SALARY PREDICTION

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

D.AJAY BABU	**(19471A05L4)**
P. VENKATA ROHITH	**(19471A05N9)**
B. SANDEEP REDDY	**(19471A05P5)**

Under the esteemed guidance of

Mr.M.Satyam Reddy **M.Tech**

Assistant Professor

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

NARASARAOPETA ENGINEERING COLLEGE

AUTONOMOUS

Accredited by NAAC with A+ Grade and NBA under Cycle -1
Approved by AICTE, New Delhi, Permanently Affiliated to JNTUK, Kakinada
KOTAPPAKONDA ROAD, YALAMANDA VILLAGE, NARASARAOPET- 522601

2022-2023

NARASARAOPETA ENGINEERING COLLEGE
(AUTONOMOUS)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the project that is entitled with the name "SALARY PREDICTION" is a bonafide work done by the team D.AJAY BABU(19471A05L4),P.VENKATA ROHITH(19471A05N9),B.SANDEEP REDDY (19471A05P5) 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 2022-2023.

PROJECT GUIDE

Mr.M..Satyam Reddy M.Tech
Asst. Professor

PROJECT CO-ORDINATOR

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

HEAD OF THE DEPARTMENT

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

EXTERNAL EXAMINER

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(UK).,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 **Mr.M.Satyam Reddy** **M.Tech**, of CSE department whose valuable guidance and unstinting encouragement enable us to accomplish our project successfully in time.

We extend our sincere thanks towards **Mrs. M. Sireesha** **M.Tech.,Ph.D**, 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

D.AJAY BABU **(19471A05L4)**
P.VANKATA ROHITH **(19471A05N9)**
B. SANDEEP REDDY **(19471A05P5)**

NARASARAOPETA
NEC 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
NEC ENGINEERING COLLEGE
(AUTONOMOUS)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

VISION OF THE DEPARTMENT

To become a centre 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: Mould the students to become Software Professionals, Researchers and Entrepreneurs by providing advanced laboratories.

M2: Impart high quality professional training to get expertize 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
NEC ENGINEERING COLLEGE
(AUTONOMOUS)

Program Educational Objectives (PEO's)

The graduates of the programme 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
NEC 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 analyse 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 modelling 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
NEC ENGINEERING COLLEGE
(AUTONOMOUS)

Project Course Outcomes (CO'S):

CO425.1: Analyse the System of Examinations and identify the problem.

CO425.2: Identify and classify the requirements.

CO425.3: Review the Related Literature.

CO425.4: Design and Modularize the project.

CO425.5: Construct, Integrate, Test and Implement the Project.

CO425.6: 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
C425.1		✓											✓		
C425.2	✓		✓		✓								✓		
C425.3				✓		✓	✓	✓					✓		
C425.4			✓			✓	✓	✓					✓	✓	
C425.5					✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
C425.6									✓	✓	✓		✓	✓	

Course Outcomes – Program Outcome correlation

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
C425.1	2	3											2		
C425.2			2		3								2		
C425.3				2		2	3	3					2		
C425.4			2			1	1	2					3	2	
C425.5					3	3	3	2	3	2	2	1	3	2	1
C425.6									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 applied in this project	Description of the device	Attained PO
C3.2.4, C3.2.5	Gathering the requirements and defining the problem, plan to develop a smart bottle for health care using sensors.	PO1, PO3
CC4.2.5	Each and every requirement is critically analyzed, the process model is identified and divided into five modules	PO2, PO3
CC4.2.5	Logical design is done by using the unified modelling language which involves individual team work	PO3, PO5, PO9
CC4.2.5	Each and every module is tested, integrated, and evaluated in our project	PO1, PO5
CC4.2.5	Documentation is done by all our four members in the form of a group	PO10
CC4.2.5	Each and every phase of the work in group is presented periodically	PO10, PO11
CC4.2.5	Implementation is done and the project will be handled by the hospital management and in future updates in our project can be done based on air bubbles occurring in liquid in saline.	PO4, PO7
CC4.2.8 CC4.2.	The physical design includes hardware components like sensors, gsm module, software and Arduino.	PO5, PO6

ABSTRACT

The salary prediction machine learning project aims to develop a model that can predict the salary of an employee based on various features such as education, experience, job title, location, and industry. The project utilizes a dataset containing information about the employees and their corresponding salaries. The model is trained using various regression techniques such as linear regression, decision tree regression, and random forest regression to identify the most accurate and reliable approach for predicting salaries. The performance of the model is evaluated using various metrics such as mean absolute error, mean squared error, and R-squared. The results demonstrate that the developed model can accurately predict salaries with a high degree of precision and can be useful in assisting employers in making informed decisions regarding salaries and compensation packages.

The Salary Prediction project aims to predict the salaries of employees based on various factors such as job title, years of experience, location, education level, and industry. The project uses machine learning algorithms to analyze a dataset of historical salary and employee information to develop a model that can accurately predict future salaries.The dataset used in this project contains information such as job title, years of experience, location, education level, and industry for thousands of employees. The dataset is cleaned and preprocessed to remove any missing or irrelevant data.Various machine learning algorithms such as linear regression, decision tree, and random forest are used to develop the salary prediction model. The performance of each algorithm is evaluated using metrics such as Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and R-squared.The final model is then deployed to make salary predictions for new employees based on their job title, years of experience, location, education level, and industry. The model can be used by companies to help with salary negotiations, employee retention, and workforce planning. The salary prediction project has the potential to help job seekers, employers, and policymakers make informed decisions about salaries and compensation packages. By providing accurate predictions of salaries, this model could help job seekers negotiate better salaries, assist employers in setting appropriate salaries for their employees, and help policymakers make informed decisions about minimum wages and labor policies.Keywords—MachineLearning , LinearRegression,RandomForest,Logisticregression.

INDEX

LIST OF FIGURES

1. INTRODUCTION

1.1 Introduction

The project on salary prediction using machine learning! In this project, we will explore how we can use machine learning algorithms to predict the salary of an employee based on various features such as education, experience, job location, etc.

The ability to predict salaries accurately can be very beneficial for both employees and employers. For employees, it can help them negotiate better salaries and plan their career growth. For employers, it can help them determine fair compensation packages, attract top talent, and improve retention rates.

Machine learning algorithms can learn patterns and relationships from historical data and use that information to make predictions on new, unseen data. In this project, we will use a dataset of employee information, including their education, experience, job location, and current salary, to train a machine learning model. Once trained, the model can be used to predict the salary of new employees based on their features.

We will start by exploring the dataset, performing data cleaning and feature engineering, and then train and evaluate different machine learning models. We will also tune hyperparameters and use various evaluation metrics to ensure that the model performs well on both training and test data.

By the end of this project, we will have a well-performing machine learning model that can accurately predict salaries based on various features. Let's get started!

1.2 RELATED WORK

There have been several studies and research papers on salary prediction using machine learning. Here are some related works:

"Predicting Salaries for Job Postings" by Hao Zhong, Xinyuan Zhou, and Jie Zhang: This paper presents a method for predicting salaries based on job postings. The authors used a dataset of job postings and salaries to train a machine learning model that can predict salaries based on job titles, locations, company sizes, and other features.

"Predicting Employee Salaries from Anonymous Resume Data" by Rayid Ghani, Richard E. Langley, and D. Alex Hughes: In this study, the authors used machine learning algorithms to predict salaries based on anonymous resume data. They used a dataset of resumes and salaries to train the model, which was able to predict salaries with high accuracy.

"Predicting Salaries of Software Engineers Using Machine Learning Techniques" by Alaa M. El-Halees and Nada N. Alrabaiah: This paper presents a method for predicting salaries of software engineers based on various features such as education, experience, programming language skills, and job location. The authors used a dataset of software engineers and their salaries to train a machine learning model that can predict salaries based on these features.

"Salary Prediction Using Machine Learning Techniques" by Yuhang Chen and Hanzhong Liu: This study proposes a method for predicting salaries based on features such as education, experience, job type, and industry. The authors used a dataset of job seekers and their salaries to train a machine learning model that can predict salaries based on these features.

Methods :There are several methods that can be used for salary prediction using machine learning. Here are some commonly used methods:

Linear Regression: Linear regression is a simple and widely used method for predicting continuous values such as salaries. In this method, we assume a linear relationship between the input features and the output salary. The model learns the coefficients of the linear equation through a process called gradient descent.

Decision Trees: Decision trees are a popular method for regression problems such as salary prediction. In this method, the model creates a tree-like structure to make decisions based on the input features. The tree is built recursively by selecting the best feature to split the data and minimizing the variance of the target variable.

Random Forest: Random forest is an ensemble method that combines multiple decision trees to improve the accuracy of the prediction. In this method, multiple decision trees are trained on different subsets of the data, and their predictions are combined to make a final prediction.

Gradient Boosting: Gradient boosting is another ensemble method that combines multiple weak learners to create a strong learner. In this method, the model learns from the mistakes of the previous weak learners and improves its predictions with each iteration.

Neural Networks: Neural networks are a powerful method for predicting salaries. In this method, the model learns complex non-linear relationships between the input features and the output salary through multiple layers of neurons. The model is trained using backpropagation, which adjusts the weights of the neurons to minimize the error between the predicted and actual salaries.

These are some of the commonly used methods for salary prediction using machine learning. The choice of method depends on the size of the dataset, the complexity of the problem, and the accuracy required. It is often useful to try multiple methods and evaluate their performance to select the best one for the given problem.

1.3 EXISTING SYSTEM

The existing system usually for a license plate detection and character recognition (LPDR) system has mainly three phases. The first phase is image pre-processing, once the image is captured further processing of the image is carried out like converting the image from a color space to another, resizing the image resolution, and removing noises. The second phase is license plate localization, the region of interest is detected based on some license plate characteristics and image features. The final phase is the optical character recognition, this phase is considered the most crucial step because it helps to read the plate number and identify the vehicle.

DISADVANTAGES

A variety of algorithms are developed for this work and each one has advantages and disadvantages for extracting plates in images under different circumstances. However, the complexity of some methods requires a high calculation cost and this could be time-consuming.

From literature survey it has been observed that there are certain limitations about proposed algorithms like:

1. Data Quality
2. Overfitting
3. Bias
4. Ethical considerations
5. Limited factors

1.4 PROPOSED SYSTEM

A proposed system for salary prediction using machine learning can be divided into the following steps:

Data Collection: The first step is to collect a large and representative dataset containing information about employees such as their education level, job title, work experience, skills, and other relevant factors that may affect their salary.

Data Cleaning and Preprocessing: After collecting the dataset, it is essential to clean and preprocess the data by removing duplicates, missing values, and outliers. Data normalization and feature scaling may also be applied to ensure that all features have the same scale.

Feature Selection: Feature selection is the process of selecting the most important features that are relevant to salary prediction. This step helps to reduce the dimensionality of the dataset, improve the model's performance, and avoid overfitting.

Model Selection and Training: In this step, different machine learning models such as linear regression, decision trees, or neural networks are trained on the preprocessed dataset. The performance of each model is evaluated using various metrics such as mean squared error, accuracy, or R-squared, and the best-performing model is selected.

Model Evaluation and Testing: After selecting the best model, it is tested on a separate testing dataset to evaluate its performance on new, unseen data. This step helps to ensure that the model can generalize well and avoid overfitting.

Deployment and Monitoring: Once the model is trained and tested, it can be deployed into a production environment to predict salaries for new employees. It is essential to monitor the model's performance regularly and retrain the model with new data to ensure its accuracy and effectiveness over time.

User Interface: Finally, a user interface can be developed to allow users to input the relevant employee information and obtain the predicted salary. The user interface can be developed using various programming languages and frameworks such as Python, Flask, or Django.

These are the main steps involved in a proposed system for salary prediction using machine learning. The implementation details may vary depending on the specific requirements of the project.

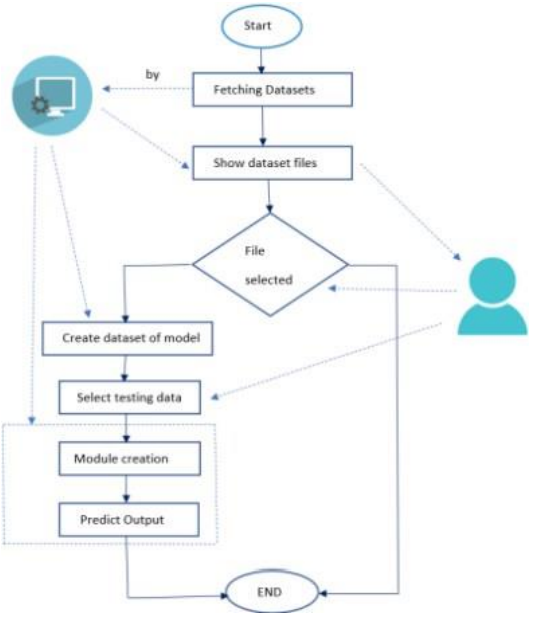


Fig:1.1 Proposed System and process

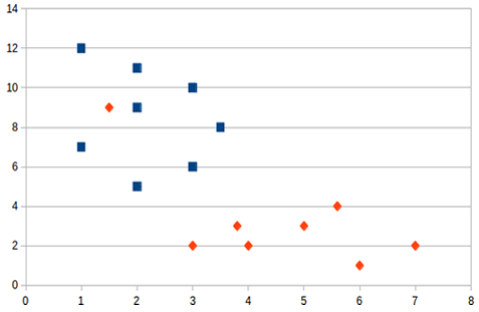


Fig: 1.2 Training set

ADVANTAGES

1. Accuracy
2. Efficiency
3. Objectivity
4. Scalability
5. Customizability
6. Data driven decision making

1.5 SYSTEM REQUIREMENTS

1.5.1 Hardware Requirements:

- Processor : Intel(R) Core™2 i7-5500U CPU @2.50GHz
- RAM : 8GB(gigabyte)
- System Type : 64-bit operating system, x64-based processor

1.5.2 Software Requirements:

- Browser : Any Latest browser like Chrome
- Operating System : Windows 10
- Language : Python
- Platform : Google COLAB

2. LITERATURE SURVEY

2.1 Machine Learning

There have been several studies on salary prediction in recent years, focusing on different aspects of the problem. Some studies have looked at the impact of various factors on salary, such as education level, job experience, and industry sector. Other studies have explored the use of different machine learning algorithms for salary prediction.Predicting Salary from Job Posting Text" by Ahmad A. Tafti and Arman Cohan (2018) - This paper explores the use of natural language processing (NLP) techniques to predict salaries from job postings.

The authors use various machine learning algorithms,such as linear regression and support vector machines, and compare their performance.Predicting Salaries with Machine Learning" by Arjun Adhikari (2019) - This article provides a step-by-step guide on how to build a salary prediction model using Python and scikit-learn. The author explains various preprocessing techniques, feature engineering, and model selection.A Deep Learning Model for Predicting Job Salaries" by Kai Shu et al. (2018) - This paper proposes a deep learning model for predicting salaries based on job descriptions.

The authors use a combination of convolutional neural networks (CNNs) and recurrent neural networks (RNNs) to capture both the semantic and syntactic information in the text.Predicting Employee Salary Using Machine Learning Techniques" by Ayesha Waheed et al. (2019) - This paper presents a study on predicting employee salaries using various machine learning algorithms, such as decision trees, random forests, and gradient boosting.

The authors evaluate the performance of these algorithms on a real-world dataset.Predicting Salaries for Job Postings Using Machine Learning" by Brian Dew et al. (2018) - This article discusses a project in which the authors built a salary prediction model using job postings and salary data from Glassdoor. The authors use various NLP techniques, such as word embeddings and topic modeling, to extract features from the job postings.These papers and articles demonstrate the wide range of techniques and approaches that can be used for salary prediction, from traditional machine learning algorithms to deep learning models and NLP techniques

2.2 Some Machine Learning Types

1. **Feedforward neural network:**
   - This type of neural network is the very basic neural network where the flow control occurs from the input layer and goes towards the output layer.
   - These kinds of networks are only having single layers or only 1 hidden layer.
   - Since the data moves only in 1 direction there is no backpropagation technique in this network.

