EMPLOYEE SALARY PREDICTION USING MACHINE LEARNING

Object Oriented Software Engineering Lab

By

1.	M. Jyothi Prasanna	2341106
2.	M.L.S. Sundar Kumar	2341107
3.	M. Suchitra	2341108
4.	M. Tirumula	2341109
5.	M. Durga Bhavani	2341110
6.	M. Bindu Madhavi	2341111
7.	M. Naga Prasanna	2341112
8.	M. Jagadeesh	2341113
9.	M.N. Ramya Sri	2341114
10.	M. Gayathri	2341115
11.	M. Keerthi Sree	2341116
12.	M. Sireesha	2341117
13.	M. Jaya Ganesh	2341118
14.	M. Lokeswari	2341119
15.	M.V. Sandeep Kumar	2341120

DEPARTMENT OF COMPUTER SCIENCE

P.G COURSES AND RESEARCH CENTER

D.N.R.COLLEGE

(Affiliated to Adikavi Nannaya University)

BHIMAVARAM-534201

DEPARTMENT OF COMPUTER SCIENCE

P.G. COURSES AND RESEARCH CENTRE

D.N.R. COLLEGE

(Reaccredited at A++ grade by NAAC)

(Affiliated to Adikavi Nannaya University)

BHIMAVARAM-534201

CERTIFICATE

This is to certify that the **“Object Oriented Software Engineering Lab”** entitled **“Employee Salary Prediction Using Machine Learning”** Submitted by **Ms. M.Bindu Madhavi Regd.No.2341111** of II **MCA(Master of Computer Application)** III semester has been done during the academic year 2024-2025.

Internal Examiner   Head OF The Department

External Examiner

EMPLOYEE SALARY PREDICTION USING MACHINE LEARNING

ABSTRACT

ABSTRACT

Machine learning is a form of technology that involves creating a computer programme capable of analysing data and use it to learn for itself in order to predict or detect accurately. Today's technology that predicts very precisely, almost like a human, is very popular and helps to solve most prediction and detection issues. In this research, we propose a machine learning approach and certain essential attributes for employee wage prediction. With the help of our salary projection system, school students will receive better support regarding the salaries they may expect to receive once they have finished their courses.Through this paper we have tried to provide a system for salary prediction during which data process in technique is employed.

Our salary prediction system is aimed toward providing better assistance to the school students regarding the salary that they will aspect after completing their course. Not only they are going to be ready to get a thought of their deserving salary but also, they will get to understand about the talents that they have to satisfy their professional goals. This may enhance the motivation of scholars who are enrolled in education institutes and supply better assistance also. We have used data mining techniques for comparison as they perform best.

From this prediction the salary of an employee can be observed according to a particular field according to their qualifications. It helps to see the growth of any field. In the project, we have used Linear Regression as an algorithm for prediction.Therefore, this kind regression technique looks for a linear type of relationship between input x and output y. Apart from Linear Regression, other types of regression techniques are also used like the Decision Tree Regressor and Random Forest Regressor. Since nothing in this universe can be termed as “perfect”, thus a lot of features can be added to make the system more widely inacceptable and more user friendly. This will not only help to predict salaries of other fields but so will be more user beneficial. In the upcoming phase of our project, we will be able to connect an even larger dataset to this model so that the training can be even better. This model could check for new data, once in a month, and incorporate them to expand the dataset and produce better resu

INDEX

TITLES PAGE NO

INTRODUCTION

1. INTRODUCTION

Nowadays as we will see, data mining is becoming the latest trend within the computer sector which involves finding beneficial patterns and knowledge from systems. Under data processing, educational data processing is also booming which involves extracting useful information from educational data systems like student admission, course registration, course management system and lots of other systems regarding students at various levels of institute from school to college levels. The major focus of educational data mining is to assist the institutes to arrange their students far better and aid them to reinforce their performance. Various machine learning concepts are applied to review the data collected from learning. Figure1 shows the application of educational data mining systems.

As now seen, many students select their course on the idea of trend, or they select course on the idea of their peer suggestion. The impact of all such is that the performance of scholars goes to pot and this has a tendency to throw in the towel from institutes. The performance of such students can be enhanced if we offer them with samples of their college graduates as they have already taken the very same path and that they may need to face the same issues. Their career and aid could help the scholars to be motivated for his or her courses and be focused. The salary rate of passed out students will function as an interesting point which can further help the scholars presently enrolled within the course to maneuver forward in their career. The salary rate however depends on the prestige of the school's name, student score in academics and other activities that he/she does during college times. So, to predict the salary, records of graduated people are used as a reference for salary rate. This model is often implemented as a tool to point out the way to achieve different salary brakets by taking reference of graduated students by considering their profiles as they were during their college durations.

