(Test_X_std_pca)
```

```
    Trr.append(round(np.sqrt(mean_squared_error(Train_Y, pred1)),2))
    Tss.append(round(np.sqrt(mean_squared_error(Test_Y, pred2)),2))
plt.plot(Trr, label='Train RMSE')
plt.plot(Tss, label='Test RMSE')
plt.legend()
plt.grid()
plt.show()


Model_Evaluation_Comparison_Matrix = pd.DataFrame(np.zeros([5,8]), columns=['Train-
R2','Test-R2','Train-RSS','Test-RSS',
                                                    'Train-MSE','Test-MSE','Train-RMSE','Test-
RMSE'])
rc=np.random.choice(Train_X_std.loc[:,Train_X_std.nunique()>=30].columns.values,1,replac
e=False)
def Evaluate(n, pred1,pred2):
    #Plotting predicted predicteds alongside the actual datapoints
    plt.figure(figsize=[15,6])
    for e,i in enumerate(rc):
        plt.subplot(2,3,e+1)
        plt.scatter(y=Train_Y, x=Train_X_std[i], label='Actual')
        plt.scatter(y=pred1, x=Train_X_std[i], label='Prediction')
        plt.legend()
    plt.show()

    #Evaluating the Multiple Linear Regression Model

    print('\n\n{}Training Set Metrics{}'.format('-'*20, '-'*20))
    print('\nR2-Score on Training set --->',round(r2_score(Train_Y, pred1),20))
    print('Residual Sum of Squares (RSS) on Training set --
->',round(np.sum(np.square(Train_Y-pred1)),20))
    print('Mean Squared Error (MSE) on Training set       --
->',round(mean_squared_error(Train_Y, pred1),20))
```

```
    print('Root Mean Squared Error (RMSE) on Training set --
->',round(np.sqrt(mean_squared_error(Train_Y, pred1)),20))

    print('\n{}Testing Set Metrics{}'.format('-'*20, '-'*20))
    print('\nR2-Score on Testing set --->',round(r2_score(Test_Y, pred2),20))
    print('Residual Sum of Squares (RSS) on Training set --
->',round(np.sum(np.square(Test_Y-pred2)),20))
    print('Mean Squared Error (MSE) on Training set       --
->',round(mean_squared_error(Test_Y, pred2),20))
    print('Root Mean Squared Error (RMSE) on Training set --
->',round(np.sqrt(mean_squared_error(Test_Y, pred2)),20))
    print('\n{}Residual Plots{}'.format('-'*20, '-'*20))

    Model_Evaluation_Comparison_Matrix.loc[n,'Train-R2'] = round(r2_score(Train_Y,
pred1),20)
    Model_Evaluation_Comparison_Matrix.loc[n,'Test-R2'] = round(r2_score(Test_Y,
pred2),20)
    Model_Evaluation_Comparison_Matrix.loc[n,'Train-RSS'] =
round(np.sum(np.square(Train_Y-pred1)),20)
    Model_Evaluation_Comparison_Matrix.loc[n,'Test-RSS'] =
round(np.sum(np.square(Test_Y-pred2)),20)
    Model_Evaluation_Comparison_Matrix.loc[n,'Train-MSE'] =
round(mean_squared_error(Train_Y, pred1),20)
    Model_Evaluation_Comparison_Matrix.loc[n,'Test-MSE'] =
round(mean_squared_error(Test_Y, pred2),20)
    Model_Evaluation_Comparison_Matrix.loc[n,'Train-RMSE']=
round(np.sqrt(mean_squared_error(Train_Y, pred1)),20)
    Model_Evaluation_Comparison_Matrix.loc[n,'Test-RMSE'] =
round(np.sqrt(mean_squared_error(Test_Y, pred2)),20)

    # Plotting y_test and y_pred to understand the spread.
    plt.figure(figsize=[15,4])

    plt.subplot(1,2,1)
```

```
    sns.distplot((Train_Y - pred1))
    plt.title('Error Terms')
    plt.xlabel('Errors')

    plt.subplot(1,2,2)
    plt.scatter(Train_Y,pred1)
    plt.plot([Train_Y.min(),Train_Y.max()],[Train_Y.min(),Train_Y.max()], 'r--')
    plt.title('Test vs Prediction')
    plt.xlabel('y_test')
    plt.ylabel('y_pred')
    plt.show()

MLR = LinearRegression().fit(Train_X_std,Train_Y)
pred1 = MLR.predict(Train_X_std)
pred2 = MLR.predict(Test_X_std)
print('{}{}\033[1m Evaluating Multiple Linear Regression Model
\033[0m{}{}\n'.format('<'*3,'-'*35 ,'-'*35,'>'*3))
print('The Coeffecient of the Regresion Model was found to be ',MLR.coef_)
print('The Intercept of the Regresion Model was found to be ',MLR.intercept_)

Evaluate(0, pred1, pred2)

RLR = Ridge().fit(Train_X_std,Train_Y)
pred1 = RLR.predict(Train_X_std)
pred2 = RLR.predict(Test_X_std)

print('{}{}\033[1m Evaluating Ridge Regression Model \033[0m{}{}\n'.format('<'*3,'-'*35 ,'-
'*35,'>'*3))
print('The Coeffecient of the Regresion Model was found to be ',MLR.coef_)
print('The Intercept of the Regresion Model was found to be ',MLR.intercept_)

Evaluate(1, pred1, pred2)

LLR = Lasso().fit(Train_X_std,Train_Y)
```

```
pred1 = LLR.predict(Train_X_std)
pred2 = LLR.predict(Test_X_std)

print('{}{}\033[1m Evaluating Lasso Regression Model \033[0m{}{}\n'.format('<'*3,'-'*35 ,'-'*35,'>'*3))
print('The Coeffecient of the Regresion Model was found to be ',MLR.coef_)
print('The Intercept of the Regresion Model was found to be ',MLR.intercept_)

Evaluate(2, pred1, pred2)

ENR = ElasticNet().fit(Train_X_std,Train_Y)
pred1 = ENR.predict(Train_X_std)
pred2 = ENR.predict(Test_X_std)

print('{}{}\033[1m Evaluating Elastic-Net Regression Model \033[0m{}{}\n'.format('<'*3,'-'*35 ,'-'*35,'>'*3))
print('The Coeffecient of the Regresion Model was found to be ',MLR.coef_)
print('The Intercept of the Regresion Model was found to be ',MLR.intercept_)
Evaluate(3, pred1, pred2)

Trr=[]; Tss=[]
n_degree=7

for i in range(2,n_degree):
    #print(f'{i} Degree')
    poly_reg = PolynomialFeatures(degree=i)
    X_poly = poly_reg.fit_transform(Train_X_std)
    X_poly1 = poly_reg.fit_transform(Test_X_std)
    LR = LinearRegression()
    LR.fit(X_poly, Train_Y)

    pred1 = LR.predict(X_poly)
    Trr.append(np.sqrt(mean_squared_error(Train_Y, pred1)))
```

```
    pred2 = LR.predict(X_poly1)
    Tss.append(np.sqrt(mean_squared_error(Test_Y, pred2)))

plt.figure(figsize=[15,6])
plt.subplot(1,2,1)
plt.plot(range(2,n_degree),Trr, label='Training')
plt.plot(range(2,n_degree),Tss, label='Testing')
#plt.plot([1,4],[1,4],'b--')
plt.title('Polynomial Regression Fit')
#plt.ylim([0,5])
plt.xlabel('Degree')
plt.ylabel('RMSE')
plt.grid()
plt.legend()
#plt.xticks()

plt.subplot(1,2,2)
plt.plot(range(2,n_degree),Trr, label='Training')
plt.plot(range(2,n_degree),Tss, label='Testing')
plt.title('Polynomial Regression Fit')
plt.ylim([25,40])
plt.xlabel('Degree')
plt.ylabel('RMSE')
plt.grid()
plt.legend()
#plt.xticks()
plt.show()

poly_reg = PolynomialFeatures(degree=2)
X_poly = poly_reg.fit_transform(Train_X_std)
X_poly1 = poly_reg.fit_transform(Test_X_std)
PR = LinearRegression()
PR.fit(X_poly, Train_Y)
```

```
pred1 = PR.predict(X_poly)
pred2 = PR.predict(X_poly1)

print('{}{}\033[1m Evaluating Polynomial Regression Model \033[0m{}{}\n'.format('<'*3,'-'*35 ,'-'*35,'>'*3))
print('The Coeffecient of the Regresion Model was found to be ',MLR.coef_)
print('The Intercept of the Regresion Model was found to be ',MLR.intercept_)