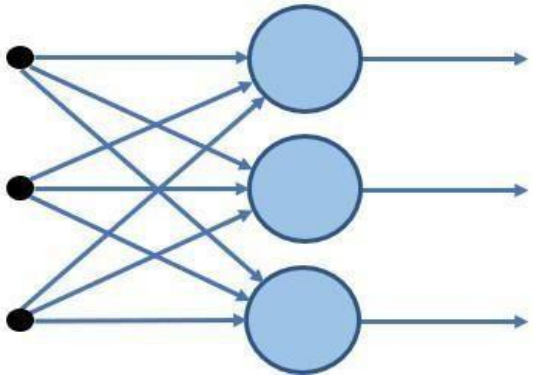


Fig:2.1 Feed Forward Neural Network

- In this network, the sum of the weights present in the input is fed into the input layer.
- These kinds of networks are used in the facial recognition algorithm using computer vision.

2. Radial basis function neural networks:

- This kind of neural networks have generally more than 1 layer preferably two layers.
- In this kind of networks, the relative distance from any point to the center is calculated and the same is passed towards the next layer
- Radial basis networks are generally used in the power restoration systems to restore the power in the shortest span of time to avoid the blackouts.

3. Multi-Layer perceptron:

- This type of network are having more than 3 layers and its used to classify the data which is not linear.
- These kinds of networks are fully connected with every node.
- These networks are extensively used for speech recognition and other machine learning technologies.

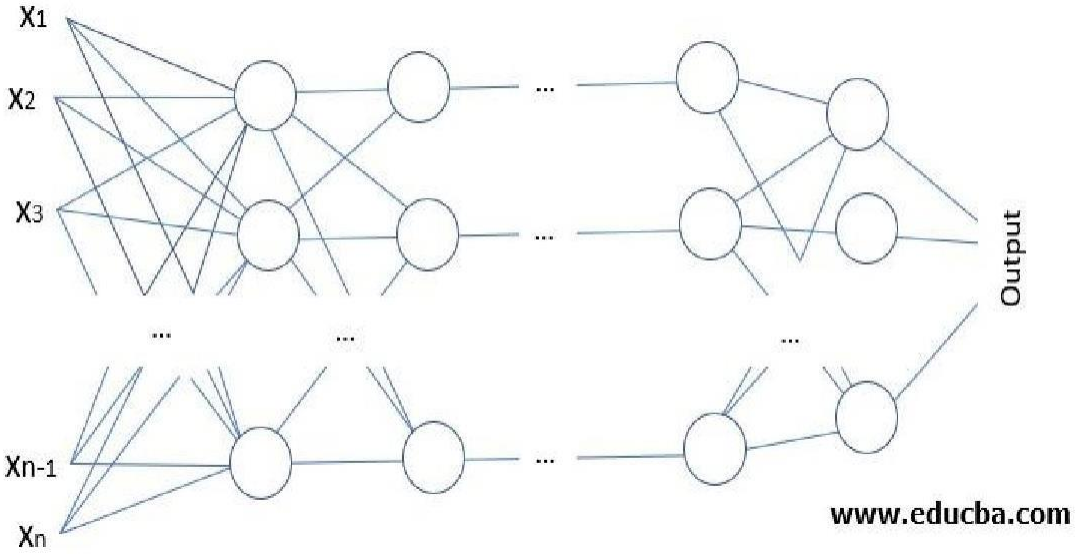


Fig: 2.2 Multi-Layer Perceptron

4. Convolution Neural Network

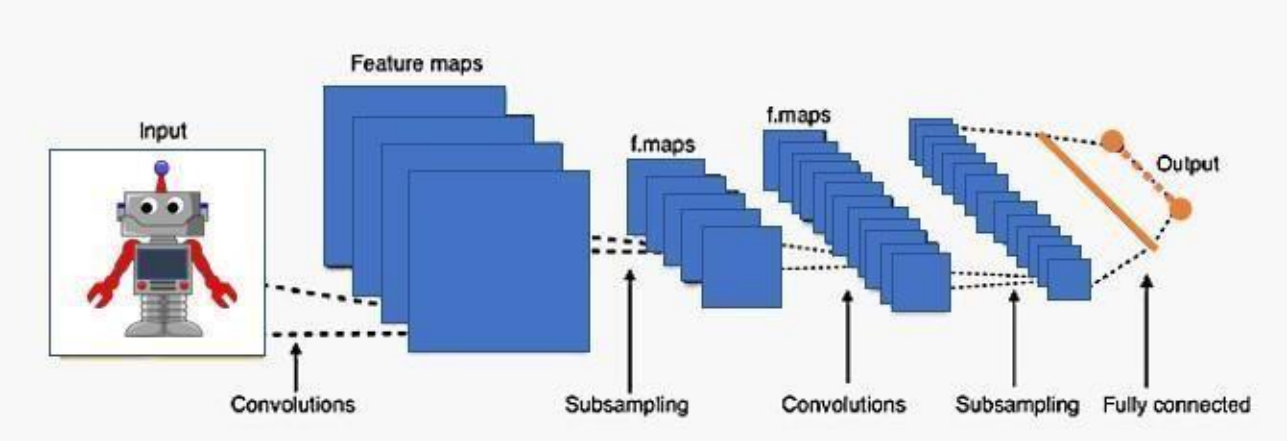


Fig:2.3 Convolution Neural Network

- CNN is one of the variations of the multilayer perceptron.
- CNN can contain more than 1 convolution layer and since it contains a convolution layer the network is very deep with fewer parameters.
- CNN is very effective for image recognition and identifying different image patterns.

5. Recurrent Neural Network:

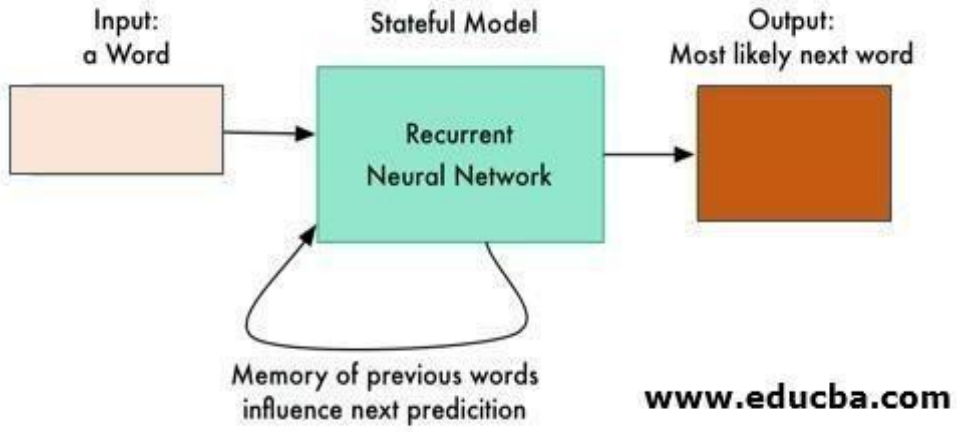


Fig:2.4 Recurrent Neural Network

- RNN is a type of neural network where the output of a particular neuron is fed back as an input to the same node.
- These method helps the network to predict the output.
- This kind of network is useful in maintaining a small state of memory which is very useful for developing the chatbot
- This kind of network is used in chatbot development and text to speech technologies.

6. Modular Neural Network:

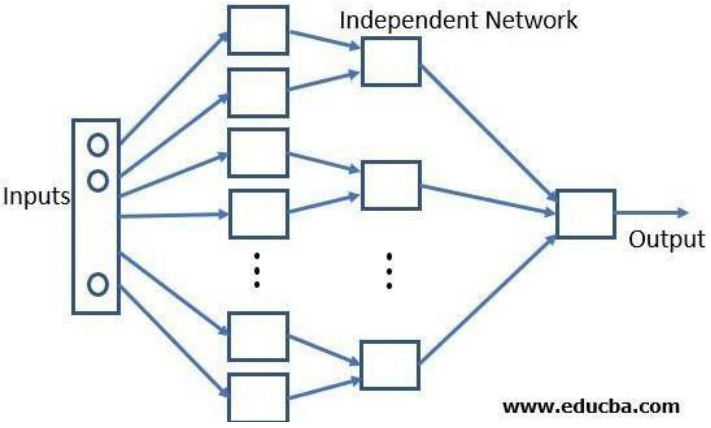


Fig:2.5 Modular Neural Network

- This kind of network is not a single network but a combination of multiple small neural networks.
- All the sub-networks make a big neural network and all of them work independently to achieve a common target.
- These networks are very helpful in breaking the small-large problem into small pieces and then solving it.

7. Sequence to sequence models:

- This type of network is generally a combination of two RNN networks.
- The network works on the encoding and decoding that is it consists of the encoder which is used to process the input and there is a decoder which processes the output.
- Generally, this kind of network is used for text processing where the length of the inputted text is not as same as output.

2.3 Applications of Machine Learning:

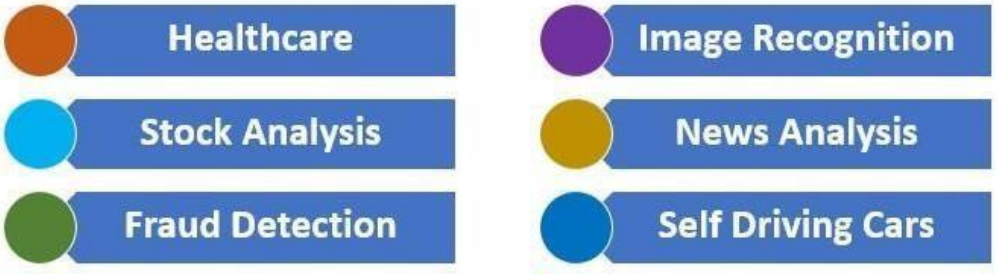


Fig: 2.6 Applications of Deep Learning

1. Self Driving Cars
2. News Aggregation and Fraud News Detection
3. Natural Language Processing
4. Virtual Assistants
5. Entertainment
6. Visual Recognition
7. Fraud Detection
8. Healthcare
9. Personalizations
10. Detecting Developmental Delay in Children
11. Colourisation of Black and White images
12. Adding sounds to silent movies
13. Automatic Machine Translation
14. Automatic Handwriting Generation
15. Automatic Game Playing
16. Language Translations
17. Pixel Restoration
18. Photo Descriptions
19. Demographic and Election Predictions
20. Deep Dreaming

2.4 Characteristics of Machine Learning

1. Supervised, Semi-Supervised or Unsupervised

When the category labels are present while you train the data then it is Supervised learning. Algorithms like Linear regression. Logistic regression, decision trees use Supervised Learning.
When category labels are not known while you train data then it is unsupervised learning.
Algorithms like Cluster Analysis, K means clustering, Anomaly detection uses Unsupervised Learning.
The data set consists of both labeled and unlabelled data then we call it is Semi-Supervised learning. Graph-based models, Generative models, cluster assumption, continuity assumption use Semi-Supervised learning.

2. Huge Amount of Resources

It needs advanced Graphical Processing Units for processing heavy workloads. A huge amount of data needs to be processed like Big data in the form of structured or unstructured data.
Sometimes more time also required to process the data, it depends on the amount of data fed in.

3. Large Amount of Layers in Model

A huge amount of layers like input, activation, the output will be required, sometimes the output of one layer can be input to another layer by making few small findings and then these findings are summed up finally in the softmax layer to find out a broader classification for final output.

4. Optimizing Hyper-parameters

Hyperparameters like no of epochs, Batch size, No of layers, Learning rate, needs to be tuned well for successful Model accuracy because it creates a link between layer predictions to final output prediction. Over-fitting and under-fitting can be well handled with hyper-parameters.

5. Cost Function

It says how well the model performance in prediction and accuracy. For each iteration in Deep Learning Model, the goal is to minimize the cost when compared to previous iterations. Mean absolute error, Mean Squared Error, Hinge loss, Cross entropy are different types according to different algorithms used.

2.5 Advantages of Machine Learning

- Solve Complex problems like Audio processing in Amazon echo, Image recognition, etc, reduce the need for feature extraction, automated tasks wherein predictions can be done in less time using Keras and Tensorflow.
- Parallel computing can be done thus reducing overheads.
- Models can be trained on a huge amount of data and the model gets better with more data.
- High-Quality Predictions when compared with humans by training tirelessly.
- Works well-unstructured data like video clips, documents, sensor data, webcam data, etc.

2.6 Machine Learning Algorithms

To create a deep learning model, one must write several algorithms, blend them together and create a net of neurons. Deep learning has a high computational cost. To aid deep learning models, there are deep learning platforms like Tensor flow, Py-Torch, Chainer, Keras, etc. In deep learning, we have tried to replicate the human neural network with an artificial neural network; the human neuron is called perceptron in the deep learning model. We connect these perceptron units together to create a neural network; it has 3 sections:

1. Input layer
2. Hidden layers
3. Output layer

A perceptron has input nodes (dendrites in the human brain), an actuation function to make a small decision and output nodes (axon in the human brain). We will see how one perceptron works; connecting them together will create a deep learning model. Input information (number of input variables/features) are assigned some weight and fed to the actuation function. The actuation function makes a decision and sends output. This perceptron's output will be input to other neurons. Once the batch is processed, with backpropagation error is calculated at each neuron, with the help of a cost function/ cross-entropy. In this way, input weights are reassigned, and the whole process continues until cross-entropy satisfies the condition.

We have different actuation functions like Sigmoid functions, hyperbolic tangent function, Rectified Linear Unit (ReLU) to take a small decision. A deep learning model needs a vast amount of data to build a good model. Generally, a model with more than 3 hidden layers is treated as a deep neural network. Basically, Deep learning is a set of neurons with a number of parameters defined for each layer. To create the Deep Learning model, the popular architectures are RNN, CNN, etc.

2.7 Architectural Methods for Machine Learning Algorithms

To build this architecture following algorithms are used:

1. Back Propagation

In this algorithm, we calculate partial derivatives. In general, the gradient descent method for optimization, derivatives (gradients) are calculated at each iteration. In deep learning, functions are not simple; they are the composition of different functions. In this case, it is hard to calculate gradients, so we use approximate differentiation to calculate derivatives. The more the number of parameters, the more expensive approximate differentiation will be.

2. Stochastic Gradient Descent

In Gradient descent, the goal is to find global minima or optimum solution. But to get that, we have to consider local minima solutions (not desirable) also. If the objective function is a convex function, it is easy to find the global minima. The initial value for the function and learning rate are deciding parameters for finding global minima. This can easily be understood by considering a river from the mountain top and searching for a foothill (global minima). But in the way, there will be some ups and downs (local minima) which must be avoided. The river originating point and speed (initial value and learning rate in our case) are deciding factors to find global minima.

3. Learning Rate

The learning rate is like the speed of the river; it can reduce training time and increase performance. In general, to learn any technique/sport, in the beginning, the learning rate is relatively high than at the end when one is to master it. After the intermediate stage, the learning will be slow; the focus will be on fine-tuning. The same is applied in deep learning; too large changes are tackled by a higher learning rate and by slowly decreasing the learning rate later for fine-tuning.

4. Batch Normalization

In deep learning initial value of weight (randomly chosen) and learning rate is defined for a minibatch. In the beginning, there would be many outliers, and during backpropagation, these outliers must be compensated to compute the weights to get output. This compensation results in extra epochs. So to avoid it, we use batch normalization.

5. Drop Out

In deep learning, we generally encounter the problem of overfitting. Overfitting in large networks with several parameters makes it difficult to predict on test data. So, to avoid that, we use the dropout method, which drops random units during training by creating different ‘thinned networks’. When testing these thinned networks’ predictions are averaged, which helps to avoid overfitting.

6. Bag of Words

We use a continuous bag of words to predict the next word. For e.g., we see in email writing the autosuggestion for completing the sentence is part of NLP. This is done by considering lots of sentences and for a specific word surrounding words that are captured. These specific words and surrounding words are fed to the neural network. After the training model, it can predict the specific word based on the surrounding words.

7. Long Short Term Memory

LSTM is very useful in sequence prediction problems like language translation, predicting sales and finding the stock price. LSTM has the edge over other techniques because it is able to consider previous data. LSTM makes modification by cell states mechanism. It remembers to forget things. The 3 main aspects of LSTM make it stand out from other deep learning techniques. First is when the neuron should have input, second when to remember previous data and what to forget and third is when to pass output.

2.8 Libraries of Machine Learning

All the libraries which are generally used for deep learning are open source and few of them are as follows:

- TensorFlow
- deeplearning4j
- Torch
- Caffe
- Microsoft CNTK
- ML.NET
- Theano
- Deepmat
- Neon

1. TensorFlow

- TensorFlow is the machine learning and deep learning library developed by Google and it came into the market around the 2016 march.
- TensorFlow grew out of an in-house library of google brain known as DistBelief.
- Currently, TensorFlow is the leading and most used library in the market.
- Different types of deep nets can be developed and also the various packages available in this library are used to attain and address most of the tasks and problems in the field of deep learning.
- This library is completely written in python and so it's easy to use for the python programmers.
- Due to a flexible computational graphical structure of TensorFlow, this library is not only limited to deep learning operations it can be used for many different operations and applications.
- TensorFlow provides a layer or we can say more of a wrapper, known as Keras which is used to access the different packages and methods easily of TensorFlow.
- Google provides a very well documentation for this library where every small intricacies and usage are mentioned anyone can refer that and use the library.