Reinforcement learning: Reinforcement learning is an area of Machine Learning. Reinforcement. It is about taking suitable action to maximize reward in a particular situation. It is employed by various software and machines to find the best possible behavior or path it should take in a specific situation. Reinforcement learning differs from supervised learning in a way that in supervised learning the training data has the answer key with it so the model is trained with the correct answer itself whereas in reinforcement learning, there is no answer but the reinforcement agent decides what to do to perform the given task. In the absence of a training dataset, it is bound to learn from its experience. Reinforcement machine learning algorithms is a learning method that interact. with its environment by producing actions and discovering errors or rewards.

Trial and error search and delayed reward are the most relevant characteristics of reinforcement learning. This method allows machines and software agents to automatically determine the ideal behavior within a specific context in order to maximize its performance. Simple reward feedback is required for the agent to learn which action is best; this is known as the reinforcement signal. The project uses various regression techniques for predicting the salary of the employees. The techniques are listed as follows.

I. Linear Regression: In Linear regression we are given a number of pred continuous response variables, and we try to find a relationship between those variables that predict variables and allows us to predict a continuous outcome.

For example, given X and Y, we fit a straight line that estimates the coefficients like Ordinary Least Squares and Gradient Descent between the sample to minimize the distance using methods to Points and the fitted line.

2. **Decision Tree Regressor**: Decision tree builds regression or classification models in the form of a tree structure. It reais down a dataset into smaller and smaller subsets while at the same an associated decision tree is incrementally developed. The final result is a tree with decision nodes and leaf nodes. A decision node has two or more branches, each representing values OF the attribute. Leaf node represents 5 decisions on the numerical target. The topmost decision node in a tree which corresponds to the best predictor called root node. Decision trees can handle both categorical and numerical data,

3.Random Forest Regressor: Random forests or random decision forests are an ensemble learning method for classification, regression and other tasks that operates by constructing a multitude of decision trees at training time for regression tasks, the mean or average prediction of the individual trees is returned.

LITERATURE SURVEY

2. LITERATURE SURVEY

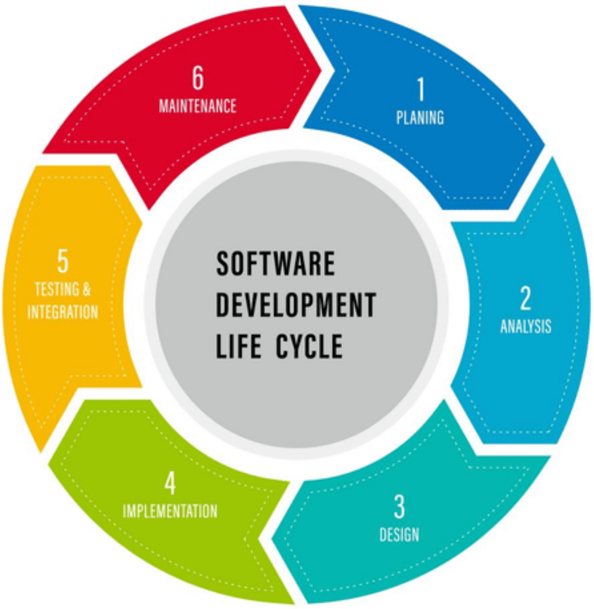
- K. Lakshmi and Paravel, performed analysis in order that they might understand the extent of students (their knowledge) by means of associative rule mining. They applied association rule mining alongside linear regression to observe the group (or cluster) to which the student's knowledge is the closest to. An important concept to provide efficient and effective learning is data mining for researchers. Different kinds of recommendations are produced by recommender systems. The concept of data mining is primarily used for studying the patterns of learners. The efficiency of the system can be enhanced by personalizing the system for the students .
- Pok pong Song Muang along with Pornthip Thongchai proposes a system of salary prediction to enhance the motivation of students in college. A seven-feature prediction model was generated by using the technique of decision tree. With more feedback from faculties and staff the performance of the system was improved. Jobs which are related to the field of study of the students were also included in the paper to enhance the performance of the system. In the Bayesian network is used for classifying students based on marks scored by them.
- The cross validation used for evaluating the model is 10-fold. In Ordinary least squares regression model is used to create models for predicting salaries of students on the basis of profiles and family background. A hierarchical linear regression is used by Karla et al for building a model by taking students profile as fixed parameter and salary as output variable. However, this model has majorly two problems, one is this system is personalized for students as it predicts the salary of the student group, second is the output of the model requires immense statistical knowledge to understand the predictions.
- Research by Rajveer Singh conducted a study for salary estimation for entry-level Indian engineering graduates indicates that the academic performance in school and college, school affiliation, college reputation are important indicators for starting salary. C.-C. Hung, E.-P. Lim in their paper proposed the "Company, Occupation, Company" (COC) model to derive unbiased salaries by aggregating job review and job post data.
- Pornthip Thongchai, Pok pong Song Muang, "Improving Students' Motivation to Study using Salary Prediction System" - proposed prediction model using Decision tree technique with seven features. Moreover, the result of the system is not only a predicted salary, but also the 3-highest salary of the graduated students which share common attributes to the users. To test the system's efficiency, they set up an experiment by using 13,541 records of actual graduated student data. The total result in accuracy is 41.39%.