Evaluate(4, pred1, pred2)

from sklearn.linear_model import BayesianRidge

# Create an instance of BayesianRidge

poly_reg = PolynomialFeatures(degree=2)
X_poly = poly_reg.fit_transform(Train_X_std)
X_poly1 = poly_reg.fit_transform(Test_X_std)
BR = BayesianRidge()
BR.fit(X_poly, Train_Y)

pred1 = BR.predict(X_poly)
pred2 = BR.predict(X_poly1)
# Fit the model with training data

# Make predictions on training and testing data



# Evaluate the model's performance
Evaluate('Bayesian Ridge Regression', pred1, pred2)

# Function to evaluate model and update evaluation comparison matrix
def Evaluate(model_name, pred_train, pred_test):
    # Plotting predicted values against actual values
    plt.figure(figsize=[15, 6])
```

```
    for e, i in enumerate(rc):
        plt.subplot(2, 3, e + 1)
        plt.scatter(y=Train_Y, x=Train_X_std[i], label='Actual')
        plt.scatter(y=pred_train, x=Train_X_std[i], label='Prediction')
        plt.legend()
    plt.show()

    # Evaluating the model
    print('\n\n{}Training Set Metrics{}'.format('-' * 20, '-' * 20))
    print('\nR2-Score on Training set --->', round(r2_score(Train_Y, pred_train), 20))
    print('Mean Squared Error (MSE) on Training set --->',
round(mean_squared_error(Train_Y, pred_train), 20))
    print('Root Mean Squared Error (RMSE) on Training set --->',
round(np.sqrt(mean_squared_error(Train_Y, pred_train)), 20))

    print('\n{}Testing Set Metrics{}'.format('-' * 20, '-' * 20))
    print('\nR2-Score on Testing set --->', round(r2_score(Test_Y, pred_test), 20))
    print('Mean Squared Error (MSE) on Testing set --->', round(mean_squared_error(Test_Y,
pred_test), 20))
    print('Root Mean Squared Error (RMSE) on Testing set --->',
round(np.sqrt(mean_squared_error(Test_Y, pred_test)), 20))
    print('\n{}Residual Plots{}'.format('-' * 20, '-' * 20))

    Model_Evaluation_Comparison_Matrix.loc[model_name, 'Train-R2'] =
round(r2_score(Train_Y, pred_train), 20)
    Model_Evaluation_Comparison_Matrix.loc[model_name, 'Test-R2'] =
round(r2_score(Test_Y, pred_test), 20)
    Model_Evaluation_Comparison_Matrix.loc[model_name, 'Train-MSE'] =
round(mean_squared_error(Train_Y, pred_train), 20)
    Model_Evaluation_Comparison_Matrix.loc[model_name, 'Test-MSE'] =
round(mean_squared_error(Test_Y, pred_test), 20)
    Model_Evaluation_Comparison_Matrix.loc[model_name, 'Train-RMSE'] =
round(np.sqrt(mean_squared_error(Train_Y, pred_train)), 20)
```

```
    Model_Evaluation_Comparison_Matrix.loc[model_name, 'Test-RMSE'] =
round(np.sqrt(mean_squared_error(Test_Y, pred_test)), 20)

    # Plotting y_test and y_pred to understand the spread.
    plt.figure(figsize=[15, 4])

    plt.subplot(1, 2, 1)
    sns.distplot((Train_Y - pred_train))
    plt.title('Error Terms')
    plt.xlabel('Errors')

    plt.subplot(1, 2, 2)
    plt.scatter(Train_Y, pred_train)
    plt.plot([Train_Y.min(), Train_Y.max()], [Train_Y.min(), Train_Y.max()], 'r--')
    plt.title('Test vs Prediction')
    plt.xlabel('y_test')
    plt.ylabel('y_pred')
    plt.show()


EMC = Model_Evaluation_Comparison_Matrix.copy()

# Update the index with meaningful labels for each regression model, including Bayesian
Ridge Regression
EMC.index = [
    'Multiple Linear Regression (MLR)',
    'Ridge Linear Regression (RLR)',
    'Lasso Linear Regression (LLR)',
    'Elastic-Net Regression (ENR)',
    'Polynomial Regression (PNR)',
    'Bayesian Ridge Regression (BR)'
]
EMC
```

```
cc = Model_Evaluation_Comparison_Matrix.columns.values
plt.bar(np.arange(6), Model_Evaluation_Comparison_Matrix[cc[6]].values, width=0.25,
label='RMSE (Training)')
plt.bar(np.arange(6) + 0.25, Model_Evaluation_Comparison_Matrix[cc[7]].values,
width=0.25, label='RMSE (Testing)')
plt.bar(np.arange(6) + 0.5, Model_Evaluation_Comparison_Matrix[cc[7]].values, width=0.25,
label='Bayesian Ridge')
plt.xticks(np.arange(6), EMC.index, rotation=35)
plt.legend()
plt.ylim([25, 35])
plt.show()


R2 = round(EMC['Train-R2'].sort_values(ascending=True),4)
plt.hlines(y=R2.index, xmin=0, xmax=R2.values)
plt.plot(R2.values, R2.index,'o')
plt.title('R2-Scores Comparison for various Regression Models')
plt.xlabel('R2-Score')
#plt.ylabel('Regression Models')
for i, v in enumerate(R2):
    plt.text(v+0.02, i-0.05, str(v*100), color='blue')
plt.xlim([0,1.1])
plt.show()