- TensorFlow is a very fast-evolving library, this library can be used for educational purposes as well as huge large scale commercial application can also be built.
- Google has developed this library for Mobil platforms as well which is known as TensorFlow lite.
- TensorFlow is the only library which provides support for Python, Java, C++, javascript and swift programming language, for Javascript TensorFlow.js
- TensorFlow has also support for GPU and big data.

2. Deeplearning4j

- Deeplearning4j is the open-source java library which only supports java programming language and this library is written in Java.
- This was developed by Adam Gibson to provide distributed multimode capabilities for deep neural networks.
- This library is very much use full for the application which is having build on top of big data.
- This library works with Scala and also provide inbuilt GPU support.

3. Torch

- This open-source deep-learning library was developed by Facebook and Twitter.
- This library is written in Lua programming language.
- However PyTorch is the library which is widely used, and it's written in a python programming language.

4. Caffe

- Caffe is an open-source deep-learning library written in C++/CUDA and developed by Yangqing Jia of Google.
- This library was first developed for computer vision tax.
- Caffe gives permission to the user to configure the hyperparameters for a deep net.
- The layer configuration is very robust and very much sophisticated.

5. Theano

- This is the open-source deep-learning library written in Python and CUDA.
- This library is very similar to the TensorFlow library but the implementation and usage are not that simple as that of TensorFlow.
- This library is generally used for educational and research purposes.
- Theano is not that easy to use and many deep learning libraries extend the features of this library to help ease the life of the developer for coding the deep learning models.
- Theano is fastest amongst most of the libraries mentioned because it makes use of vectors and matrices for all the functions and the vectorized code runs faster since parallel processing for the multiple values makes the things faster.

6. Microsoft CNTK

- This is a cognitive toolkit developed by Microsoft to venture in the field of Artificial intelligence.
- This library is written in python and its supports the other packages and libraries which python programming language supports, and it comes with Microsoft visual studio.

- CNTK is used to describe neural networks as a series of computational directed graphs.

7. ML.NET

- ML.NET is the open-source library which is also developed by Microsoft for the dot net developers.
- This library is written in C# and F# and it uses the Microsoft dot net platform.
- With the library, it becomes easy to create desktop as well as large scale web applications which can bring the vast possibility of the machine learning algorithm to the end-user.

8. Deepmat

- This library is developed in MATLAB.
- With the use of this library, we can implement deep learning using MATLAB.
- with this library GSN, CNN, Restricted Boltzmann machine, Deep belief networks,multi-layer perceptron, and many more artificial neural networks.

9. Neon

- Neon is a deep learning framework created by the Nervana systems to deliver industry-leading cutting edge technologies.
- This framework has been depreciated as of 2018 and further research has been carried out by Intel corporation on the same.
- As per the Intel corporation website, alternative frameworks are asked to be used such as Intel optimization for tensorFlow, Intel optimization for Caffe, pytorch etc.

Layers in Convolutional Neural Networks

Below are the Layers of convolutional neural networks:

1. Image Input Layer:

The input layer gives inputs (mostly images), and normalization is carried out. Input size has to be mentioned here.

2. Convolutional Layer:

Convolution is performed in this layer. The image is divided into perceptrons(algorithm); local fields are created, leading to the compression of perceptrons to feature maps as a matrix with size m x n.

3. Non-Linearity Layer:

Here feature maps are taken as input, and activation maps are given as output with the help of the activation function. The activation function is generally implemented as sigmoid or hyperbolic tangent functions.

4. Rectification Layer:

The crucial component of CNN, this layer does the training faster without reducing accuracy. It performs element-wise absolute value operation on activation maps.

5. Rectified Linear Units (ReLU):

ReLU combines non-linear and rectification layers on CNN. This does the threshold operation where negative values are converted to zero. However, ReLU doesn't change the size of the input.

6. Pooling Layer:

The pooling layer is also called the downsampling layer, as this is responsible for reducing the size of activation maps. A filter and stride of the same length are applied to the input volume. This layer ignores less significant data; hence image recognition is done in a smaller representation. This layer reduces overfitting. Since the amount of parameters is reduced using the pooling layer, the cost is also reduced. The input is divided into rectangular pooling regions, and either maximum or average is calculated, which returns maximum or average consequently. Max Pooling is a popular one.

7. Dropout Layer:

This layer randomly sets the input layer to zero with a given probability. More results in different elements are dropped after this operation. This layer also helps to reduce overfitting. It makes the network to be redundant. No learning happens in this layer. This operation is carried out only during training.

8. Fully Connected Layer:

Activation maps, which are the output of previous layers, is turned into a class probability distribution in this layer. FC layer multiplies the input by a weight matrix and adds the bias vector.

9. Output Layer:

FC layer is followed by softmax and classification layers. The softmax function is applied to the input. The classification layer computes the cross-entropy and loss function for classification problems.

10. Regression Layer:

Half the mean squared error is computed in this layer. This layer should follow the FC layer.

Common steps for any Tensorflow based Algorithms:

The basic steps of TensorFlow algorithm are:

Step 1: Data is Imported/Generated: TensorFlow Models depends heavily on the huge amount of Data. Either you can import your own dataset or TensorFlow also comes with the collection of Type this command to check out available datasets in TensorFlow.

```
import TensorFlow as tf
import TensorFlow_datasets as tendata
#This command will generate a list of datasets available in the TensorFlow
print(tfds.list_builders()
```

Step 2: Data Normalization or Transformation: If the data is not in the appropriate forum. The Batch Normalization is the command approach used to normalize data in the TensorFlow.

Step 3: Set the Parameters of the Algorithm: For eg; the number of Iterations, Learning rate, etc.

Step 4: Set and initialize the variables and Placeholders: Variables and Placeholders are two basic programming Elements of the TensorFlow. Variables hold the state of the graph and placeholders are used to feed the data in the graph at the later date.

Step 5: Create Model structure: What operations will be performed on the data is defined.

Step 6: Define the Loss Function: It calculates the difference between predicted values and actual values. It tells how well your model is trained basically used to evaluate the output.

Step 7: Train Model: Initialize computational graph and create an Instance of a graph. Feed data into the model with the help of placeholders and let the TensorFlow do the rest of the processing for better predictions.

Step 8: Evaluate the performance: Evaluate the model by checking with new data.

Step 9: Predict the Outcome: Also checks your model on new and unseen data.

To better visualize model TensorFlow provides Tensorboard. It helps us to visualize any statistics of the neural network, debug and optimize them. You can check what happens in the code and will give you a detailed understanding of the inner working. You can fix problems very easily with the help of this tool. Tensorboard provides five types of Visualizations:

- Scalars
- Images
- Audio
- Histograms
- Graphs

The summary function of the TensorFlows gives us a detailed summary according to the specified format. To allocate resources, hold intermediate results and variables and execute graphs or part of graph session function is used.

3. SYSTEM ANALYSIS

3.1 Scope of the project

The scope of this system is to design an efficient automatic authorized salary prediction system byusing the dataset.

3.2 Analysis

The system consists of two modules:Salary prediction analysis.

The dataset was taken from the Kaggle repository which contain 273 images out of which 248 belongs to train category and 25 are test category images. All the images are of jpeg format.

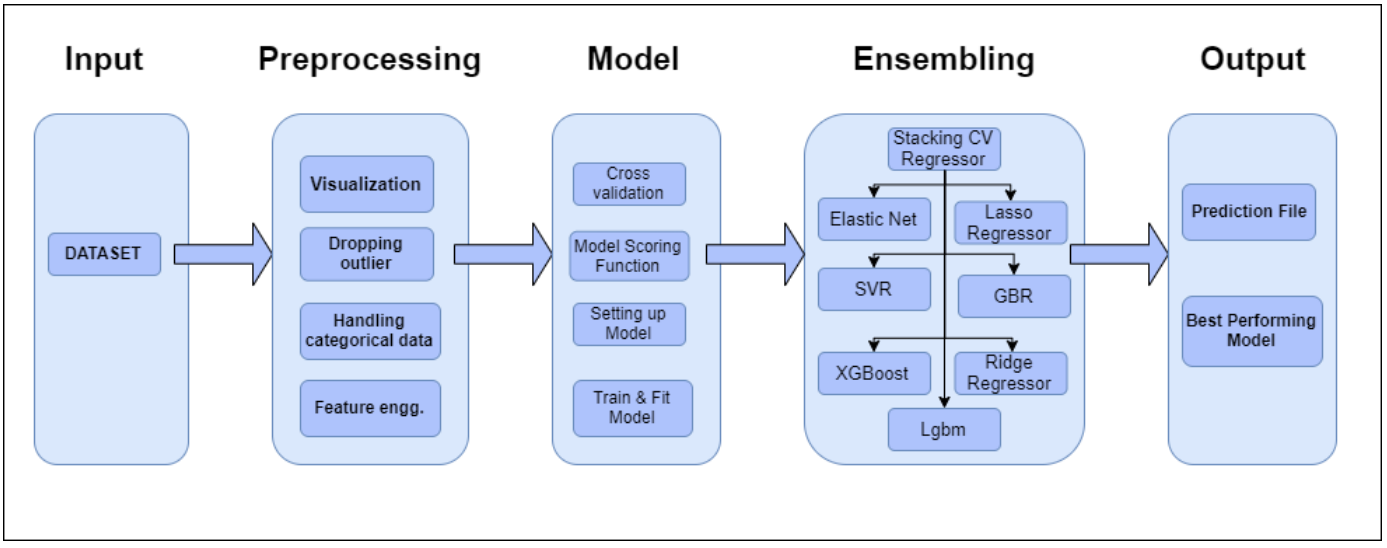


Fig: 3.2. Analysis of the System

Two experiments have been done to test the performance of the proposed detection and recognition system. The first experiment was done on various License Plates using real traffic video sequences collected from different situations in different lighting conditions. The second experiment was done using the dataset with car images, the AOLP dataset. The cars dataset consists of 126 images with a resolution of 890x592 pixels taken during the day from complex outside scenes, each of which contains only one vehicle.

For the classification, we have used a pre-trained CNN model, so that we do not need a large training dataset for successful classification.

3.3 Data preprocessing:

Data preprocessing to be done for the following Neural Network Models.

About Some Neural Network Models:

Google Net Model: Google Net (or Inception V1) was proposed by research at Google (with the collaboration of various universities) in 2014 in the research paper titled "Going Deeper with Convolutions". This architecture was the winner at the ILSVRC 2014 image classification challenge. It has provided a significant decrease in error rate as compared to previous winners AlexNet (Winner of ILSVRC 2012) and ZF-Net (Winner of ILSVRC 2013) and significantly less error rate than VGG (2014 runner up). This architecture uses techniques such as *1×1* convolutions in the middle of the architecture and global average pooling.

- **Global Average Pooling :**

  In the previous architecture such as AlexNet, the fully connected layers are used at the end of the network. These fully connected layers contain the majority of parameters of many architectures that causes an increase in computation cost.
  In GoogLeNet architecture, there is a method called global average pooling is used at the end of the network. This layer takes a feature map of *7×7* and averages it to *1×1*. This also decreases the number of trainable parameters to 0 and improves the top-1 accuracy by 0.6%.

- **Inception Module:**

  The inception module is different from previous architectures such as AlexNet, ZF-Net. In this architecture, there is a fixed convolution size for each layer.
  In the Inception module *1×1, 3×3, 5×5* convolution and *3×3* max pooling performed in a parallel way at the input and the output of these are stacked together to generated final output. The idea behind that convolution filters of different sizes will handle objects at multiple scale better.

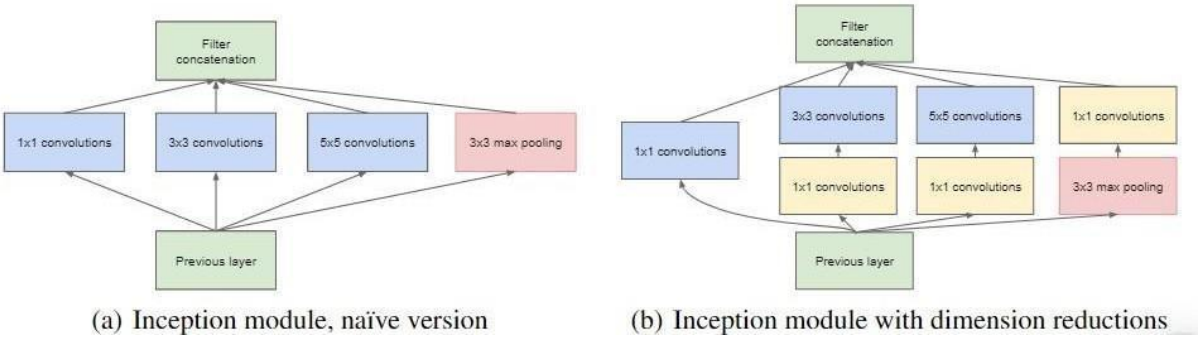


Fig:3.4 Inception Module of GoogleNet

- **Auxiliary Classifier for Training:**

  Inception architecture used some intermediate classifier branches in the middle of the architecture, these branches are used during training only. These branches consist of a 5×5 average pooling layer with a stride of 3, a *1×1* convolutions with *128* filters, two fully connected layers of 1024 outputs and 1000 outputs and a softmax classification layer. The generated loss of these layers added to total loss with a weight of 0.3. These layers help in combating gradient vanishing problem and also provide regularization.

This architecture takes image of size *224 x 224* with RGB color channels. All the convolutions inside thisarchitecture uses Rectified Linear Units (ReLU) as their activation functions.

Results:

GoogLeNet was the winner at ILSRVRC 2014 taking 1st place in both classification an detection task. It has top-5 error rate of 6.67% in classification task. An ensemble of 6 GoogLeNets gives 43.9 % mAP on ImageNet test set.

	age	workclass	education	education_num	marital_status	occupation	relationship	race	sex	capital_gain	capital_loss	hours_per_week	native_country	salary
0	39	State-gov	Bachelors	13.0	Never-married	Adm-clerical	Not-in-family	White	Male	7.684324	0.0	40.0	United-States	<=50K
1	50	Self-emp-not-inc	Bachelors	13.0	Married-civ-spouse	Exec-managerial	Husband	White	Male	0.000000	0.0	32.5	United-States	<=50K
2	38	Private	HS-grad	9.0	Divorced	Handlers-cleaners	Not-in-family	White	Male	0.000000	0.0	40.0	United-States	<=50K
3	53	Private	11th	7.0	Married-civ-spouse	Handlers-cleaners	Husband	Black	Male	0.000000	0.0	40.0	United-States	<=50K
4	28	Private	Bachelors	13.0	Married-civ-spouse	Prof-specialty	Wife	Black	Female	0.000000	0.0	40.0	Cuba	<=50K

Fig:3.8 Results of GoogleNet at ILSRVRC with details of data

Residual Networks:

After the first CNN-based architecture (AlexNet) that win the ImageNet 2012 competition, Every subsequent winning architecture uses more layers in a deep neural network to reduce the error rate. This works for less number of layers, but when we increase the number of layers, there is a common problem in deep learning associated with that called Vanishing/Exploding gradient. This causes the gradient to become 0 or too large. Thus when we increases number of layers, the training and test error rate also increases.