R2 = round(EMC['Test-R2'].sort_values(ascending=True),4)
plt.hlines(y=R2.index, xmin=0, xmax=R2.values)
plt.plot(R2.values, R2.index,'o')
plt.title('R2-Scores Comparison for various Regression Models')
plt.xlabel('R2-Score')
#plt.ylabel('Regression Models')
for i, v in enumerate(R2):
    plt.text(v+0.02, i-0.05, str(v*100), color='blue')
plt.xlim([0,1.1])
plt.show()
```

6. RESULT AND CONCLUSION

6.1 RESULTS AND CONCLUSION:

After evaluating the performance of Polynomial Regression and Bayesian Ridge Regression on the testing set, the following results were obtained:

Polynomial Regression:

- R2-Score: 0.1678

- Root Mean Squared Error (RMSE): 29.41

Bayesian Ridge Regression:

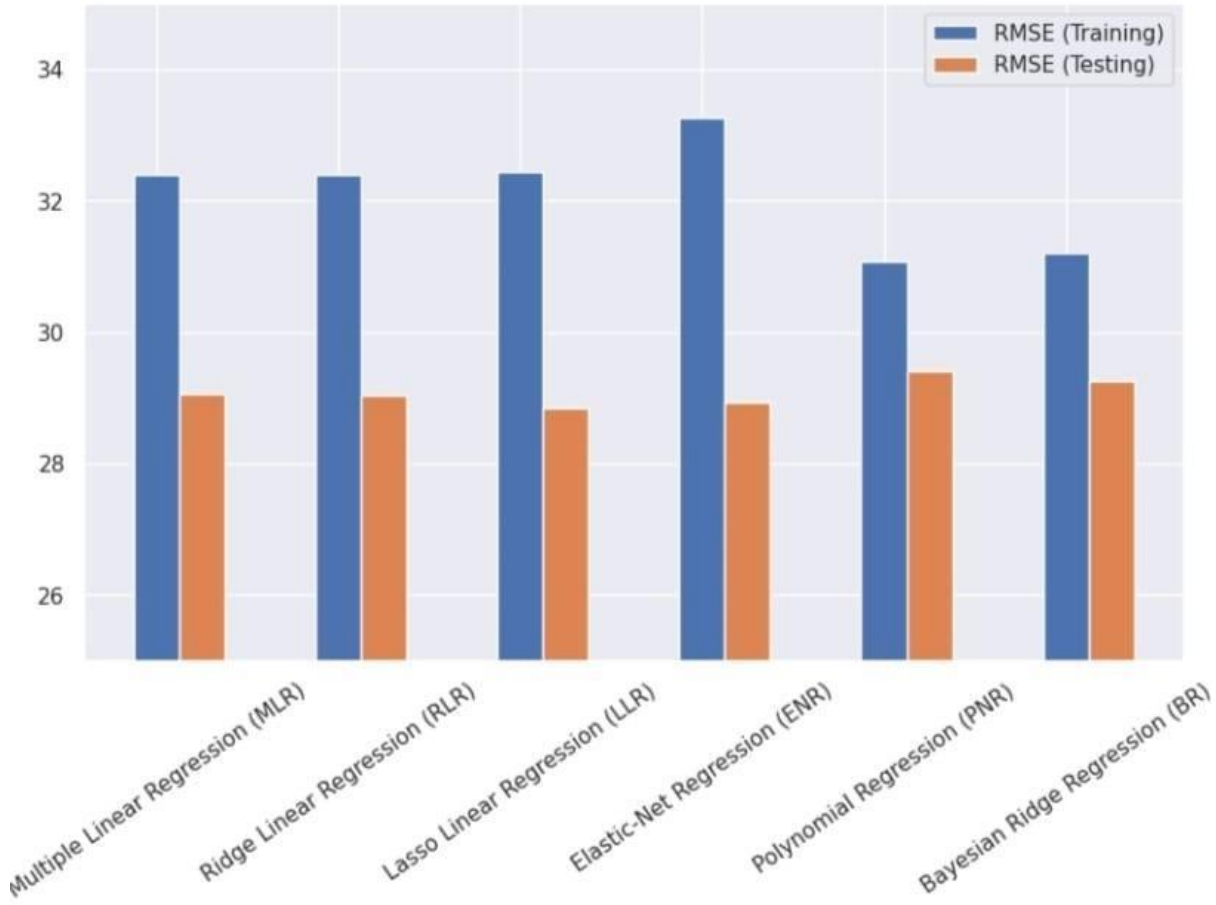
- R2-Score: 0.1773

- Root Mean Squared Error (RMSE): 29.25

Comparing the two models, Bayesian Ridge Regression demonstrates slightly better performance in terms of the R2-Score (0.1773) and a slightly lower RMSE (29.25) compared to Polynomial Regression with an R2-Score of 0.1678 and an RMSE of 29.41. However, the differences in performance between the models are relatively small.

Therefore, based solely on the provided evaluation metrics for the testing set, the conclusion is that Bayesian Ridge Regression marginally outperforms Polynomial Regression for this specific dataset.

Lesser the RMSE, better the model! Also, provided the model should have close proximity with the training & testing scores. For this problem, it is can be said that Bayesian ridge regression clearly is the best fit for the given dataset.


RMSE (Training)
RMSE (Testing)
34
32
30
28
26
Multiple Linear Regression (MLR)
Ridge Linear Regression (RLR)
Lasso Linear Regression (LLR)
Elastic-Net Regression (ENR)
Polynomial Regression (PNR)
Bayesian Ridge Regression (BR)

7. FUTURE SCOPE

The future scope of predicting employees' salaries is very promising as advancements in technology and data analysis continue to improve. With the development of machine learning algorithms and predictive analytics, it is becoming increasingly easier to accurately forecast salaries based on a variety of factors such as job title, location, experience, and industry trends.

Companies are increasingly turning to data-driven approaches to make informed decisions about compensation, and employees are also demanding more transparency and fairness in salary structures. By leveraging data and analytics, companies can ensure that their employees are fairly compensated and that they remain competitive in the market.

In the future, we can expect to see even more sophisticated tools and models for predicting employees' salaries, as well as a greater emphasis on using data to drive decision-making around compensation. This will not only benefit companies in terms of attracting and retaining top talent, but also help create a more equitable and inclusive workplace for all employees.

8.REFERENCES

1. Galton, F.(1886).regression towards mediocrity in hereditary stature the journal of the anthropological institute of great Britain and ireland,15,246-263.

2. https://jespublication.com/uploads/2023-V14I9033.pdf employee salaries prediction

3. Jiang,y.,He,y.,& Zhang H.(2016).Variable selection with prior information for generalized linear models via the prior Lasso method .journal of the American statistical association,111(513),355-376.

4. https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3526707 salary prediction using machine learning.

5. Wright ,s.j.(2015).coordinate descent algorithms ,Mathematical programming,151(1),3- 34

6. M.Yasser, H.(2021).Employee salaries dataset(version 1).

7. Ostertagova ,E.(2012).Modelling using polynomial regression. procedia Engineering ,48,500-506.

8. Agarwal, N., Gokhale, C. S., & Kulkarni, V. (2020). Predictive Analytics for Employee Attrition and Salary Prediction Using Machine Learning Algorithms. In Proceedings of the International Conference on Data Science and Applications.

9. Chen, C., & Wu, S. (2019). A Machine Learning-Based Approach for Salary Prediction. In Proceedings of the International Conference on Machine Learning and Cybernetics.
10. Gaur, M., Mittal, A., & Singh, P. (2019). Employee Salary Prediction and Attrition Rate Analysis Using Machine Learning. In Proceedings of the International Conference on Intelligent Sustainable Systems.
11. Li, X., Hu, Y., & Zhang, Y. (2018). A Salary Prediction Model for IT Professionals Based on Machine Learning. In Proceedings of the International Conference on Advanced Computational Intelligence Systems.
12. Suresh, S., & Sabitha, R. (2017). Predictive Analytics on Employee Salary with Machine Learning Techniques. International Journal of Computer Applications, 161(11), 36-39.

EMPLOYEE SALARY ANALYSIS AND PREDICTION USING MACHINE LEARNING

M.Suneetha
Department of Computer Science and Engineering
Narasaraopeta Engineering College
Narasaraopeta
msuneetha973@gmail.com

K. Narasimha Charyulu
Department of Computer Science and Engineering
Narasaraopeta Engineering College
Narasaraopeta
kanjarlanarashimhacharyulu@gmail.com

A. Madhava Rao
Department of Computer Science and Engineering
Narasaraopeta Engineering College
Narasaraopeta

U. Sai Kumar
Department of Computer Science and Engineering
Narasaraopeta Engineering College
Narasaraopeta

Abstract

Accurately estimating employee pay is essential for both companies and jobseekers in today's competitive employmentmarket. This paper suggests a techniquethat makes use of machine learning algorithms to forecast employee pay. The dataset includes a number of characteristics, including industry, region, employment function, years of experience, and education level. Different machine learning models, such as gradient boosting methods, decision trees, random forests, and linear regression, are trained andevaluated using these features. Several performance indicators are employed to evaluate the models' accuracy and efficacy,The outcomes show how well machine learning approaches work in predicting employee pay with a high level of precision. Employers can use these modelsto make well-informed judgments about compensation offers, and job seekers can use them to learn what kind of salary range they should expect given their experience and qualifications.

1. INTRODUCTION

Precisely forecasting employee pay is a basic duty in the field of human resources management that has major consequences for businesses and employees alike.

Conventional techniques for determining salaries frequently rely on opinionated evaluations or accepted practices in the field, which could introduce bias or incorrect information. On the other hand data-driven approaches to salary prediction are now available in machine learning which make decision making processes more accurate and transparent. For this endeavour, regression algorithms are very useful since they let us model the link between a variety of characteristics, including industry, location, years of experience, job function, and education level, and the related salary. Regression algorithms are highly predictive because of their capacity to identify intricate links in the data and reveal patterns and trends that would not be visible using more conventional techniques. We aim to determine the best method for wage prediction in a real-world setting by evaluating the output of several regression models, such as Linear regression, polynomial regression, lasso regression, multivariate linear regression. Using univariate regression analysis, the relationships between a dependent and independent variable are examined. Since multiple algorithms will perform differently for a given data set, our approaches compare each regression model's performance to determine which

algorithm performs best for pay predictions.

2. LITERATURE SURVEY

A quick overview of the different machine learning algorithms that are mostcommonly used to solve problems with classification, regression, and clustering is provided by **Susmita Ray[8]** in her paper . **Sananda Dutta, Airiddha Halder, and Kousik Dasgupta[6]** "Creation of an innovative Prediction Engine to Estimate Appropriate Compensation for an Employment". The dataset supplied by ADZUNA serves as the foundation for our investigation.

Pornthep Khongchai and Pokpong Songmuang's study[3], "Improving Students' Motivation to Study using Salary Prediction System," suggested a method yield a predicted wage.

In their study, "Salary Predictor System for Thailand Labour Workforce using Deep Learning," **Phuwadol Viroonluecha and Thongchai Kaewkiriya[9]** employed deep learning techniques to build a model that accurately predicts the monthly salary of job seekers in Thailand by solving a regression problem with a numerical outcome.

Associative rule mining was used by **K. Lakshmi and A. Parkavil[1]** to analyze data in order to determine the extent of students' knowledge. They used linear regression and association rule mining to determine which group (or cluster) the student knowledge is most closely related to Data mining .

According to research by **Rajveer Singh[7]**, academic achievement in school and college, affiliation with the institution, and college reputation are significant predictors of starting income for entry-level Indian engineering graduates. In their work, **C.-C. Hung and E.-P. Lim[10]** suggested the "Company, Occupation, Company" (COC) model as a means of obtaining objective salaries through the combination of jobreview and job post data. This algorithm is able to accurately forecast organizations' inflation, competitiveness, and unbiased compensation.

3. PROPOSED MODEL

3.1 Factors considered in Proposed work

Age
PHD
Gender
Salary

3.2 Dataset Description

Name: Employee Salaries
Source: Kaggle
Attributes: 4
Instances:100
Type: Supervised
Training Data:80%
Testing Data:20%

3.3 Data Preproccesing

Preprocessing is the first step in creating a machine learning model. The process entails transforming an unclean dataset into a cleaned one. Because raw data includes noise, missing values, and duplicate values, it is not appropriate for machine learning models. For this reason, preprocessing is required.

3.4 Splitting Data into Training and Test Data

When estimating the effectiveness of machine learning algorithms that can be used to prediction-based algorithms and applications, the train-test split is utilized.This is a quick and simple process that allows us to compare the output of our machine learning model with that of other machines. By default, 30% of the real data is divided into the Test set and 70% of the real data is divided into the Training set. To assess how well our machine learning model works, we must divide a dataset into train and test sets. The most reliable and practical Python machine learning library is called scikit-learn, sometimes known as sklearn. The splitter function train_test_split() is available in the model_selection module of the scikit-learn toolkit.

TrainingData:80%
Testing Data:20%

3.5 Feature Extraction

Dimensionality reduction techniques aimto simplify complex datasets by reducing the number of variables or features while preserving essential information. One common method within dimensionality reduction is feature extraction. By extracting relevant features from raw data, the process becomes more manageable and computationally efficient, particularly when dealing with large datasets characterized by numerous variables The abundance of variables in large datasets poses a significant computational challenge, requiring substantial computer power for processing. Feature extraction addresses this challenge by selecting and combining variables into meaningful features, thereby reducing the data's dimensionality without losing crucial information. This streamlined approach enhances the efficiency of processing large datasets by focusing on extracting the most relevant features for analysis.

3.6 Alogirthms Used in our Model

3.6.1 Linear Regression

Linear regression is a fundamental and widely utilized machine learning technique for predictive analysis. It serves as a statistical method for modeling the relationship between one or more independent variables .

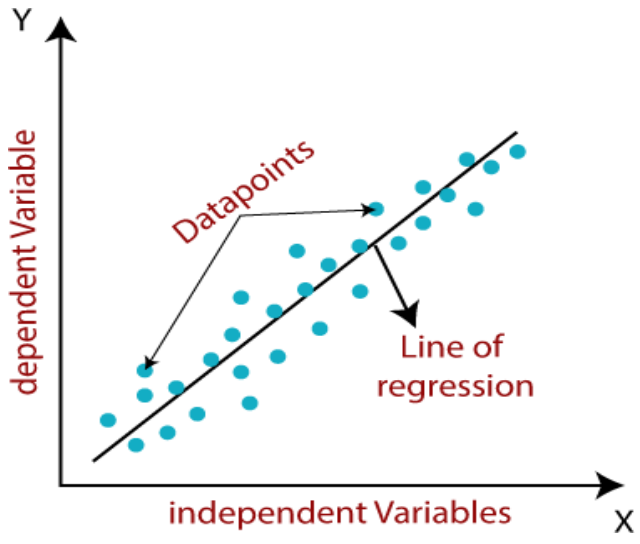


Fig-3.6.1.1 Linear Regression

3.6.2 Multi Linear Regression

Multiple linear regression indeed extends the concept of simple linear regression by accommodating multiple independent variables to predict a continuous dependent variable. It's a valuable regression technique for modeling complex relationships between the dependent variable and multiple predictors.

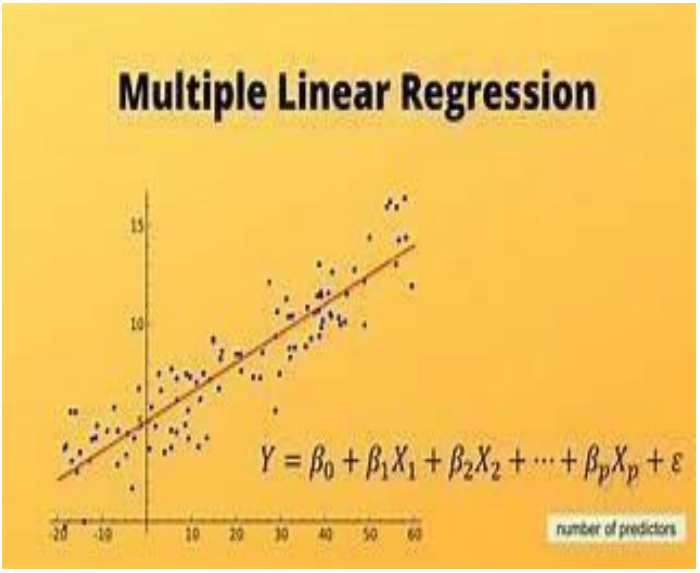


Fig-3.6.2.1 Multi Linear Regression

3.6.4Polynomial Regression