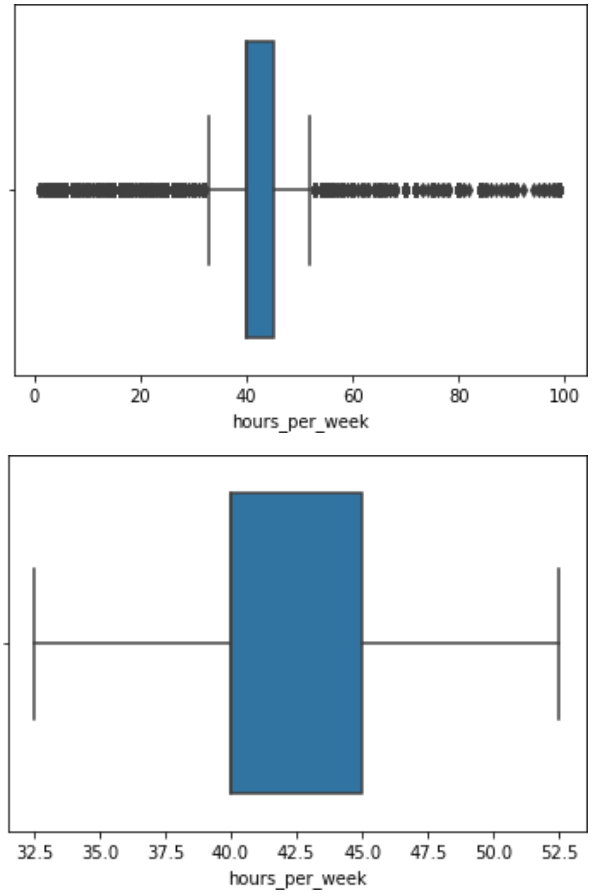


Fig:3.9 Residual Networks with training and testing rates

In the above plot, we can observe that a 56-layer CNN gives more error rate on both training and testing dataset than a 20-layer CNN architecture, If this was the result of over fitting, then we should have lower training error in 56-layer CNN but then it also has higher training error. After analyzing more on error rate the authors were able to reach conclusion that it is caused by vanishing/exploding gradient.
ResNet, which was proposed in 2015 by researchers at Microsoft Research introduced a new architecture called Residual Network.

Residual Block:
In order to solve the problem of the vanishing/exploding gradient, this architecture introduced the concept called Residual Network. In this network we use a technique called ***skip connections*** . The skip connection skips training from a few layers and connects directly to the output.
The approach behind this network is instead of layers learn the underlying mapping, we allow network fit the residual mapping. So, instead of say H(x), initial mapping, let the network fit, $F(x) := H(x) - x$ which gives $H(x) := F(x) + x$.

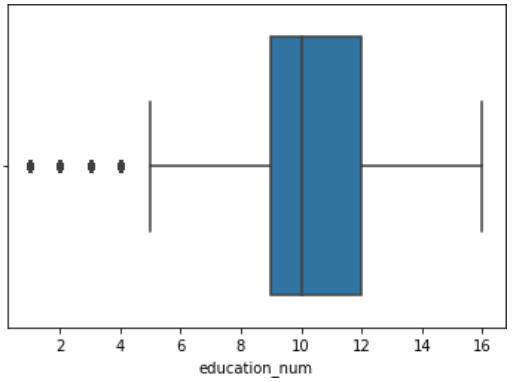


Fig:3.10 analysis of data

The advantage of adding this type of skip connection is because if any layer hurt the performance of architecture then it will be skipped by regularization. So, this results in training very deep neural network without the problems caused by vanishing/exploding gradient. The authors of the paper experimented on 100-1000 layers on CIFAR-10 dataset.

There is a similar approach called “highway networks”, these networks also uses skip connection. Similar to LSTM these skip connections also uses parametric gates. These gates determine how much information passes through the skip connection. This architecture however has not provide accuracy better than ResNet architecture.

Network Architecture:

This network uses a 34-layer plain network architecture inspired by VGG-19 in which then the shortcut connection is added. These shortcut connections then convert the architecture into residual network.

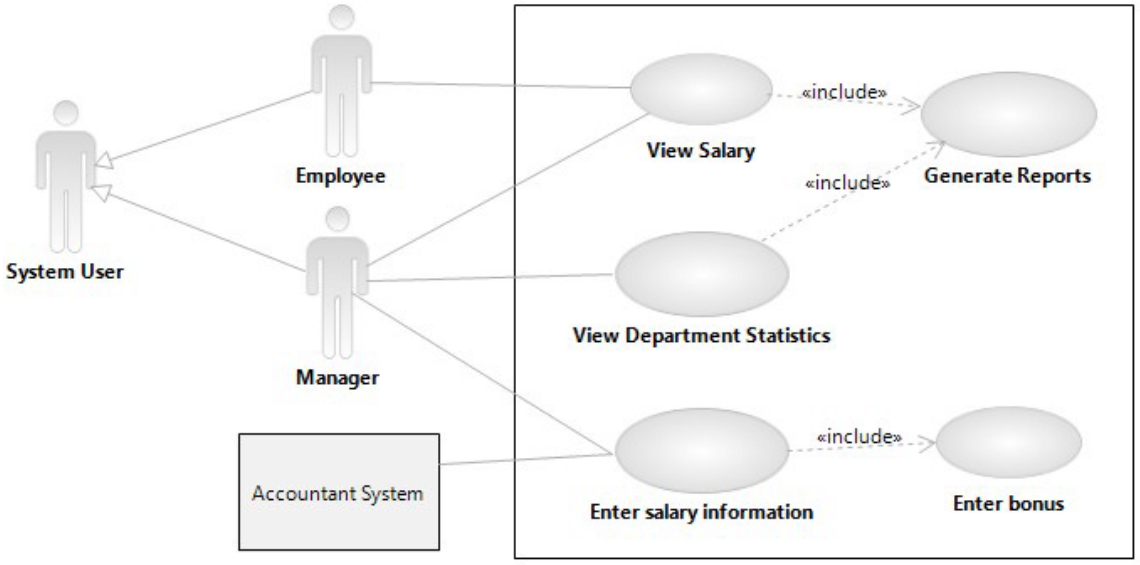


Fig:3.11 Architecture

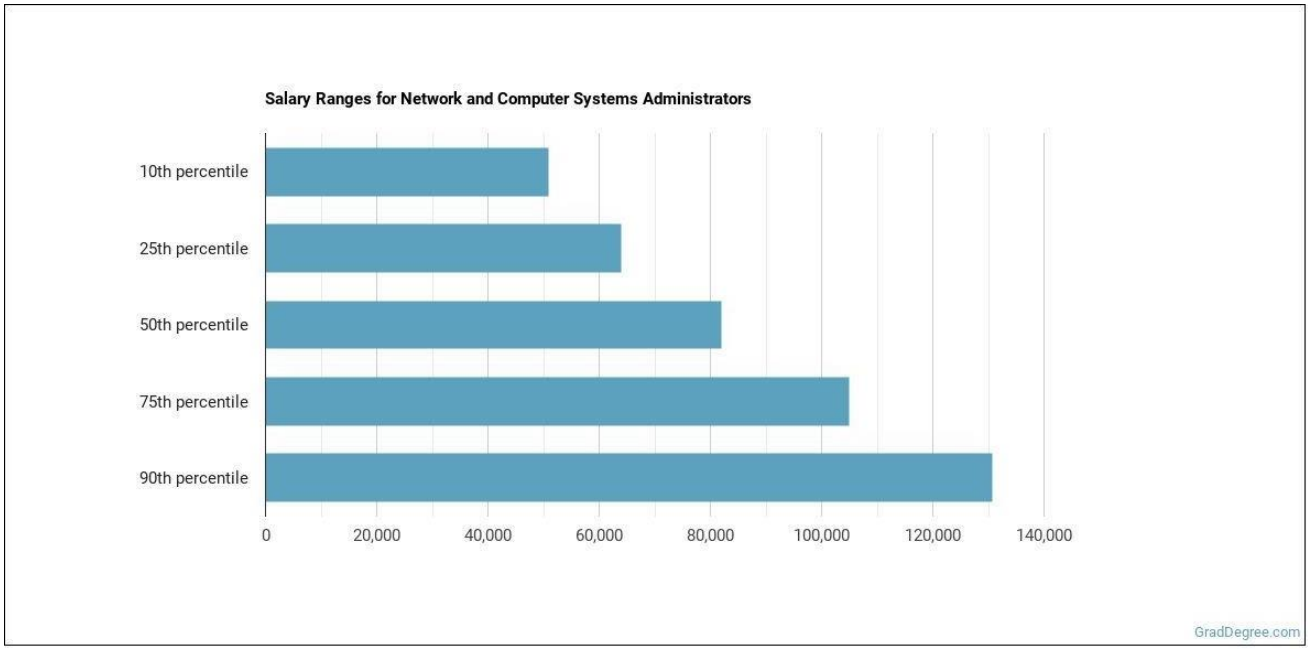


Fig:3.12 Architecture
of distribution of salary

The Architecture

There are various approaches to building an architecture for salary prediction, but one common method is to use a machine learning model such as a regression model or a neural network. Here is a high-level overview of the steps involved in building an architecture for salary prediction using a regression model:

Data collection: Collect data on past salaries and related features such as education level, years of experience, job title, company size, etc.

Data preprocessing: Clean the data and perform any necessary data transformations such as scaling or encoding categorical variables.

Feature selection: Select the most relevant features for the model using techniques such as correlation analysis or feature importance ranking.

Model selection: Choose a regression model such as linear regression, polynomial regression, or decision tree regression based on the nature of the data and the problem at hand.

Model training: Train the model on a subset of the data and use cross-validation techniques to evaluate its performance.

Model evaluation: Test the model on a holdout set of data and evaluate its performance using metrics such as mean absolute error, mean squared error, or R-squared.

Deployment: Deploy the model to a production environment where it can be used to predict salaries for new data.

It's worth noting that neural networks can also be used for salary prediction, but they typically require more data and computational resources than regression models. Additionally, deep learning architectures such as convolutional neural networks (CNNs) or recurrent neural networks (RNNs) may be used for specific cases with an extensive dataset or complex input data.

Configurations

When it comes to configuring a machine learning model for salary prediction, there are several hyperparameters that need to be tuned to ensure optimal performance. Here are a few important configurations to consider:

Model type: Choose the type of regression model that best suits your data and problem. For example, you could use a linear regression model, polynomial regression model, or decision tree regression model.

Learning rate: This is a hyperparameter that controls the step size taken in the gradient descent algorithm used to optimize the model. A too-small learning rate can result in slow convergence, while a too-large learning rate can cause the model to overshoot the optimal solution.

Regularization: Regularization is a technique used to prevent overfitting by adding a penalty term to the cost function. Two common types of regularization are L1 regularization (Lasso regression) and L2 regularization (Ridge regression).

Feature selection: Choose the most relevant features for the model based on domain knowledge, correlation analysis, or feature importance ranking.

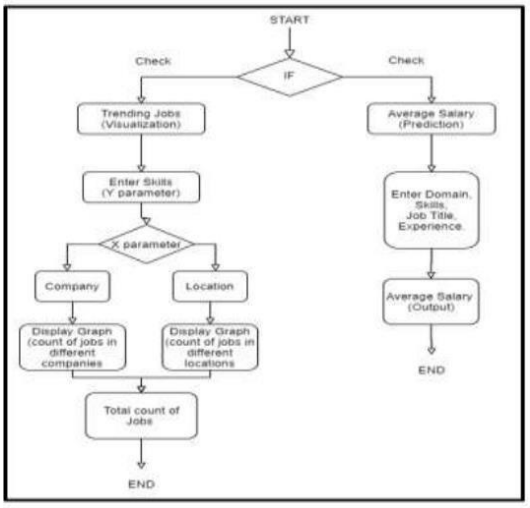


Fig:3.14 Methodology

Methodology:

The methodology for salary prediction typically involves the following steps:

Data collection: Collect data on past salaries and related features such as education level, years of experience, job title, company size, etc.

Data preprocessing: Clean the data and perform any necessary data transformations such as scaling or encoding categorical variables.

Feature selection: Select the most relevant features for the model using techniques such as correlation analysis or feature importance ranking.

Model selection: Choose a regression model such as linear regression, polynomial regression, or decision tree regression based on the nature of the data and the problem at hand.

Model training: Train the model on a subset of the data and use cross-validation techniques to evaluate its performance.

Hyperparameter tuning: Optimize the hyperparameters of the model to improve its performance on the holdout set of data.

Model evaluation: Test the model on the holdout set of data and evaluate its performance using metrics such as mean absolute error, mean squared error, or R-squared.

Deployment: Deploy the model to a production environment where it can be used to predict salaries for new data.

Additionally,important to consider ethical considerations when building a salary prediction model. For example, it's important to ensure that the model is not biased against certain groups.

Result

The result for salary prediction depends on the performance of the model and the evaluation metrics used. The goal of salary prediction is to accurately predict the salary of an individual based on their relevant features such as education level, years of experience, job title, etc. Therefore, the model's performance is typically measured using metrics such as mean absolute error (MAE), mean squared error (MSE), or R-squared.

A good model for salary prediction will have a low MAE and MSE, indicating that the predictions are close to the true values. The R-squared value should be close to 1, indicating that the model explains a high proportion of the variance in the data.

It's important to note that the result of salary prediction can be influenced by a variety of factors such as the quality of the data, the choice of model, and the hyperparameter tuning. Therefore, it's essential to carefully evaluate the model's performance using appropriate evaluation metrics and fine-tune the model as necessary to achieve the best possible result.

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.

$$TNR = tn / (tn + fp) = 1\text{-}FPR$$

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.

$$ACC=(tp+tn) / (tp+tn+fp+fn)$$

F1 score

It is the harmonic mean of precision and sensitivity sensitivity

$$F1=2tp/ (2tp+fp +fn)$$

1. DESIGN

The most important part of this system is the software design. The software design uses series of image processing techniques which are implemented in Android mobile platform which is supported minimum API 5 or android 2.0 (cupcake).

- Taking dataset as input
- Filtering all the possible ways
- Tesing and generating the predicted values

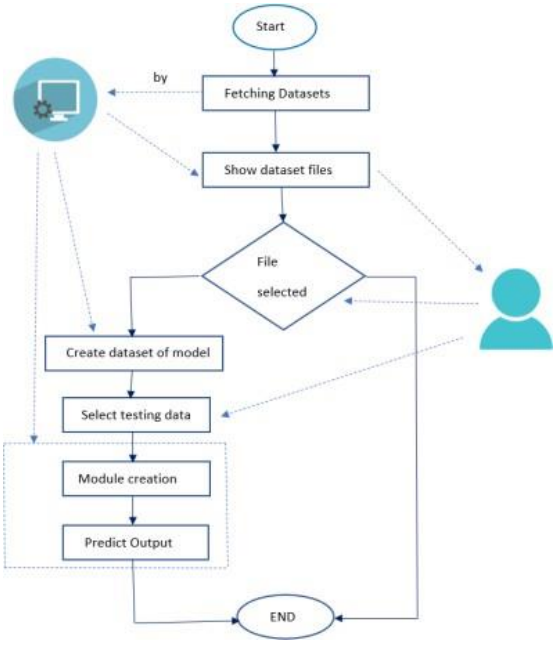


Fig: 4.1 Overview of the model

2. IMPLEMENTATION

CODE

Deploy the Web App on Cloud

- **import** pandas **as** pd

```
import numpy as np
import seaborn as sns
from sklearn.linear_model import LogisticRegression
import matplotlib.pyplot as plt
%matplotlib inline
```

- df = pd.read_csv('adult_data.csv')

- df.columns = ['age','workclass', 'fnlwgt', 'education', 'education_num', 'marital_status', 'occupation','relationship', 'race', 'sex', 'capital_gain', 'capital_loss', 'hours_per_week', 'native_country', 'salary']

- df.head()

- df.shape

- **def** handle_capital_gain(df):

```
df['capital_gain'] = np.where(df['capital_gain'] == 0, np.nan, df['capital_gain'])
df['capital_gain'] = np.log(df['capital_gain'])
df['capital_gain'] = df['capital_gain'].replace(np.nan, 0)
```

- handle_capital_gain(df)

- df.head()

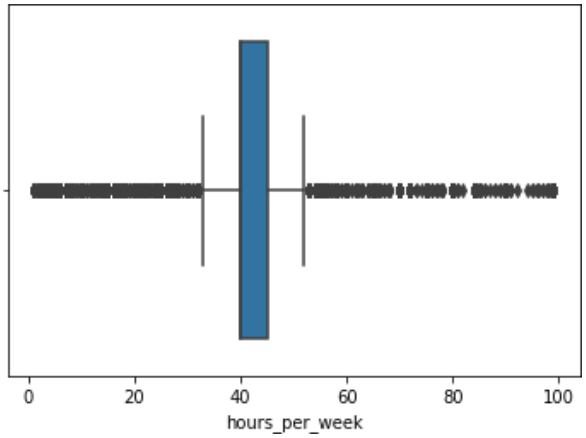
- df.describe()

- df.isnull().sum()

- df['salary'].unique()

Removing outliers from hours_per_week:

- sns.boxplot(df['hours_per_week'])