In polynomial regression, a polynomial of degree is used to model the relationship between an independent variable(x) and a dependent variable (y) y). Unlike simple linear regression, which assumes a linear relationship between the variables, polynomial regression allows for more flexible modeling of nonlinear relationships.
y = b0 + b1 * 1 + b2 * 12 + b2 * 13 +.....
bnx1n

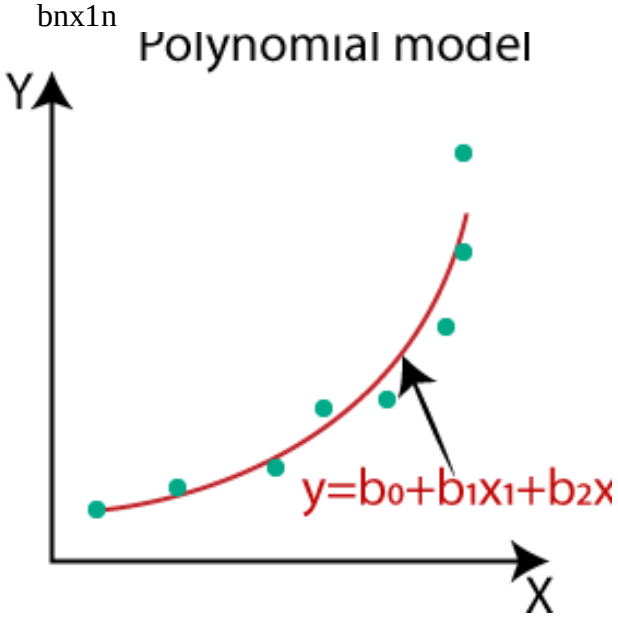


Fig-3.6.4.1 Polynomial Regression

3.6.3 Bayesian Ridge Regression

A kind of linear regression called Bayesian regression makes use of Bayesian statistics toestimate a model's unknown parameters. To determine the likelihood of a collection of parameters given observed data, it applies the Bayes theorem. While Bayesian regression makes deeper assumptions aboutthe structure of the data and assigns a prior probability distribution to the parameters, traditional linear regression makes the assumption that the data follows a Gaussian or normal distribution.

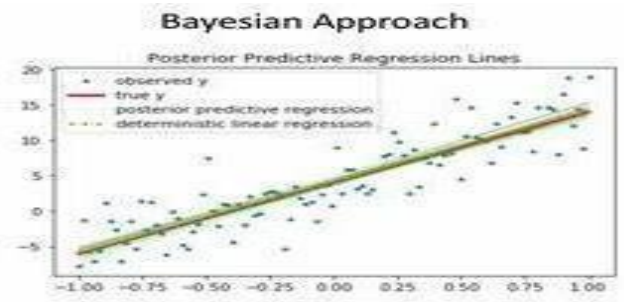


Fig-3.6.5.1 Baysian Ridge Regression

3.7 System Architecture

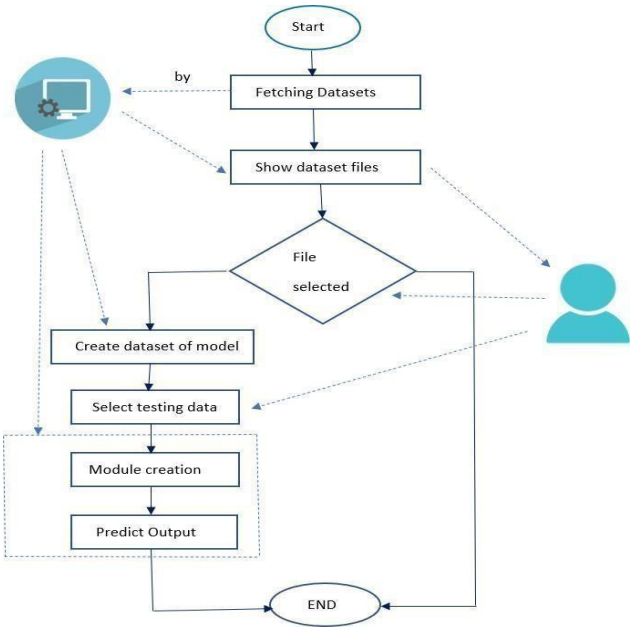


Fig-3.7.1 System Architecture

4. RESULTS

4.1 Linear Regression

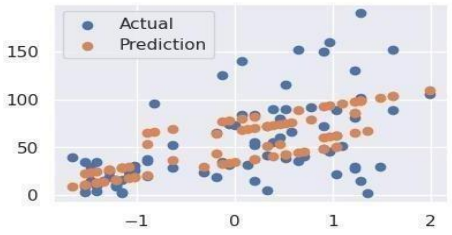


Fig 4.1.1 Actual Prediction plot in linear Regression

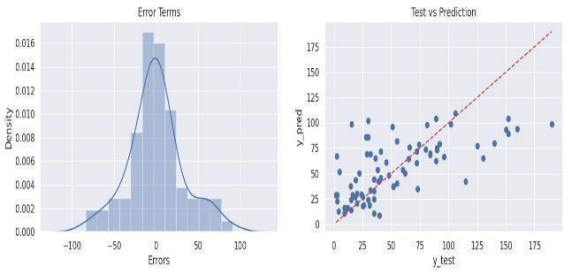


Fig 4.1.2 Test Vs Prediction plot in Linear

4.2 Muli Linear Regression

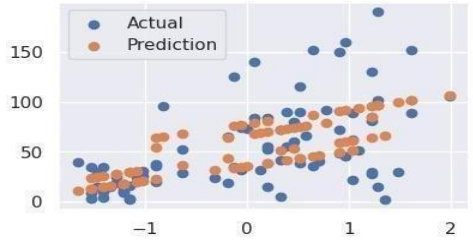


Fig 4.2.1 Actual Prediction plot in Multi Linear Regression

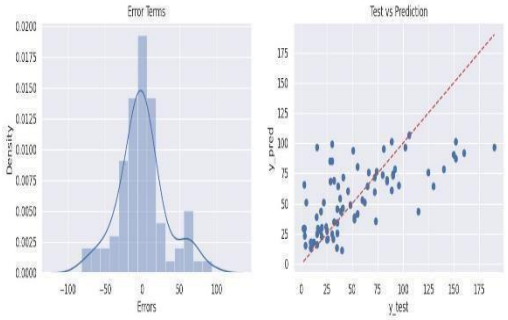


Fig 4.2.2 Test Vs Prediction plot in Multi linearRegression

4.3.Polynomial Regression

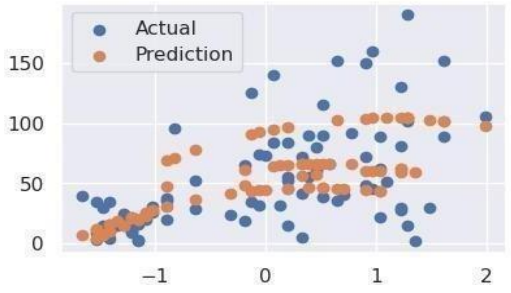


Fig 4.3.1 Actual Prediction plot inPolynomial Regression

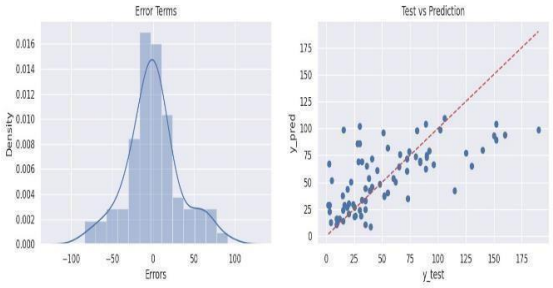


Fig 4.3.2 Test Vs Prediction plot inPolynomial Regression

4.4 Bayesian Ridge Regression

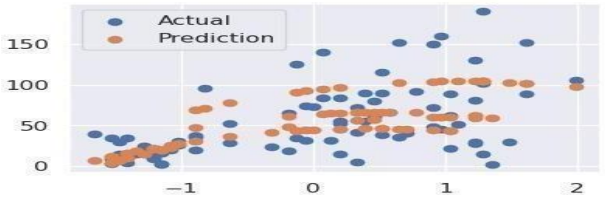


Fig 4.4.1 Actual Prediction plot in Bayesian Ridge Regression

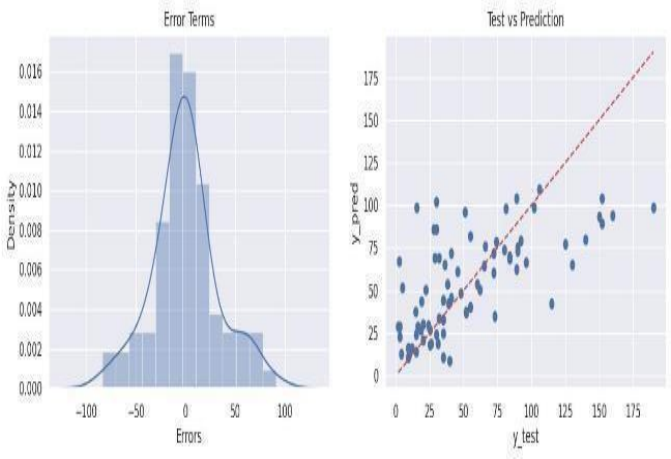


Fig 4.4.2 Test Vs Prediction plot inBayesian Ridge Regression

4.5 Correlation Matrix

Fig 4.5.1 Correlation Matrix

4.6 Metrics of our Model

4.6.1 For Training

Algorithm	RMSE Score	MSE Score
Multi Linear Regression	32.39518598926559	1049.448075279104
Polynomial Regression	31.061863962110078	964.8393928006328
Ridge Regression	32.396025086822239	1049.5024414260256
Bayesian Ridge Regression	31.201159454277683	973.5123512911262

3.6.4 For Testing

Algorithm	RMSE Score	MSE Score
Multi Linear Regression	29.065548067327374	844.8060844541118
Polynomial Regression	29.413758605667862	865.1691953125
Ridge Regression	29.048327516633826	843.8053315136254
Bayesian Ridge Regression	29.245864822820437	855.3206092346859

5. CONCLUSION

For the purpose of employee prediction, Bayesian regression clearly outperforms models after a thorough analysis of linear regression, multi-linear regression, and ridge regression models. Bayesian Ridge regression is the algorithm that consistently shows better performance metrics on the test dataset among those that are being investigated. While preserving competitive predictive accuracy, its regularization approach successfully reduces overfitting. Furthermore, because Bayesian ridge regression features are resilient to multicollinearity—a major problem in employee prediction models—it can handle highly correlated features. The best method for our project is Bayesian ridge regression, which offers trustworthy predictions for outcomes related to employees by balancing bias and variance. Because ofthis, we can infer with confidence that Bayesian ridge regression is the best option for employee prediction tasks based on our data, opening up a promising new field for applications in the human resource management

REFERENCES

[1] A. Parkavil and A. K. Lakshmi 2017 IEEE International Conference on Smart Computing, Communication, Controls, Energy, and Materials: Predicting students' course knowledge levels using data mining techniques

[2] N.A. Rashid, A.M. Shahiri, and A.W. Husain A Review of Data Mining Techniques for Predicting Student Performance .

[3] A Survey of Educational Data-Mining Research by Richard A. Huebner, Academic and Business Research Institute, 2013 [4] Using a salary prediction system to increase students' incentive to study, Pornthep Khongchai and Pokpong Songmuang presented at the 13th International Joint Conference on

Computer Science and Software Engineering in 2016.
[4] S. Vijayalakshmi Anupama Kumar M.N. "Inference of Naïve Baye's Technique on Student Assessment Data," Communications in Computer and Information Science book series (volume 270), 2011.

[5] Do college students anticipate their future income more accurately than young adults already employed? John Jerrim, Education Economics, Taylor & Francis Journals, vol. 23(2), pages 162-179, 2015.

[6] In 2018, Kousik Dasgupta, Sananda Dutta, and Airiddha Halder presented at the Fourth International Conference on Research in Computational Intelligence and Communication Networks (ICRCICN) on "Design of a novel Prediction Engine for predicting suitable salary for a job."
[7] Rajveer Singh, Masters Dissertation: A Regression Analysis of Salary Determinants in Indian Employment Markets for Entry-Level Engineering Graduates. Institute of Technology, Dublin,2016.

[8] Susmita Ray, "A Brief Overview of Machine Learning Algorithms," in 2019 International Conference on Machine Learning, Big Data, Cloud, and Parallel Computing (Com-IT-Con), held in India from February 14–16 .
[9] Thongchai Kaewkiriya and Phuwadol Viroonluecha, "Salary Predictor System for Thailand Labour Workforce using Deep Learning" The 18th International Conference on Information and Communication Technologies (ISCIT2018)
[10] Hung, C.C. and E.-P. Lim, "Combining Occupation Salaries from Job Post and Review Data," IEEE Access, Volume 9, 2021

RE-2022-249858 - Turnitin Plagiarism Report

by Gonuguntla Pavani

Submission date: 27-Apr-2024 06:32PM (UTC+0100)
Submission ID: 271714259071
File name: RE-2022-249858.pdf (361.29K)
Word count: 1862
Character count: 10980

EMPLOYEE SALARY ANALYSIS AND PREDICTION USING MACHINE LEARNING

Abstract

Accurately estimating employee pay is essential for both companies and job seekers in today's competitive employment market. This paper suggests a technique that makes use of machine learning algorithms to forecast employee pay. The dataset includes a number of characteristics, including industry, region, employment function, years of experience, and education level. Different machine learning models, such as gradient boosting methods, decision trees, random forests, and linear regression, are trained and evaluated using these features. Several performance indicators are employed to evaluate the models' accuracy and efficacy, The outcomes show how well machine learning approaches work in predicting employee pay with a high level of precision. Employers can use these models to make well-informed judgments about compensation offers, and job seekers can use them to learn what kind of salary range they should expect given their experience and qualifications.

1.INTRODUCTION

Precisely forecasting employee pay is a basic duty in the field of human resources management that has major consequences for businesses and employees alike.

Conventional techniques for determining salaries frequently rely on opinionated evaluations or accepted practices in the field, which could introduce bias or incorrect information. On the other hand data-driven approaches to salary prediction are now available in machine learning which make decision making processes more accurate and transparent. For this endeavour, regression algorithms are very useful since they let us model the link between a variety of characteristics, including industry, location, years of experience, job function, and education level, and the related salary. Regression algorithms are highly predictive because of their capacity to identify intricate links in the data and reveal patterns and trends that would not be visible using more conventional techniques. We aim to determine the best method for wage prediction in a real-world setting by evaluating the output of several regression models, such as Linear regression, polynomial regression, lasso regression, multivariate linear regression. Using univariate regression analysis, the relationships between a dependent and independent variable are examined. Since multiple algorithms will perform differently for a given data set, our approaches compare each regression model's performance to determine which

algorithm performs best for pay predictions.

2.LITERATURE SURVEY

A quick overview of the different machine learning algorithms that are most commonly used to solve problems with classification, regression, and clustering is provided by **Susmita Ray[8]** in her paper .

Sananda Dutta, Airiddha Halder, and Kousik Dasgupta[6] "Creation of an innovative Prediction Engine to Estimate Appropriate Compensation for an Employment". The dataset supplied by ADZUNA serves as the foundation for our investigation.

Pornthep Khongchai and Pokpong Songmuang's study[3], "Improving Students' Motivation to Study using Salary Prediction System," suggested a method yield a predicted wage.

In their study, "Salary Predictor System for Thailand Labour Workforce using Deep Learning," **Phuwadol Viroonluecha and Thongchai Kaewkiriya[9]** employed deep learning techniques to build a model that accurately predicts the monthly salary of job seekers in Thailand by solving a regression problem with a numerical outcome.

Associative rule mining was used by **K. Lakshmi and A. Parkavil[1]** to analyze data in order to determine the extent of students' knowledge. They used linear regression and association rule mining to determine which group (or cluster) the student knowledge is most closely related to Data mining .

According to research by **Rajveer Singh[7]**, academic achievement in school and college, affiliation with the institution, and college reputation are significant predictors of starting income for entry-level Indian engineering graduates. In their work, **C.-C. Hung and E.-P. Lim[10]** suggested the "Company, Occupation, Company" (COC) model as a means of obtaining objective salaries through the combination of job review and job post data. This algorithm is able to accurately forecast organizations' inflation, competitiveness, and unbiased compensation.

3.PROPOSED MODEL

3.1 Factors considered in Proposed work

Age
PHD
Gender
Salary

3.2 Dataset Description

Name: Employee Salaries
Source: Kaggle
Attributes: 4
Instances:100
Type: Supervised
Training Data:80%
Testing Data:20%

3.3 Data Preproccesing

Preprocessing is the first step in creating a machine learning model. The process entails transforming an unclean dataset into a cleaned one. Because raw data includes noise, missing values, and duplicate values, it is not appropriate for machine learning models. For this reason, preprocessing is required.

3.4 Splitting Data into Training and Test Data

When estimating the effectiveness of machine learning algorithms that can be used to prediction-based algorithms and applications, the train-test split is utilized.This is a quick and simple process that allows us to compare the output of our machine learning model with that of other machines. By default, 30% of the real data is divided into the Test set and 70% of the real data is divided into the Training set. To assess how well our machine learning model works, we must divide a dataset into train and test sets. The most reliable and practical Python machine learning library is called scikit-learn, sometimes known as sklearn. The splitter function train_test_split() is available in the model_selection module of the scikit-learn toolkit.

TrainingData:80%
Testing Data:20%

3.5 Feature Extraction

Dimensionality reduction techniques aim to simplify complex datasets by reducing the number of variables or features while preserving essential information. One common method within dimensionality reduction is feature extraction. By extracting relevant features from raw data, the process becomes more manageable and computationally efficient, particularly when dealing with large datasets characterized by numerous variables The abundance of variables in large datasets poses a significant computational challenge, requiring substantial computer power for processing. Feature extraction addresses this challenge by selecting and combining variables into meaningful features, thereby reducing the data's dimensionality without losing crucial information. This streamlined approach enhances the efficiency of processing large datasets by focusing on extracting the most relevant features for analysis.