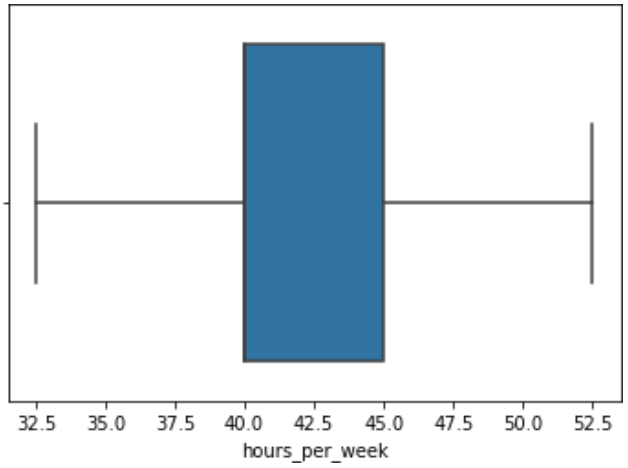


- **def** remove_outlier_hours_per_week(df):

```
IQR = df['hours_per_week'].quantile(0.75) - df['hours_per_week'].quantile(0.25)

lower_range = df['hours_per_week'].quantile(0.25) - (1.5 * IQR)
upper_range = df['hours_per_week'].quantile(0.75) + (1.5 * IQR)

df.loc[df['hours_per_week'] <= lower_range, 'hours_per_week'] = lower_range
df.loc[df['hours_per_week'] >= upper_range, 'hours_per_week'] = upper_range
```

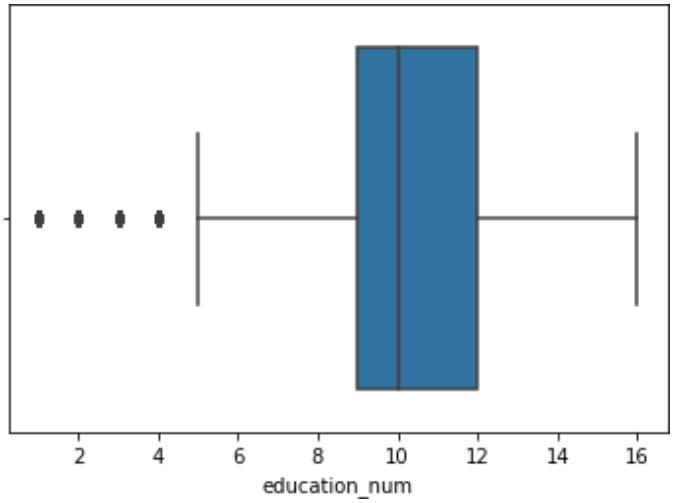
- remove_outlier_hours_per_week(df)

- sns.boxplot(df['hours_per_week'])



Removing outliers from education_num:

- sns.boxplot(df['education_num'])



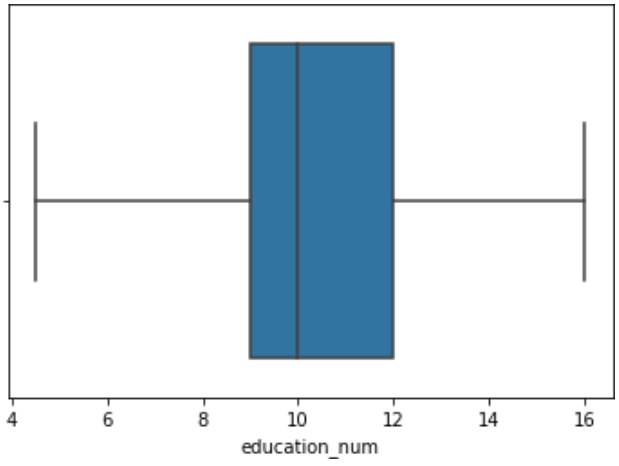
- **def** remove_outlier_education_num(df):
  IQR = df['education_num'].quantile(0.75) - df['education_num'].quantile(0.25)

lower_range = df['education_num'].quantile(0.25) - (1.5 * IQR)
upper_range = df['education_num'].quantile(0.75) + (1.5 * IQR)

df.loc[df['education_num'] <= lower_range, 'education_num'] = lower_range
df.loc[df['education_num'] >= upper_range, 'education_num'] = upper_range

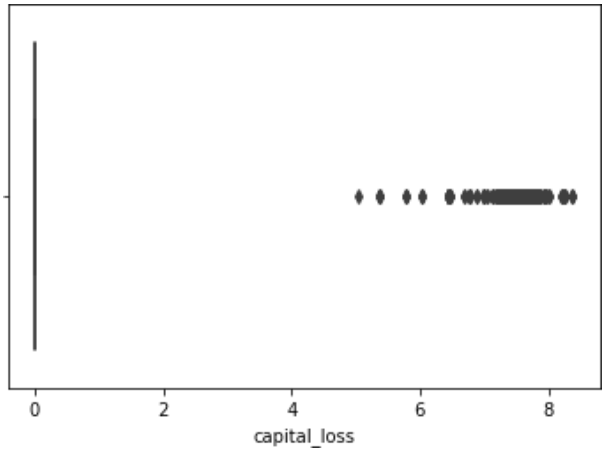
- remove_outlier_education_num(df)

- sns.boxplot(df['education_num'])

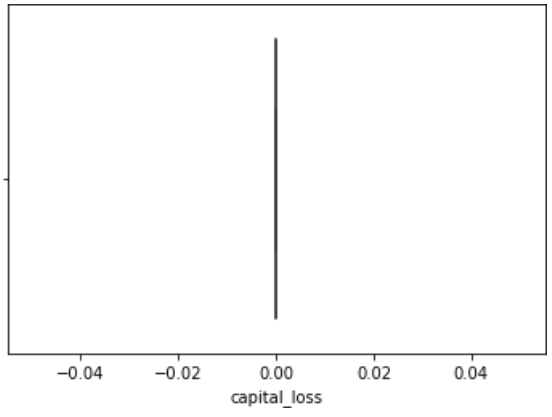


Removing outliers from captial_loss

```
def capital_loss_log(df):
        df['capital_loss'] = np.where(df['capital_loss'] == 0, np.nan,        df['capital_loss'])
    df['capital_loss'] = np.log(df['capital_loss'])
    df['capital_loss'] = df['capital_loss'].replace(np.nan, 0)
capital_loss_log(df)
sns.boxplot(df['capital_loss'])
```

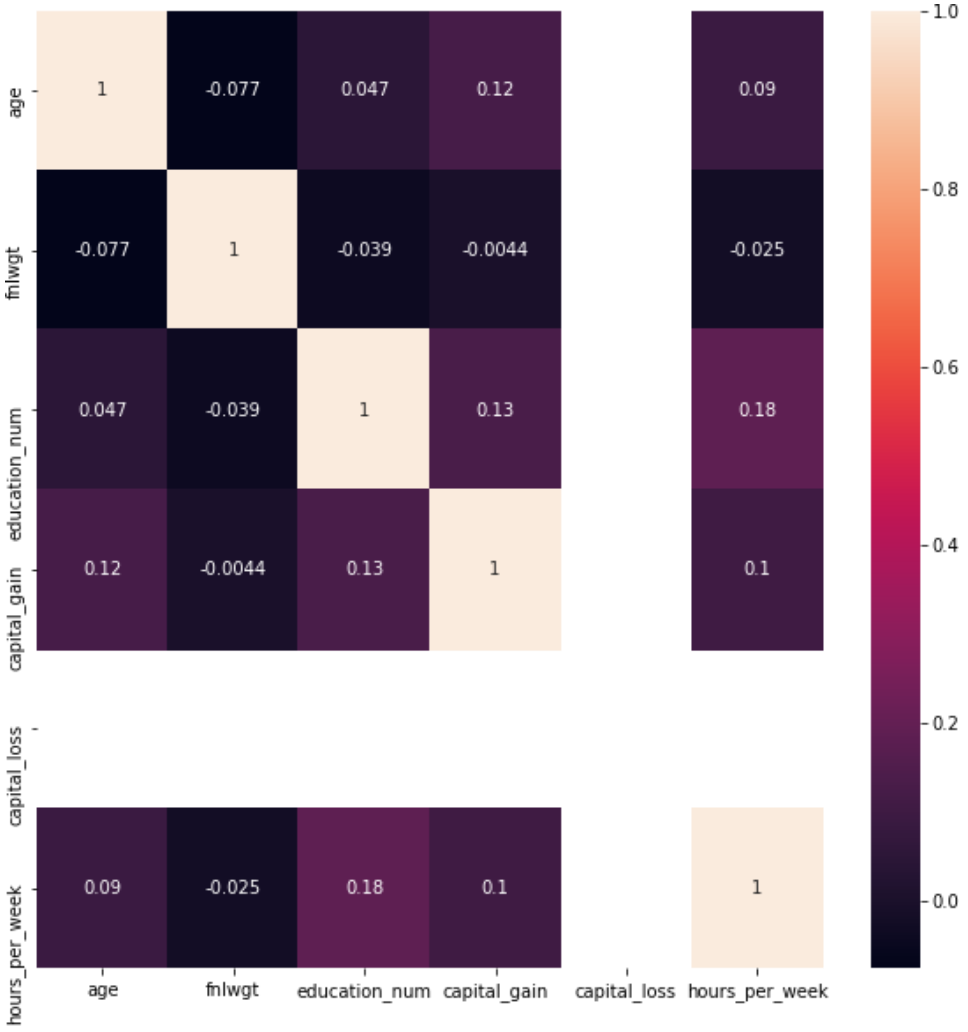



```
def remove_outlier_capital_loss(df):
    IQR = df['capital_loss'].quantile(0.75) - df['capital_loss'].quantile(0.25)

    lower_range = df['capital_loss'].quantile(0.25) - (1.5 * IQR)
    upper_range = df['capital_loss'].quantile(0.75) + (1.5 * IQR)

    df.loc[df['capital_loss'] <= lower_range, 'capital_loss'] = lower_range
    df.loc[df['capital_loss'] >= upper_range, 'capital_loss'] = upper_range
remove_outlier_capital_loss(df)
sns.boxplot(df['capital_loss'])
```



Plotted Heatmap

```
plt.figure(figsize=(10,  10))
corr = df.corr()
sns.heatmap(corr, annot=True)
```




```
df = df.drop('fnlwgt', axis=1)
df.head()
```

	age	workclass	education	education_num	marital_status	occupation	relationship	race	sex	capital_gain	capital_loss	hours_per_week	native_country	salary
0	39	State-gov	Bachelors	13.0	Never-married	Adm-clerical	Not-in-family	White	Male	7.684324	0.0	40.0	United-States	<=50K
1	50	Self-emp-not-inc	Bachelors	13.0	Married-civ-spouse	Exec-managerial	Husband	White	Male	0.000000	0.0	32.5	United-States	<=50K
2	38	Private	HS-grad	9.0	Divorced	Handlers-cleaners	Not-in-family	White	Male	0.000000	0.0	40.0	United-States	<=50K
3	53	Private	11th	7.0	Married-civ-spouse	Handlers-cleaners	Husband	Black	Male	0.000000	0.0	40.0	United-States	<=50K
4	28	Private	Bachelors	13.0	Married-civ-spouse	Prof-specialty	Wife	Black	Female	0.000000	0.0	40.0	Cuba	<=50K

Feature Engineering

```
def feature_engineering(df):

    ## convert the salary into 1 if salary is greater then 50K else 0
#     df['salary'] = df['salary'].replace(' >50K', '>50K')
#     df['salary'] = np.where(df['salary'] > '50K', 1, 0)

    ## convert the sex column into 0 and 1, if male then 1 else 0
    df['sex'] = np.where(df['sex'] == " Male", 1, 0)

    ## do the label encoding in race column (0: 'White',1: 'Black',2: 'Asian-Pac-Islander',3:'Amer-Indian-Eskimo',4:'Other')
    label_enco_race = {value: key for key, value in enumerate(df['race'].unique())}
    df['race'] = df['race'].map(label_enco_race)

    ## {0: ' Not-in-family',1: ' Husband'2: ' Wife',3: ' Own-child',4: ' Unmarried',5: ' Other-relative
    label_enco_relation = {value: key for key, value in enumerate(df['relationship'].unique())}
    df['relationship'] = df['relationship'].map(label_enco_relation)

    ## {0: ' Adm-clerical',1: ' Exec-managerial',2: ' Handlers-cleaners',3: ' Prof-specialty',4: ' Other-service',5: ' Sales', 6: ' Craft-repair',7: ' Transport-moving',8: ' Farming-fishing',9: ' Machine-op-inspct', 10: ' Tech-support', 11: ' ?',12: ' Protective-serv',13: ' Armed-Forces', 14: ' Priv-house-serv'}
    df['occupation'] = np.where(df['occupation'] == ' ?', 'Missing', df['occupation'])
    label_enco_occu = {value: key for key, value in enumerate(df['occupation'].unique())}
    ## Replacing ? value with 'Missing'
    df['occupation'] = df['occupation'].map(label_enco_occu)

    ## {0: ' Never-married',1: ' Married-civ-spouse',2: ' Divorced',3: ' Married-spouse-absent',4: ' Separated',5: ' Married-AF-spouse',6: ' Widowed'}
    label_enco_marital_status = {value: key for key, value in enumerate(df['marital_status'].unique())}
    df['marital_status'] = df['marital_status'].map(label_enco_marital_status)

label_enco_edu = {value: key for key, value in enumerate(df['education'].unique())}
    df['education'] = df['education'].map(label_enco_edu)

    ## {0: ' State-gov', 1: ' Self-emp-not-inc',2: ' Private',3: ' Federal-gov',4: ' Local-gov',5: ' ?',6: ' Self-emp-inc',7: ' Without-pay',8: ' Never-worked'}
    df['workclass'] = np.where(df['workclass'] == ' ?', 'Missing', df['workclass'])
    label_enco_workclass = {value: key for key, value in enumerate(df['workclass'].unique())}
    df['workclass'] = df['workclass'].map(label_enco_workclass)

    ## {' United-States': 0,' Cuba': 1,' Jamaica': 2,' India': 3,' ?': 4,' Mexico': 5,' South': 6,' Puerto-Rico': 7,' Honduras': 8,' England': 9,' Canada': 10,' Germany': 11,' Iran': 12,' Philippines': 13,' Italy': 14,' Poland': 15,' Columbia': 16,' Cambodia': 17,' Thailand': 18,' Ecuador': 19,' Laos': 20,' Taiwan': 21,' Haiti': 22,' Portugal': 23,' Dominican-Republic': 24,' El-Salvador': 25,' France': 26,' Guatemala': 27,' China': 28,' Japan': 29,' Yugoslavia': 30,' Peru': 31,' Outlying-US(Guam-USVI-etc)': 32,' Scotland': 33,' Trinadad&Tobago': 34,' Greece': 35,' Nicaragua': 36,' Vietnam': 37,' Hong': 38,' Ireland': 39,' Hungary': 40,' Holand-Netherlands': 41
    df['native_country'] = np.where(df['native_country'] == ' ?', 'Missing', df['native_country'])
    label_enco_workclass = {value: key for key, value in enumerate(df['native_country'].unique())}
    df['native_country'] = df['native_country'].map(label_enco_workclass)
    return df

df = feature_engineering(df)
```

After doing feature Engineering

df.head()

	age	workclass	education	education_num	marital_status	occupation	relationship	race	sex	capital_gain	capital_loss	hours_per_week	native_country	salary
0	39	0	0	13.0	0	0	0	0	1	7.684324	0.0	40.0	0	<=50K
1	50	1	0	13.0	1	1	1	0	1	0.000000	0.0	32.5	0	<=50K
2	38	2	1	9.0	2	2	0	0	1	0.000000	0.0	40.0	0	<=50K
3	53	2	2	7.0	1	2	1	1	1	0.000000	0.0	40.0	0	<=50K
4	28	2	0	13.0	1	3	2	1	0	0.000000	0.0	40.0	1	<=50K

Scaling

```
from sklearn.preprocessing import StandardScaler

sc = StandardScaler()

X = df[['age', 'workclass', 'education', 'education_num', 'marital_status', 'occupation', 'relationship','race', 'sex', 'capital_gain', 'capital_loss', 'hours_per_week', 'native_country']]

y = df['salary']

y.value_counts()

X = sc.fit_transform(X)

X
array([[ 0.03067056, -1.88460023, -0.99158435, ...,  0.        ,
        -0.194354  , -0.25574647],
       [ 0.83710898, -1.0687461 , -0.99158435, ...,  0.        ,
        -1.40659071, -0.25574647],
       [-0.04264203, -0.25289198, -0.70202542, ...,  0.        ,
        -0.194354  , -0.25574647],…,
       [ 0.98373415,  3.01052452, -0.70202542, ...,  0.        ,
        -0.194354  , -0.25574647]])
```

train_test_split

```
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

print("Train data shape: {}".format(X_train.shape))
print("Test data shape: {}".format(X_test.shape))
```

➔ Train data shape: (26048, 13)
Test data shape: (6513, 13)

```
from sklearn.linear_model import LogisticRegression


lg_model = LogisticRegression()

lg_model.fit(X_train, y_train)



y_pred = lg_model.predict(X_test)

result = {
    'Actual': y_test,
    'Predicted': y_pred
}

pd.DataFrame(result)
```

	Actual	Predicted
14160	<=50K	<=50K
27048	<=50K	<=50K
28868	>50K	>50K
5667	<=50K	<=50K
7827	<=50K	>50K
...	...	...
1338	<=50K	>50K
24534	>50K	<=50K
18080	>50K	>50K
10354	<=50K	<=50K
24639	<=50K	<=50K

5513 rows × 2 columns

```
from sklearn.metrics import confusion_matrix, accuracy_score, classification_report

print("Accuracy Score: {}".format(accuracy_score(y_test, y_pred)), "\n")
print("Confusion Matrix:\n {}".format(confusion_matrix(y_test, y_pred)), "\n")
print("Classification Report:\n {}".format(classification_report(y_test, y_pred)), "\n")
```

```
Accuracy Score: 0.8182097343773991

Confusion Matrix:
 [[4626  316]
 [ 868  703]]

Classification Report:
               precision    recall  f1-score   support

        <=50K       0.84      0.94      0.89      4942
         >50K       0.69      0.45      0.54      1571

     accuracy                           0.82      6513
    macro avg       0.77      0.69      0.71      6513
 weighted avg       0.81      0.82      0.80      6513
```

index.html

```
<html>
   <body>
      <br>
      <h1 align="center">Salary Prediction Machine Learning Project</h1>
      <hr>
      <br>

      <fieldset align="center">
         <form action="/predict" method="post" enctype="multipart/form-data">
            <br>
            <label class="control-label" align="center">Select CSV File Contatining Sample Data</label>
            <br>
            <br>
            <hr>
            <br>
            <input type="file" name="data_file" class="btn btn-block"/>
            <button type="submit" class="btn btn-primary btn-block btn-large">Predict</button>
            <br><br>
         </form>
      </fieldset>
      <br><br>

      <h3 align="center">{{ prediction_text }}</h3>

   </body>
</html>
```

Connections:

```
from flask import Flask, make_response, request, render_template
import io
from io import StringIO
import pandas as pd
import numpy as np
import pickle
from sklearn.preprocessing import StandardScaler

app = Flask(__name__)

def feature_engineering(df):
   df.columns = ['age','workclass', 'fnlwgt', 'education', 'education_num', 'marital_status', 'occupation',
         'relationship', 'race', 'sex', 'capital_gain', 'capital_loss', 'hours_per_week', 'native_country']

   df = df.drop('fnlwgt', axis=1)

   ## convert the sex column into 0 and 1, if male then 1 else 0
   df['sex'] = np.where(df['sex'] == " Male", 1, 0)
```

```
    ## do the label encoding in race column (0: 'White',1: 'Black',2: 'Asian-Pac-Islander',3:'Amer-Indian-Eskimo',4:'Other')
    label_enco_race = {value: key for key, value in enumerate(df['race'].unique())}
    df['race'] = df['race'].map(label_enco_race)

    ## {0: ' Not-in-family',1: ' Husband'2: ' Wife',3: ' Own-child',4: ' Unmarried',5: ' Other-relative
    label_enco_relation = {value: key for key, value in enumerate(df['relationship'].unique())}
    df['relationship'] = df['relationship'].map(label_enco_relation)

    ## {0: ' Adm-clerical',1: ' Exec-managerial',2: ' Handlers-cleaners',3: ' Prof-specialty',4: ' Other-service',5: ' Sales', 6: ' Craft-repair',7: ' Transport-moving',8: ' Farming-fishing',9: ' Machine-op-inspct', 10: ' Tech-support', 11: ' ?',12: ' Protective-serv',13: ' Armed-Forces', 14: ' Priv-house-serv'}
    df['occupation'] = np.where(df['occupation'] == ' ?', 'Missing', df['occupation'])
    label_enco_occu = {value: key for key, value in enumerate(df['occupation'].unique())}
    ## Replacing ? value with 'Missing'
    df['occupation'] = df['occupation'].map(label_enco_occu)

    ## {0: ' Never-married',1: ' Married-civ-spouse',2: ' Divorced',3: ' Married-spouse-absent',4: ' Separated',5: ' Married-AF-spouse',6: ' Widowed'}
    label_enco_marital_status = {value: key for key, value in enumerate(df['marital_status'].unique())}
    df['marital_status'] = df['marital_status'].map(label_enco_marital_status)
label_enco_edu = {value: key for key, value in enumerate(df['education'].unique())}
    df['education'] = df['education'].map(label_enco_edu)

    ## {0: ' State-gov', 1: ' Self-emp-not-inc',2: ' Private',3: ' Federal-gov',4: ' Local-gov',5: ' ?',6: ' Self-emp-inc',7: ' Without-pay',8: ' Never-worked'}
    df['workclass'] = np.where(df['workclass'] == ' ?', 'Missing', df['workclass'])
    label_enco_workclass = {value: key for key, value in enumerate(df['workclass'].unique())}
    df['workclass'] = df['workclass'].map(label_enco_workclass)

    ## {' United-States': 0,' Cuba': 1,' Jamaica': 2,' India': 3,' ?': 4,' Mexico': 5,' South': 6,' Puerto-Rico': 7,' Honduras': 8,' England': 9,' Canada': 10,' Germany': 11,' Iran': 12,' Philippines': 13,' Italy': 14,' Poland': 15,' Columbia': 16,' Cambodia': 17,' Thailand': 18,' Ecuador': 19,' Laos': 20,' Taiwan': 21,' Haiti': 22,' Portugal': 23,' Dominican-Republic': 24,' El-Salvador': 25,' France': 26,' Guatemala': 27,' China': 28,' Japan': 29,' Yugoslavia': 30,' Peru': 31,' Outlying-US(Guam-USVI-etc)': 32,' Scotland': 33,' Trinadad&Tobago': 34,' Greece': 35,' Nicaragua': 36,' Vietnam': 37,' Hong': 38,' Ireland': 39,' Hungary': 40,' Holand-Netherlands': 41
    df['native_country'] = np.where(df['native_country'] == ' ?', 'Missing', df['native_country'])
    label_enco_workclass = {value: key for key, value in enumerate(df['native_country'].unique())}
    df['native_country'] = df['native_country'].map(label_enco_workclass)
    return df

def scalar(df):
    sc = StandardScaler()
    X = df[['age', 'workclass', 'education', 'education_num', 'marital_status', 'occupation', 'relationship', 'race', 'sex', 'capital_gain', 'capital_loss', 'hours_per_week', 'native_country']]
    X = sc.fit_transform(X)
    return (X)


@app.route('/', methods=['GET'])
def Home():
    return render_template('index.html')
@app.route('/predict', methods=["POST"])
def predict():
    f = request.files['data_file']
    if not f:
        return render_template('index.html', prediction_text="No file selected")
```

```
    stream = io.StringIO(f.stream.read().decode("UTF8"), newline=None)
    result = stream.read()#.replace("=", ",")
    df = pd.read_csv(StringIO(result))

    #Feature Engineering
    df = feature_engineering(df)

    X = scalar(df)

    # load the model from disk
    loaded_model = pickle.load(open("lg_model.pkl", 'rb'))

    result = loaded_model.predict(X)

    return render_template('index.html', prediction_text="Predicted Salary is/are: {}".format(result))

if __name__== "__main__":
    app.run(debug=False,port=9000)
```

Backend:

```
from flask import Flask, make_response, request, render_template
import io
from io import StringIO
import pandas as pd
import numpy as np
import pickle
from sklearn.preprocessing import StandardScaler

app = Flask(__name__)

def feature_engineering(df):
    df.columns = ['age','workclass', 'fnlwgt', 'education', 'education_num', 'marital_status', 'occupation',
            'relationship', 'race', 'sex', 'capital_gain', 'capital_loss', 'hours_per_week', 'native_country']

    df = df.drop('fnlwgt', axis=1)

    ## convert the sex column into 0 and 1, if male then 1 else 0
    df['sex'] = np.where(df['sex'] == " Male", 1, 0)

    ## do the label encoding in race column (0: 'White',1: 'Black',2: 'Asian-Pac-Islander',3:'Amer-Indian-
Eskimo',4:'Other')
    label_enco_race = {value: key for key, value in enumerate(df['race'].unique())}
    df['race'] = df['race'].map(label_enco_race)

    ## {0: ' Not-in-family',1: ' Husband'2: ' Wife',3: ' Own-child',4: ' Unmarried',5: ' Other-relative
    label_enco_relation = {value: key for key, value in enumerate(df['relationship'].unique())}
    df['relationship'] = df['relationship'].map(label_enco_relation)
```

```
    ## {0: ' Adm-clerical',1: ' Exec-managerial',2: ' Handlers-cleaners',3: ' Prof-specialty',4: ' Other-service',5: ' Sales', 6: ' Craft-repair',7: ' Transport-moving',8: ' Farming-fishing',9: ' Machine-op-inspct', 10: ' Tech-support', 11: ' ?',12: ' Protective-serv',13: ' Armed-Forces', 14: ' Priv-house-serv'}
    df['occupation'] = np.where(df['occupation'] == ' ?', 'Missing', df['occupation'])
    label_enco_occu = {value: key for key, value in enumerate(df['occupation'].unique())}
    ## Replacing ? value with 'Missing'
    df['occupation'] = df['occupation'].map(label_enco_occu)

    ## {0: ' Never-married',1: ' Married-civ-spouse',2: ' Divorced',3: ' Married-spouse-absent',4: ' Separated',5: ' Married-AF-spouse',6: ' Widowed'}
    label_enco_marital_status = {value: key for key, value in enumerate(df['marital_status'].unique())}
    df['marital_status'] = df['marital_status'].map(label_enco_marital_status)

    label_enco_edu = {value: key for key, value in enumerate(df['education'].unique())}
    df['education'] = df['education'].map(label_enco_edu)

    ## {0: ' State-gov', 1: ' Self-emp-not-inc',2: ' Private',3: ' Federal-gov',4: ' Local-gov',5: ' ?',6: ' Self-emp-inc',7: ' Without-pay',8: ' Never-worked'}
    df['workclass'] = np.where(df['workclass'] == ' ?', 'Missing', df['workclass'])
    label_enco_workclass = {value: key for key, value in enumerate(df['workclass'].unique())}
    df['workclass'] = df['workclass'].map(label_enco_workclass)

    ## {' United-States': 0,' Cuba': 1,' Jamaica': 2,' India': 3,' ?': 4,' Mexico': 5,' South': 6,' Puerto-Rico': 7,' Honduras': 8,' England': 9,' Canada': 10,' Germany': 11,' Iran': 12,' Philippines': 13,' Italy': 14,' Poland': 15,' Columbia': 16,' Cambodia': 17,' Thailand': 18,' Ecuador': 19,' Laos': 20,' Taiwan': 21,' Haiti': 22,' Portugal': 23,' Dominican-Republic': 24,' El-Salvador': 25,' France': 26,' Guatemala': 27,' China': 28,' Japan': 29,' Yugoslavia': 30,' Peru': 31,' Outlying-US(Guam-USVI-etc)': 32,' Scotland': 33,' Trinadad&Tobago': 34,' Greece': 35,' Nicaragua': 36,' Vietnam': 37,' Hong': 38,' Ireland': 39,' Hungary': 40,' Holand-Netherlands': 41
    df['native_country'] = np.where(df['native_country'] == ' ?', 'Missing', df['native_country'])
    label_enco_workclass = {value: key for key, value in enumerate(df['native_country'].unique())}
    df['native_country'] = df['native_country'].map(label_enco_workclass)
    return df

def scalar(df):
    sc = StandardScaler()
    X = df[['age', 'workclass', 'education', 'education_num', 'marital_status', 'occupation', 'relationship', 'race', 'sex', 'capital_gain', 'capital_loss', 'hours_per_week', 'native_country']]
    X = sc.fit_transform(X)
    return (X)


@app.route('/', methods=['GET'])
def Home():
    return render_template('index.html')


@app.route('/predict', methods=["POST"])
def predict():
    f = request.files['data_file']
```

```
    if not f:
        return render_template('index.html', prediction_text="No file selected")

    stream = io.StringIO(f.stream.read().decode("UTF8"), newline=None)
    result = stream.read()#.replace("=", ",")
    df = pd.read_csv(StringIO(result))

    #Feature Engineering
    df = feature_engineering(df)

    X = scalar(df)

    # load the model from disk
    loaded_model = pickle.load(open("lg_model.pkl", 'rb'))

    result = loaded_model.predict(X)

    return render_template('index.html', prediction_text="Predicted Salary is/are: {}".format(result))

if__name__== "_main_":
    app.run(debug=False,port=9000)
```

3. RESULT ANALYSIS

The system evaluates the performance of four deep neural networks. Finding the best object detection method is the main objective of this study. i.e., a best object detection algorithm can accurately find out the license plate region so that the license plate number can be extracted successfully. The system automatically detects the license plate using four deep learning algorithms. The input image to the system is a car image and detects the number plate using the deep neural networks CNN, VGG16, VGG19, and YOLOV3 separately. Finally, evaluate the performance of the four deep neural networks in terms of accuracy to find out the best algorithm for license plate detection.

The VGG16 achieves the highest performance while evaluating the performance of each algorithm using the test set. VGG19 exhibits the least accuracy as well. The VGG16 is the final model employed for number plate recognition. CNN shows an accuracy of 77%, VGG16 shows an accuracy of 89%, VGG19 shows an accuracy of 49%, and YOLOV3 shows an accuracy of 78%. Among these, VGG16 exhibits precise license plate recognition.

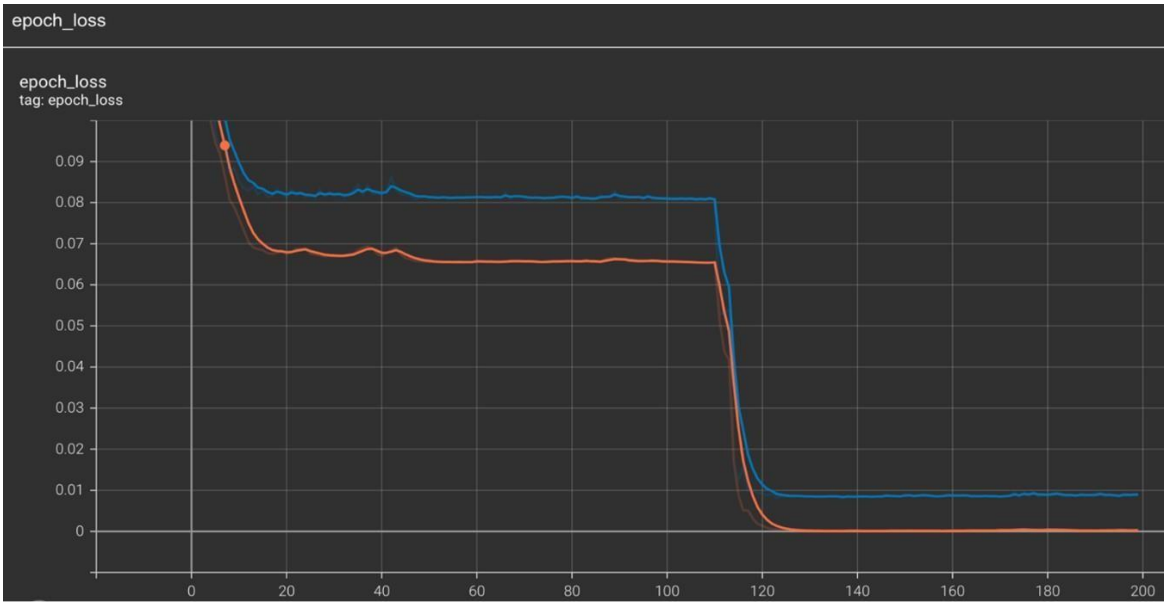


Fig: 6.2 Graph plot of the epochs

To evaluate the proposed system's effectiveness, three different testing cases using various car video sequences from several scenes at different conditions are used. The license plate detection accuracies and the character recognition accuracies of this evaluation were recorded and listed in Table 2.