3.6 Algorithms Used in our Model

3.6.1 Linear Regression

Linear regression is a fundamental and widely utilized machine learning technique for predictive analysis. It serves as a statistical method for modeling the relationship between one or more independent variables .

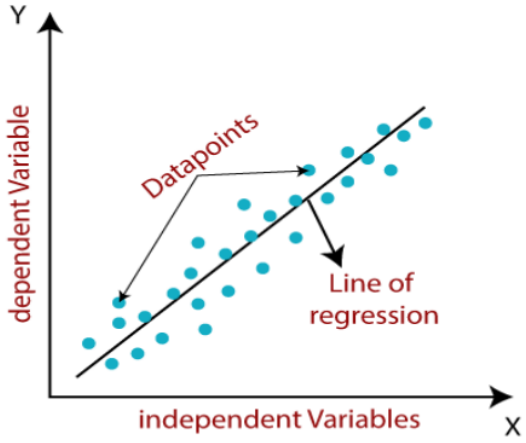


Fig-3.6.1.1 Linear Regression

3.6.2 Multi Linear Regression

Multiple linear regression indeed extends the concept of simple linear regression by accommodating multiple independent variables to predict a continuous dependent variable. It's a valuable regression technique for modeling complex relationships between the dependent variable and multiple predictors.

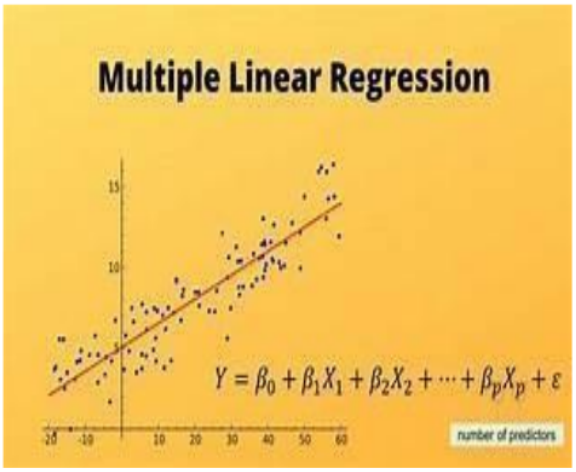


Fig-3.6.2.1 Multi Linear Regression

3.6.4Polynomial Regression

In polynomial regression, a polynomial of degree is used to model the relationship between an independent variable(x) and a dependent variable (y) y). Unlike simple linear regression, which assumes a linear relationship between the variables, polynomial regression allows for more flexible modeling of nonlinear relationships.

y = b0 + b1 * 1 + b2 * 12 + b2 * 13 +.....
bnx1n

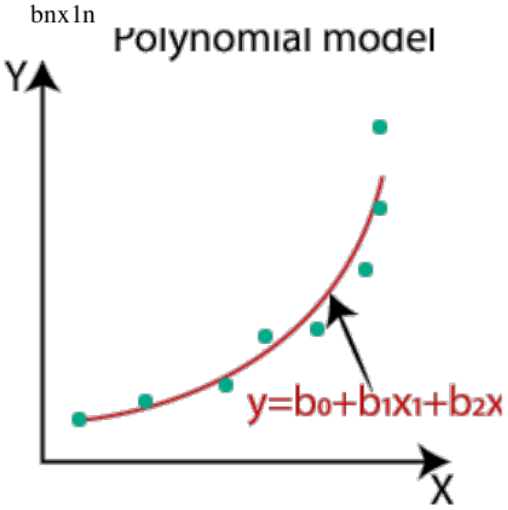


Fig-3.6.4.1 Polynomial Regression

3.6.3 Bayesian Ridge Regression

A kind of linear regression called Bayesian regression makes use of Bayesian statistics toestimate a model's unknown parameters.To determine the likelihood of a collection of parameters given observed data, it applies the Bayes theorem. While Bayesian regression makes deeper assumptions about the structure of the data and assigns a prior probability distribution to the parameters, traditional linear regression makes the assumption that the data follows a Gaussian or normal distribution.

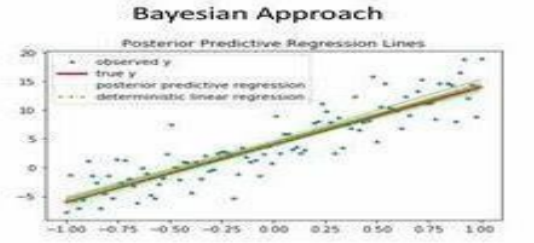


Fig-3.6.5.1 Baysian Ridge Regression

3.7 System Architecture

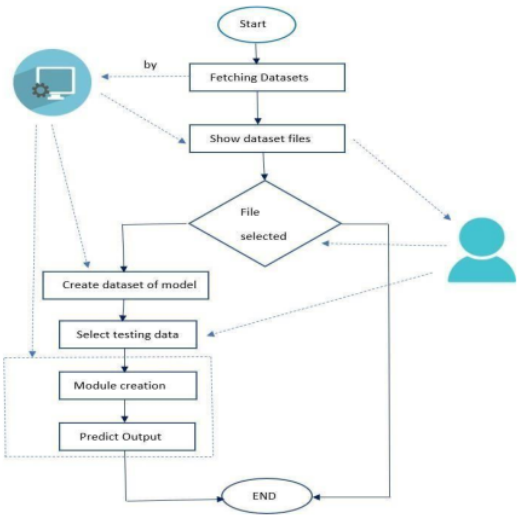


Fig-3.7.1 System Architecture

4.RESULTS

4.1Linear Regression

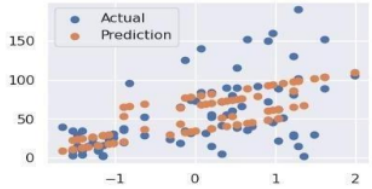


Fig 4.1.1 Actual Prediction plot in linear Regression

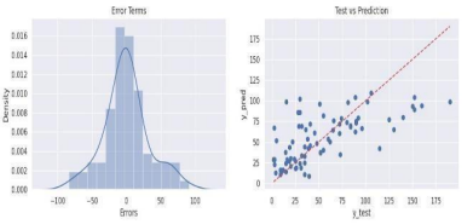


Fig 4.1.2 Test Vs Prediction plot in Linear

4.2 Muli Linear Regression

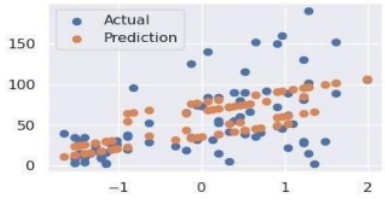


Fig 4.2.1 Actual Prediction plot in Multi Linear Regression

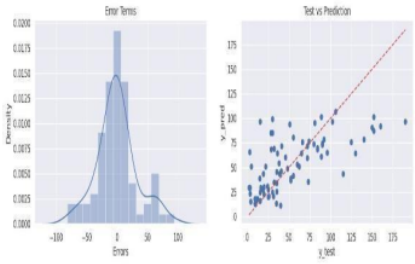


Fig 4.2.2 Test Vs Prediction plot in Multi linearRegression

4.3.Polynomial Regression

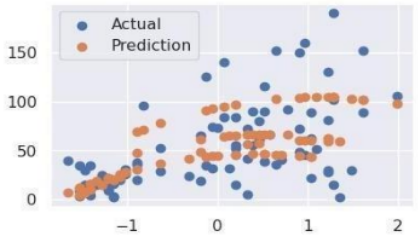


Fig 4.3.1 Actual Prediction plot inPolynomial Regression

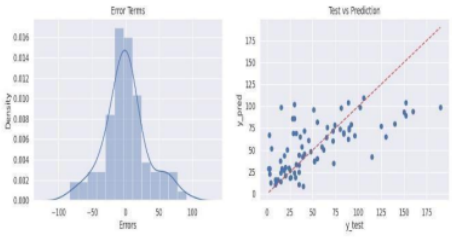


Fig 4.3.2 Test Vs Prediction plot inPolynomial Regression

4.4 Bayesian Ridge Regression

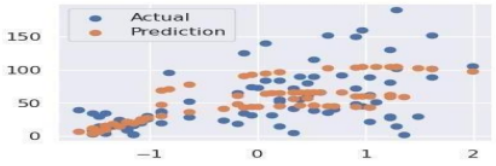


Fig 4.4.1 Actual Prediction plot in Bayesian Ridge Regression

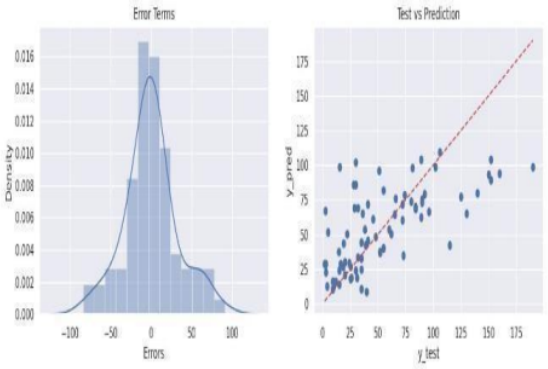


Fig 4.4.2 Test Vs Prediction plot inBayesian Ridge Regression

4.5 Correlation Matrix

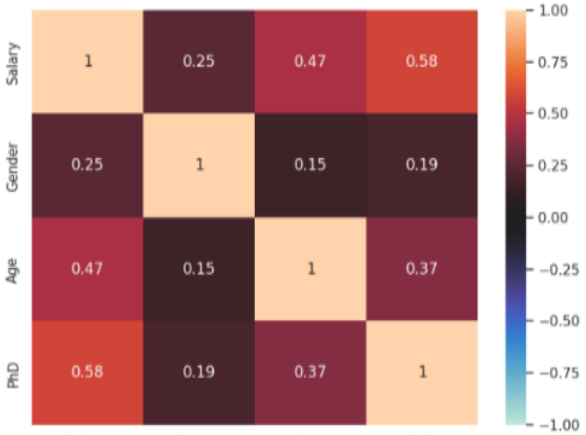


Fig 4.5.1 Correlation Matrix

4.6 Metrics of our Model

4.6.1 For Training

Algorithm	RMSE Score	MSE Score
Multi Linear Regression	32.395185989265	1049.448075279104
Polynomial Regression	31.061863962110078	964.8393928006328
Ridge Regression	32.396025086822239	1049.5024414260256
Bayesian Ridge Regression	31.201159454277683	973.512351291262

3.6.4 For Testing

Algorithm	RMSE Score	MSE Score
Multi Linear Regression	29.065548067327374	844.8060844541118
Polynomial Regression	29.413758605667862	865.1691953125
Ridge Regression	29.048327516633826	843.8053315136254
Bayesian Ridge Regression	29.245864822820437	855.3206092346859

5.CONCLUSION

For the purpose of employee prediction, Bayesian regression clearly outperforms models after a thorough analysis of linear regression, multi-linear regression, and ridge regression models. Bayesian Ridge regression is the algorithm that consistently shows better performance metrics on the test dataset among those that are being investigated. While preserving competitive predictive accuracy, its regularization approach successfully reduces overfitting. Furthermore, because Bayesian ridge regression features are resilient to multicollinearity—a major problem in employee prediction models—it can handle highly correlated features. The best method for our project is Bayesian ridge regression, which offers trustworthy predictions for outcomes related to employees by balancing bias and variance. Because of this, we can infer with confidence that Bayesian ridge regression is the best option for employee prediction tasks based on our data, opening up a promising new field for applications in the human resource management

REFERENCES

[9] A. Parkavil and A. K. Lakshmi 2017 IEEE International Conference on Smart Computing, Communication, Controls, Energy, and Materials: Predicting students' course knowledge levels using data mining techniques

[2] N.A. Rashid, A.M. Shahiri, and A.W. Husain A Review of Data Mining Techniques for Predicting Student Performance .

[3] A Survey of Educational Data-Mining Research by Richard A. Huebner, Academic and Business Research Institute, 2013 [4] Using a salary prediction system to increase students' incentive to study, Pornthep Khongchai and Pokpong Songmuang presented at the 13th International Joint Conference on

Computer Science and Software Engineering in 2016.

[4] S. Vijayalakshmi Anupama Kumar M.N. "Inference of Naïve Baye's Technique on Student Assessment Data," Communications in Computer and Information Science book series (volume 270), 2011.

[5] Do college students anticipate their future income more accurately than young adults already employed? John Jerrim, Education Economics, Taylor & Francis Journals, vol. 23(2), pages 162-179, 2015.

[6] In 2018, Kousik Dasgupta, Sananda Dutta, and Airiddha Halder presented at the Fourth International Conference on Research in Computational Intelligence and Communication Networks (ICRCICN) on "Design of a novel Prediction Engine for predicting suitable salary for a job"

[7] Rajveer Singh, Masters Dissertation: A Regression Analysis of Salary Determinants in Indian Employment Markets for Entry-Level Engineering Graduates. Institute of Technology, Dublin,2016.