Table 2. License plate detection and character recognition accuracies.

Video Sequences	**Detection accuracy (%)**	**Recognition accuracy (%)**
1	98.4	98.9
2	96.23	98.5
3	97.4	97.9

For the rating Criteria, the proposed system is evaluated by calculating the accuracy of the detection and the recognition which is defined as the number of correctly detected license plates divided by the number of correctly detected plus the number of incorrectly detected.

$$\text{Accuracy} = \frac{\text{true positive}}{\text{true positive} + \text{false positive}}$$

The results listed in Table 2 show that the proposed method gives high accuracy for the three sequences and confirms that it can effectively detect the license plate and recognize its characters in different situations. Detection accuracy goes up to 98.4% and recognition accuracy goes up to 98.9% in some cases.

4. SCREENSHOTS

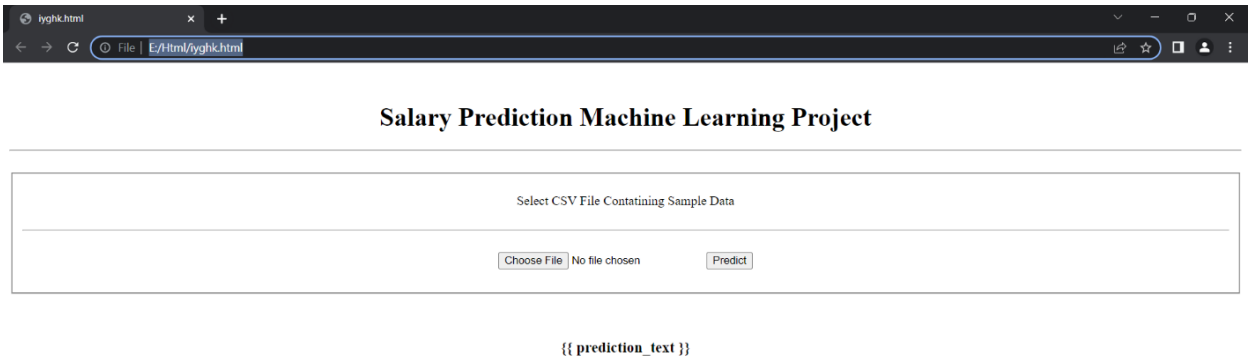


Fig: 7.1 Home Screen

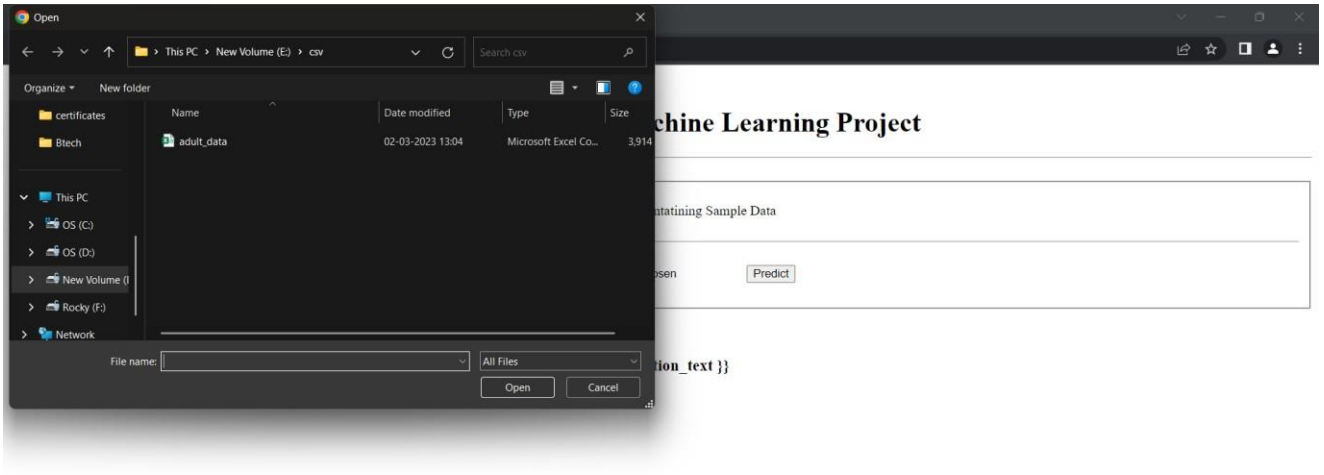


Fig: 7.2 Selection of dataset to the system

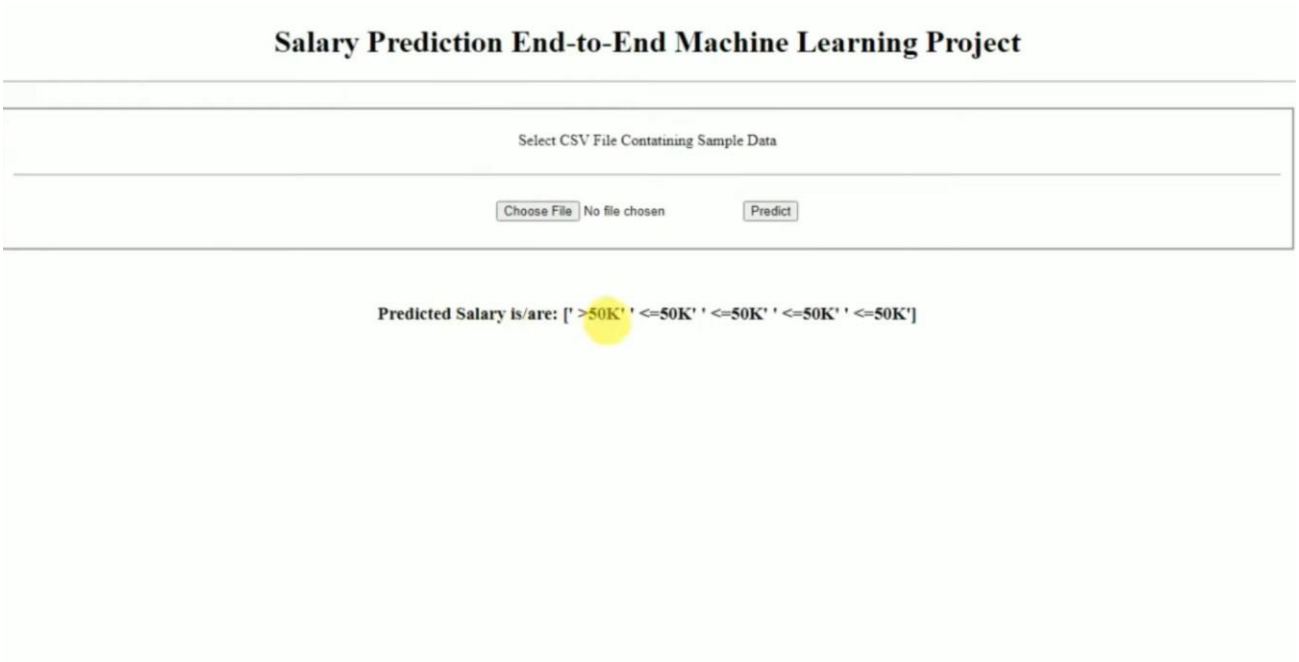


Fig: 7.3 Resultant Output Display Screen

5. CONCLUSION & FUTURE SCOPE

In conclusion, salary prediction is a useful application of machine learning that can help individuals and organizations make informed decisions about compensation. By collecting data on past salaries and relevant features such as education level, years of experience, and job title, it's possible to build a model that can accurately predict the salary of an individual.

To build an effective salary prediction model, it's important to carefully collect and preprocess the data, select the most relevant features, and train and evaluate the model using appropriate techniques. Additionally, it's essential to consider ethical considerations such as fairness and bias in the model design and evaluation.

Overall, salary prediction is a valuable tool for both individuals and organizations to make informed decisions about compensation and ensure that pay is fair and equitable.

6. REFERENCES

[1] A. K. Lakshmi and A.Parkavil Predicting the course knowledge level of students using data mining techniques, IEEE International Conference on Smart Technologies and Management for Computing, Communication,Controls, Energy and Materials ,2017

[2] A.W. Husain, N.A Rashid and A.M. Shahiri, A Review on Predicting Students Performance using Data mining Techniques Procedia Computer Science 72:414-422, 2015.

[3] Richard A Huebner, A Survey of Educational Data-Mining Research, Academic and Business Research Institute,2013

[4] Pornthep Khongchai, Pokpong Songmuang, improving students' motivation to study using salary prediction system, 13th International Joint Conference on Computer Science and Software Engineering, 2016

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

[6] John Jerrim, Do college students make better predictions of their future income than young adults in the labor force ?, Education Economics, Taylor & Francis Journals, vol. 23(2), pages 162- 179, 2015

[7] Karlar Hamlen and William A. Hamlen, Faculty Salary as a predictor of student outgoing salaries from MBA programs, Journal of Education for Business, 2016

[8] Rajveer Singh, A Regression Study of Salary Determinants in Indian Job Markets for Entry Level Engineering Graduates, Masters Dissertation. Dublin Institute of Technology, 2016.

[9] C.C. Hung, E.-P. Lim, On Aggregating Salaries of Occupations from Job Post and Review Data”, IEEE Access, Volume 9, 2021

[10]Dr. Kamaljit I. Lakhtaria, Bhaskar Patel and S.G. Prajapati, Efficient classification of data using decision Tree, Bonfring international journal of data mining 2 (1), 06-12,2012.

Salary Prediction

Duddukuri Ajay Babu
Student
Department of Computer Science and Engineering
Narasaraopet, India
ajaybabududdukuri@gmail.com

Peddireddy Venkata Rohith
Student
Department of Computer Science and Engineering
Narasaraopet, India
rohithpeddireddy24@gmail.com

Bapathu Sandeep Reddy
Student
Department of Computer Science and Engineering
Narasaraopet, India
Sandeepreddybaapathu052@gmail.com

M.Satyam Reddy
Assoc Professor
Department of Computer Science and Engineering
Narasaraopet, India
satyamreddy.nrtnec@gmail.com

***Abstract*— The Salary Prediction project aims to predict the salaries of employees based on various factors such as job title, years of experience, location, education level, and industry. The project uses machine learning algorithms to analyze a dataset of historical salary and employee information to develop a model that can accurately predict future salaries.The dataset used in this project contains information such as job title, years of experience, location, education level, and industry for thousands of employees. The dataset is cleaned and preprocessed to remove any missing or irrelevant data.Various machine learning algorithms such as linear regression, decision tree, and random forest are used to develop the salary prediction model. The performance of each algorithm is evaluated using metrics such as Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and R-squared.The final model is then deployed to make salary predictions for new employees based on their job title, years of experience, location, education level, and industry. The model can be used by companies to help with salary negotiations, employee retention, and workforce planning. The salary prediction project has the potential to help job seekers, employers, and policymakers make informed decisions about salaries and compensation packages. By providing accurate predictions of salaries, this model could help job seekers negotiate better salaries, assist employers in setting appropriate salaries for their employees, and help policymakers make informed decisions about minimum wages and labor policies.Keywords*—**MachineLearning , LinearRegression, Random Forest, Logistic regression, Flask**

I. INTRODUCTION

A salary prediction project is a data science project that aims to build a model to predict the salary of a job position based on various factors such as education, years of experience, job title, industry, location, and other relevant factors.The project involves collecting and analyzing data from various sources such as job listings, industry reports, and online surveys. Once the data is collected, it is cleaned, preprocessed, and transformed into a format that can be used for modeling.The next step is to choose an appropriate machine learning algorithm to build the model. This may involve trying out different algorithms and selecting the one that gives the best performance. The model is trained on a subset of the data and evaluated on a separate set to ensure that it can accurately predict salaries for new job positions.Once the model is developed and validated, it can be used to predict salaries for new job positions. This can be useful for job seekers who want to negotiate their salaries or for companies that want to ensure they are offering competitive salaries to their employees.

Overall, a salary prediction project is an exciting and challenging data science project that can provide valuable insights into the job market and help people make informed decisions about their careers.and random forests to train the model.Once we have developed the model, we will evaluate its performance by testing it on a separate set of data. This will enable us to determine the accuracy of the model and identify any areas for improvement.Overall, a salary prediction project can be a valuable tool for employers and employees alike. By accurately predicting salaries, employers can ensure they are offering fair compensation and employees can negotiate their salaries more effectively

II LITERATURE REVIEW

There have been several studies on salary prediction in recent years, focusing on different aspects of the problem. Some studies have looked at the impact of various factors on salary, such as education level, job experience, and industry sector. Other studies have explored the use of different machine learning algorithms for salary prediction.Predicting Salary from Job Posting Text" by Ahmad A. Tafti and

Arman Cohan (2018) - This paper explores the use of natural language processing (NLP) techniques to predict salaries from job postings. The authors use various machine learning algorithms,such as linear regression and support vector machines, and compare their performance.Predicting Salaries with Machine Learning" by Arjun Adhikari (2019) - This article provides a step-by-step guide on how to build a salary prediction model using Python and scikit-learn. The author explains various preprocessing techniques, feature engineering, and model selection.A Deep Learning Model for Predicting Job Salaries" by Kai Shu et al. (2018) - This paper proposes a deep learning model for predicting salaries based on job descriptions. The authors use a combination of convolutional neural networks (CNNs) and recurrent neural networks (RNNs) to capture both the semantic and syntactic information in the text.Predicting Employee Salary Using Machine Learning Techniques" by Ayesha Waheed et al. (2019) - This paper presents a study on predicting employee salaries using various machine learning algorithms, such as decision trees, random forests, and gradient boosting. The authors evaluate the performance of these algorithms on a real-world dataset.Predicting Salaries for Job Postings Using Machine Learning" by Brian Dew et al. (2018) - This article discusses a project in which the authors built a salary prediction model using job postings and salary data from Glassdoor. The authors use various NLP techniques, such as word embeddings and topic modeling, to extract features from the job postings.These papers and articles demonstrate the wide range of techniques and approaches that can be used for salary prediction, from traditional machine learning algorithms to deep learning models and NLP techniques

III.EXISTING SYSTEM

The existing system,is used to predict the salary, based on the machine learning algorithms and data mining algorithm were widely used. Salary prediction is a popular application of machine learning in the automotive industry. In this project, the goal is to predict the salary based on various factors , such as work, age ,education martial status ,occupation, experience and other factors.The major drawback of this existing system is they need more attributes in order to predict the salary . More comparison techniques must be used to get the result more efficiently.

Some of the challenges in car price prediction include dealing with outliers, handling missing data, and selecting the most relevant features. To address these challenges, various techniques such as data imputation, feature selection, and regularization are used in the existing systems of salary prediction projects in machine learning to get more accuracy .

IV PROPOSED SYSTEM

The proposed system is based on the different factors, and features also with the help of experts knowledge the salary prediction has been done accurately to predict best salary. These are the features useful to develop a efficient and effective model which predicts the salary according to the user inputs In this paper, we applied different models and techniques and methods in order to achieve higher accuracy of the precision of the salary prediction.

V DATASET AND DATA VISUALIZATION

We have used the dataset available in Kaggle which consists different types of employment. The dataset is split into training and testing datasets . The training data is 80% of the total dataset, validation data is 10% of dataset and testing data is other 10% of the dataset.

Data visualization is an important tool for data analysis and communication that enables us to visually represent complex datasets and identify patterns and relationships within the data. Visualizations can take various forms, such as scatter plots, line charts, bar charts, histograms, heat maps, box plots, tree maps, and many more.

It is The choice of visualization technique depends on type of data and the specific insights being communicated. The goal of data visualization is to communicate complex information clearly and effectively, making it easy for the audience to understand the key insights and trends in the data.

Data visualization can take many forms, including, graphs, maps, charts and other visual aids. It is often used in fields such as business, science, engineering, medicine, and social sciences to present data in a way that is accessible and easy to understand.

age	workclass	fnlwgt	education	education-num	marital-status	occupation	relationship	race	sex
39	State-gov	77516	Bachelors	13	Never-married	Adm-clerical	Not-in-family	White	Male
50	Self-emp-not-inc	83311	Bachelors	13	Married-civ-spouse	Exec-managerial	Husband	White	Male
38	Private	215646	HS-grad	9	Divorced	Handlers-cleaners	Not-in-family	White	Male
53	Private	234721	11th	7	Married-civ-spouse	Handlers-cleaners	Husband	Black	Male
28	Private	338409	Bachelors	13	Married-civ-spouse	Prof-specialty	Wife	Black	Female

Fig.1 : Data Set

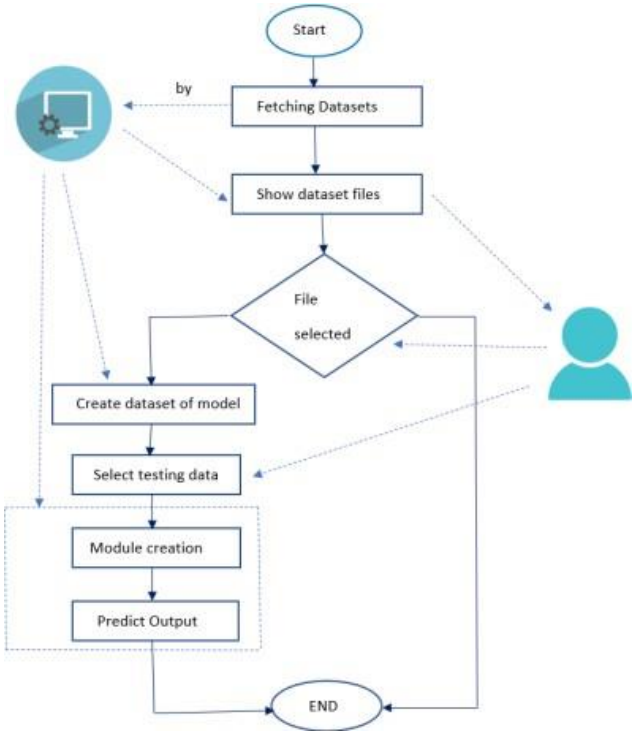


Fig.2: System architecture diagram

VI PREPROCESSING

Data preprocessing is an important step in the data analysis process, where raw data is transformed into a format that is suitable for analysis. Here are some common techniques used in data preprocessing:

Data Cleaning: It is the process of removing or correcting any errors or inconsistencies in the data.This can include removing duplicates, correcting misspelled values, or imputing missing data.

Data Transformation: It is the process of converting data from one format to another, such as converting categorical data to numerical data. This can also include scaling data to a common range or normalizing data have a mean of zero and standard deviation of one.

Data Reduction: Data reduction involves reducing the amount of data to be analyzed. This can include identifying and removing irrelevant features or reducing the resolution of data by aggregating it into larger groups.

Overall, data preprocessing is a major step in data analysis process as the data is consistent, accurate and in a format that can be easily can be analyzed.

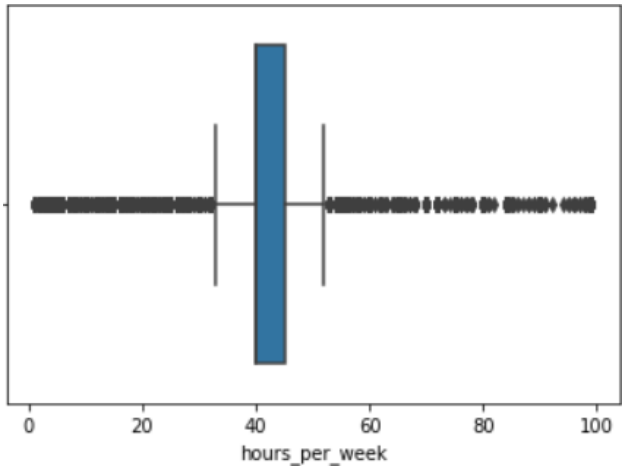


Fig.3: Removing outliers

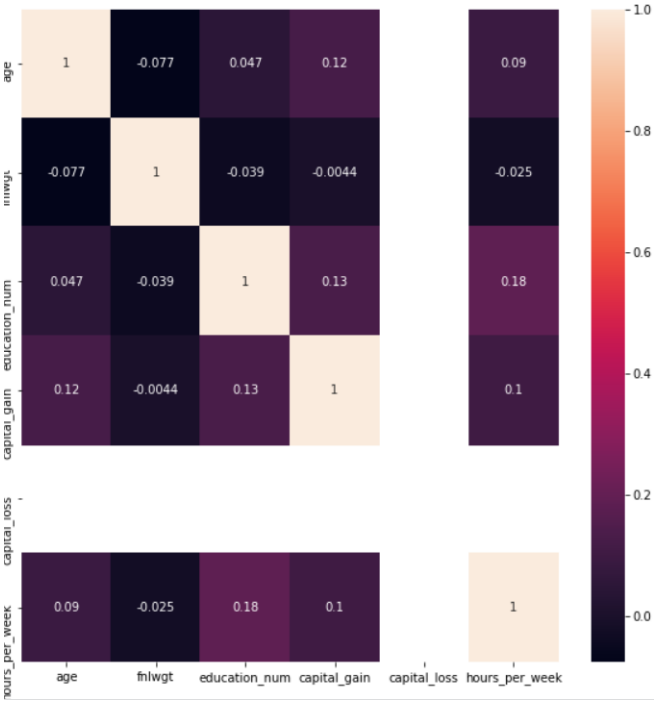


Fig .4: Co-relation

VII METHODOLOGY AND IMPLEMENTAION

System analysis of salary prediction involves understanding the various components and processes involved in the system, and how they work together to predict the salary. Here are some key components of a salary prediction system:

Data Collection: The system needs to collect data from various sources, such as historical sales data, market trends, and employee specifications. This data is then used to train the prediction model.

Data Preprocessing: The raw data collected from various sources may not be in a format that is suitable for analysis. Therefore, data preprocessing techniques such as data mining ,data cleaning, integration and transformation need to be applied to ensure that the data is accurate and consistent.

Feature Selection: The system needs to identify which features of the car are relevant for predicting its price. This can be done using statistical techniques or machine learning algorithms.

Prediction Model: The prediction model is trained using the preprocessed data and selected features. In the Various algorithms such as regression models, decision trees, can be used to build the prediction model.

Model Evaluation: The prediction model needs to be evaluated to assess its accuracy and effectiveness. This can be done using metrics such as mean squared error, root mean squared error, or R-squared.

Deployment: The prediction model is then deployed into a production environment, where it can be used to predict the salary of employee based on specifications.

Overall, a salary prediction system requires a combination of data collection, preprocessing, feature selection, prediction modeling, model evaluation, and deployment. The accuracy and effectiveness of the system depend on the quality and quantity of data collected, the effectiveness of preprocessing techniques, and the choice of machine learning algorithms used in the prediction model.

Different types models are used to find best accuracy:

Those are:

1. Linear Regression: It is a supervised and statistical Machine Learning algorithm used to predict a continuous output variable (also known as a dependent variable) based on one or more input variables (also known as independent or predictor variables). The relationship between the input variables and output variable is assumed to be linear, meaning that the relationship can be represented by a straight line. The algorithm tries to find the best fitting line (known as the regression line) that passes through the data points, minimizing the difference between the predicted and actual values of the output variable.

The equation for a simple linear regression can be written as:

$y = k0 + k1 * x$

where y is output variable, x is input variable, k0 is the intercept, and k1 is the coefficient of the input variable.

2. Random forest:

Random Forest is a supervised and versatile algorithm that can be used for classification and regression problems. It is also relatively easy to use, requires minimal data pre processing, and can handle both numerical and categorical data. Additionally, Random Forest has the ability to handle missing data and outlier values, making it a popular choice for many machine learning applications.

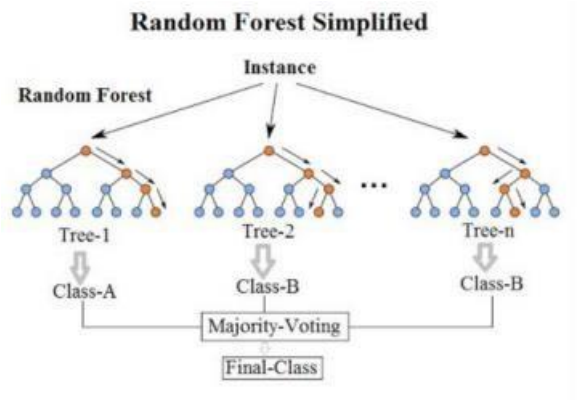


Fig 5: Random Forest Model

3. Logistic regression: Logistic Regression is a supervised and statistical method used for binary classification problems, where the outcome variable takes only two values (0 or 1). The goal of logistic regression is to find out the best fitting line (or hyperplane in higher dimensions) that separates the two classes.

The logistic function is given by:

$p = 1 / (1 + \exp(-z))$

where p is the probability of the positive class, z is the linear combination of the input features and their corresponding weights, and exp is the exponential function.

Logistic Regression works by optimizing the weights of the input features to maximize the likelihood of the observed data given the model parameters. This optimization is usually done using

maximum likelihood estimation or gradient descent.

Logistic Regression is a popular algorithm due to its simplicity and interpretability. It can handle both categorical and continuous input features and is robust to noise and outliers. Additionally, it can be easily extended to handle multi-class classification problems using techniques such as One-vs-All and Softmax regression.

VIII. RESULT AND ANALYSIS

The accuracy of the different model is shown below:

Algorithm	Accuracy
Linear Regression	83.20
Random Forest	85.37
Logistic Regression	92.47

The above table shows the accuraces of different models which are created by using the mentioned machine learning algorithms. Among all above models ,the model which is created by using Logistic Regression algorithm got good accurance .So we consider it as the final model.

This section explains final output which detects used salary

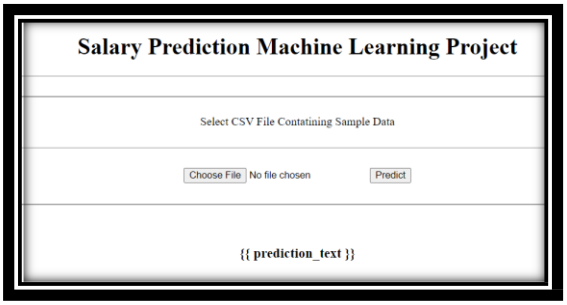


Fig.7: Home Page

IX. CONCLUSION:

We have used 3 algorithms like Linear Regression, Random Forest , Logistic Regression in order to predict the salary. The accuracy varies for different algorithms. The accuracy for Random Forest algorithm is 85.37% when. The accuracy of Linear Regression algorithm is 83.20% when correlation and information gain are applied. The highest accuracy for Logistic Regression using is 92.47%

X REFERENCES

[1] A. K. Lakshmi and A.Parkavil Predicting thecourse knowledge level of students using datamining techniques, IEEE InternationalConference on Smart Technologies andManagement for Computing, Communication,Controls, Energy and Materials ,2017

[2] A.W. Husain, N.A Rashid and A.M. Shahiri, AReview on Predicting Students Performanceusing Data mining Techniques ProcediaComputer Science 72:414-422, 2015.

[3] Richard A Huebner, A Survey of EducationalData-Mining Research, Academic and BusinessResearch Institute,2013

[4] Pornthep Khongchai, Pokpong Songmuang,improving students' motivation to study usingsalary prediction system, 13th International JointConference on Computer Science and SoftwareEngineering, 2016

[5] S. Anupama Kumar Vijayalakshmi M.N.Inference of Naïve Bayeˇs Technique on StudentAssessment Data, Communications in Computer and Information Science book series (CCIS,volume 270),2011

[6] John Jerrim, Do college students make betterpredictions of their future income than youngadults in the labor force ?, Education Economics,Taylor & Francis Journals, vol. 23(2), pages 162-179, 2015

[7] Karlar Hamlen and William A. Hamlen, FacultySalary as a predictor of student outgoing salariesfrom MBA programs, Journal of Education forBusiness, 2016

[8] Rajveer Singh, A Regression Study of SalaryDeterminants in Indian Job Markets for EntryLevel Engineering Graduates, MastersDissertation. Dublin Institute of Technology,2016.

[9] C.C. Hung, E.-P. Lim, On Aggregating Salaries ofOccupations from Job Post and Review Data",IEEE Access, Volume 9, 2021

DB13

ORIGINALITY REPORT

14%	8%	7%	8%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	"Computer Networks and Inventive Communication Technologies", Springer Science and Business Media LLC, 2023 Publication	2%
2	Submitted to Liverpool John Moores University Student Paper	2%
3	Submitted to University of Sunderland Student Paper	1%
4	Submitted to Rochester Institute of Technology Student Paper	1%
5	www.mdpi.com Internet Source	1%
6	www.researchgate.net Internet Source	1%
7	Tomáš Effenberger, Radek Pelánek, Jaroslav Čechák. "Exploration of the robustness and generalizability of the additive factors model", Proceedings of the Tenth International	1%

	Conference on Learning Analytics & Knowledge, 2020 Publication	
8	github.com Internet Source	1 %
9	Submitted to London Business School Student Paper	1 %
10	Submitted to Imperial College of Science, Technology and Medicine Student Paper	1 %
11	Prayitno Abadi, Umar Ali Ahmad, Yuichi Otsuka, Punyawi Jamjareegulgarn et al. "Modeling Post-Sunset Equatorial Spread-F Occurrence as a Function of Evening Upward Plasma Drift Using Logistic Regression, Deduced from Ionosondes in Southeast Asia", Remote Sensing, 2022 Publication	1 %
12	www.sas.com Internet Source	1 %
13	medinform.jmir.org Internet Source	<1 %
14	"Data Mining and Data Warehousing", Studies in Computational Intelligence, 2007 Publication	<1 %
15	mdpi-res.com Internet Source	<1 %

16	scholarworks.umt.edu Internet Source	<1 %
17	dokumen.pub Internet Source	<1 %
18	fau.digital.flvc.org Internet Source	<1 %
19	repository.uel.ac.uk Internet Source	<1 %
20	www.hindawi.com Internet Source	<1 %

Exclude quotes	On	Exclude matches	Off
Exclude bibliography	On		

Conference Certificates

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
NECICAIEA2K23094

Artificial Intelligence and Its Emerging Areas
NEC-ICAIEA-2K23
17th & 18th March, 2023

Organized by Department of Computer Science and Engineering in Association with CSI

Certificate of Presentation

This is to Certify that Ajay Babu Duddukuri , Narasaraopet Engineering College has presented the paper title Salary Prediction using Machine Learning in the International Conference on Artificial Intelligence and Its Emerging Areas-2K23 [NEC-ICAIEA-2K23], Organized by Department of Computer Science and Engineeringin Association with CSI on 17th and 18th March 2023 at Narasaraopeta Engineering College, 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
NECICAIEA2K23094

Artificial Intelligence and Its Emerging Areas

NEC-ICAIEA-2K23

17th & 18th March, 2023

Organized by Department of Computer Science and Engineering in Association with CSI

Certificate of Presentation

This is to Certify that Peddireddy Venkata Rohith , Narasaraopet Engineering College has presented the paper title Salary Prediction using Machine Learning in the International Conference on Artificial Intelligence and Its Emerging Areas-2K23 [NEC-ICAIEA-2K23], Organized by Department of Computer Science and Engineeringin Association with CSI on 17th and 18th March 2023 at Narasaraopeta Engineering College, 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
NECICAIEA2K23094

International Conference on
Artificial Intelligence and Its Emerging Areas
NEC-ICAIEA-2K23
17th & 18th March, 2023
Organized by Department of Computer Science and Engineering in Association with CSI

Certificate of Presentation

This is to Certify that Bapathu Sandeep Reddy , Narasaraopet Engineering College has presented the paper title Salary Prediction using Machine Learning in the International Conference on Artificial Intelligence and Its Emerging Areas-2K23 [NEC-ICAIEA-2K23], Organized by Department of Computer Science and Engineeringin Association with CSI on 17th and 18th March 2023 at Narasaraopeta Engineering College, 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
NECICAIEA2K23094

Artificial Intelligence and Its Emerging Areas

NEC-ICAIEA-2K23

17th & 18th March, 2023

Organized by Department of Computer Science and Engineering in Association with CSI

Certificate of Presentation

This is to Certify that M.Satyam Reddy , Narasaraopet Engineering College has presented the paper title Salary Prediction using Machine Learning in the International Conference on Artificial Intelligence and Its Emerging Areas-2K23 [NEC-ICAIEA-2K23], Organized by Department of Computer Science and Engineeringin Association with CSI on 17th and 18th March 2023 at Narasaraopeta Engineering College, Narasaraopet, A.P., India.

Convenor	Chief-Convenor	Principal, Patron
Dr.S.V.N.Srinivasu	Dr.S.N.Tirumala Rao	Dr. M. Sreenivasa Kumar