[8] Susmita Ray, "A Brief Overview of Machine Learning Algorithms," in 2019 International Conference on Machine Learning, Big Data, Cloud, and Parallel Computing (Com-IT-Con), held in India from February 14–16 .

[9] Thongchai Kaewkiriya and Phuwadol Viroonluecha, "Salary Predictor System for Thailand Labour Workforce using Deep Learning" The 18th International Conference on Information and Communication Technologies (ISCIT 2018)

[10] Hung, C.C. and E.-P. Lim, "Combining Occupation Salaries from Job Post and Review Data," IEEE Access, Volume 9, 2021

RE-2022-249858_(1)[1].pdf

ORIGINALITY REPORT

8%	3%	2%	4%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Universiti Kebangsaan Malaysia Student Paper	2%
2	fastercapital.com Internet Source	1%
3	www.mdpi.com Internet Source	1%
4	Submitted to University of North Texas Student Paper	1%
5	Praveen Mishra, Shivansh Srivastava, Priyanshi Gupta, Atul Anand, Subhash Chandra Gupta. "A Comparative Study of Machine Learning Algorithms for Salary Estimation", 2021 3rd International Conference on Advances in Computing, Communication Control and Networking (ICAC3N), 2021 Publication	1%
6	Submitted to University of Huddersfield Student Paper	1%

7	Anil Kumari Shalini, Sameer Saxena, Billakurthi Suresh Kumar. "Automatic detection of fake news using recurrent neural network—Long short-term memory", Journal of Autonomous Intelligence, 2023 Publication	1 %
8	datafloq.com Internet Source	1 %

Exclude quotes	On	Exclude matches	Off
Exclude bibliography	On		

NARASARAOPETA ENGINEERING COLLEGE
(AUTONOMOUS)

Approved by AICTE, Permanently Affiliated to JNTUK, Kakinada, NIRF Ranking (251-300 Band), Accredited by NBA (Tier-I) & NAAC with 'A+' Grade
Kotappakonda Road, Yellamanda (Post), Narasaraopet - 522601, Palnadu Dist., Andhra Pradesh, INDIA. Website:www.nrtec.in

Paper ID
NECICAIEA-2K24-205

International Conference on
Artificial Intelligence and Its Emerging Areas
NEC-ICAIEA-2K24
12th & 13th April, 2024
Organized by Departments of CSE, IT, CSE(AI), CSE(AI&ML), CSE(DS), CSE(CS) & MCA in Association with CSI

Certificate of Presentation

This is to Certify that **M. Suneetha** , Narasaraopeta engineering college has presented the paper title **Employee salary analysis and prediction using machine learning algorithms** in the International Conference on Artificial Intelligence and Its Emerging Areas-2K24 [NEC-ICAIEA- 2K24], Organized by Department of Computer Science and Engineering, CSE(AI),IT,CSE(AIML),CSE(DS),CSE(CS) and MCA in Association with CSI on 12th and 13th April 2024 at

NARASARAOPETA ENGINEERING COLLEGE (AUTONOMOUS), Narasaraopet, A.P., India.

Convenor
Dr.S.V.N.Srinivasu

Chief-Convenor
Dr.S.N.Tirumala Rao

Principal, Patron
Dr. M. Sreenivasa Kumar

NARASARAOPETA ENGINEERING COLLEGE
(AUTONOMOUS)

Approved by AICTE, Permanently Affiliated to JNTUK, Kakinada, NIRF Ranking (251-300 Band), Accredited by NBA (Tier-I) & NAAC with 'A+' Grade
Kotappakonda Road, Yellamanda (Post), Narasaraopet - 522601, Palnadu Dist., Andhra Pradesh, INDIA. Website:www.nrtec.in

Paper ID
NECICAIEA-2K24-205

International Conference on
Artificial Intelligence and Its Emerging Areas
NEC-ICAIEA-2K24
12th & 13th April, 2024
Organized by Departments of CSE, IT, CSE(AI), CSE(AI&ML), CSE(DS), CSE(CS) & MCA in Association with CSI

Certificate of Presentation

This is to Certify that **Kanjarla Narasimha Charyulu** , Narasaraopeta engineering college has presented the paper title **Employee salary analysis and prediction using machine learning algorithms** in the International Conference on Artificial Intelligence and Its Emerging Areas-2K24 [NEC-ICAIEA-2K24], Organized by Department of Computer Science and Engineering, CSE(AI),IT,CSE(AIML),CSE(DS),CSE(CS) and MCA in Association with CSI on 12th and 13th April 2024 at NARASARAOPETA ENGINEERING COLLEGE (AUTONOMOUS), Narasaraopet, A.P., India.

Convenor
Dr.S.V.N.Srinivasu

Chief-Convenor
Dr.S.N.Tirumala Rao

Principal, Patron
Dr. M. Sreenivasa Kumar

NARASARAOPETA ENGINEERING COLLEGE
(AUTONOMOUS)

Approved by AICTE, Permanently Affiliated to JNTUK, Kakinada, NIRF Ranking (251-300 Band), Accredited by NBA (Tier-I) & NAAC with 'A+' Grade
Kotappakonda Road, Yellamanda (Post), Narasaraopet - 522601, Palnadu Dist., Andhra Pradesh, INDIA. Website:www.nrtec.in

Paper ID
NECICAIEA-2K24-205

International Conference on
Artificial Intelligence and Its Emerging Areas
NEC-ICAIEA-2K24
12th & 13th April, 2024
Organized by Departments of CSE, IT, CSE(AI), CSE(AI&ML), CSE(DS), CSE(CS) & MCA in Association with CSI

Certificate of Presentation

This is to Certify that Akula Madhava Rao , Narasaraopeta engineering college has presented the paper title **Employee salary analysis and prediction using machine learning algorithms** in the International Conference on Artificial Intelligence and Its Emerging Areas-2K24 [NEC-ICAIEA-2K24], Organized by Department of Computer Science and Engineering, CSE(AI),IT,CSE(AIML),CSE(DS),CSE(CS) and MCA in Association with CSI on 12th and 13th April 2024 at NARASARAOPETA ENGINEERING COLLEGE (AUTONOMOUS), Narasaraopet, A.P., India.

Convenor
Dr.S.V.N.Srinivasu

Chief-Convenor
Dr.S.N.Tirumala Rao

Principal, Patron
Dr. M. Sreenivasa Kumar

NARASARAOPETA ENGINEERING COLLEGE
(AUTONOMOUS)

Approved by AICTE, Permanently Affiliated to JNTUK, Kakinada, NIRF Ranking (251-300 Band), Accredited by NBA (Tier-I) & NAAC with 'A+' Grade
Kotappakonda Road, Yellamanda (Post), Narasaraopet - 522601, Palnadu Dist., Andhra Pradesh, INDIA. Website:www.nrtec.in

International Conference on

Paper ID
NECICAIEA-2K24-205

Artificial Intelligence and Its Emerging Areas

NEC-ICAIEA-2K24

12th & 13th April, 2024

Organized by Departments of CSE, IT, CSE(AI), CSE(AI&ML), CSE(DS), CSE(CS) & MCA in Association with CSI

Certificate of Presentation

This is to Certify that Udala Sai Kumar , Narasaraopeta engineering college has presented the paper title **Employee salary analysis and prediction using machine learning algorithms** in the International Conference on Artificial Intelligence and Its Emerging Areas-2K24 [NEC-ICAIEA-2K24], Organized by Department of Computer Science and Engineering, CSE(AI),IT,CSE(AIML),CSE(DS),CSE(CS) and MCA in Association with CSI on 12th and 13th April 2024 at NARASARAOPETA ENGINEERING COLLEGE (AUTONOMOUS), Narasaraopet, A.P., India.

Convenor
Dr.S.V.N.Srinivasu

Chief-Convenor
Dr.S.N.Tirumala Rao

Principal, Patron
Dr. M. Sreenivasa Kumar