SYSTEM ANALYSIS

3. SYSTEM ANALYSIS

Software Development Life Cycle (SDLC)

SDLC is a process followed for a software project, within a software organization. It consists of a detailed plan describing how to develop, maintain, replace and alter or enhance specific software. The life cycle defines a methodology for improving the quality of software and the overall development process.



Requirements:

Requirement analysis is the most important and fundamental stage in SDLC. It is performed by the senior members of the team with inputs from the customer, the sales department, market surveys and domain experts in the industry. This information is then used to plan the basic project approach and to conduct product feasibility study in the economical, operational and technical areas.Business requirements are gathered in this phase. This phase is the main focus of the project managers and stakeholders. Meetings with managers, stakeholders and users are held in order to determine the requirements.

Design:

The software system design is produced from the results of the requirements phase. Architects have the ball in their court during this phase and this is the phase in which their focus lies. This is where the details on how the system will work is produced. Architecture, including hardware and software, communication, software design (UML is produced here are all part of the deliverables of a design phase.

Implementation:

During the implementation stage, developers complete the application per established specifications. Security activities in this stage focus on technology-specific as well (automated) code reviews. Technology-specific guidance is typically in the form of checklists that help developers implement things securely. Checklists may also contain things to avoid along with secure alternatives. The guidelines should be actionable and language- or framework-specific For instance,PHP developers should use disodium's crypto_aead_aes256gcm_encrypt function to encrypt data.

Testing:

During testing io is tested against the requirements to make sure that the product 1s actually solving the needs address and gathered during the requirements phase. Unit tests and system/acceptance tests are done during this phase. Unit tests act on a specific component of the system, while system tests act on the system as a whole

STUDY OF THE SYSTEM

In the flexibility of uses the interface has been developed with graphics concepts in mind associated through a browser interface. The GUTs at the top level have been categorized.

1.Administrative User Interface Design

2.The Operational and Generic User Interface Design. The administrative user interface concentrates on the information that is practically part of the organizational activities and which needs proper authentication for the data collection.

3.1 EXISTING SYSTEM

1.Traditional Methods:

- Description: Employee salary determination is often based on manual evaluations, HR policies, and benchmarks without data-driven insights.

Limitations:

- Subjective decision-making leads to inconsistencies.
- Lack of data analysis can result in underpaying or overpaying employees.
- Difficulty in adapting to market changes or industry trends.

2.Spreadsheet-Based Systems:

- Description: Use of spreadsheets for salary management, which requires manual input and updates.

Limitations:

- Prone to human error.
- Time-consuming for data entry and analysis.
- Limited analytical capabilities; no predictive modeling.

3. Basic Statistical Analysis:

- Description: Some organizations may use basic statistical techniques to analyze salary data.

Limitations:

- Limited ability to handle complex datasets.
- Often fails to incorporate various influencing factors (experience, education, location).

3.2 PROPOSED SYSTEM

1. Machine Learning-Based Prediction:

- Description: A data-driven approach using machine learning algorithms to predict employee salaries based on various features (experience, education, job role, etc.).

Advantages:

- Data-Driven Insights: Provides objective predictions based on historical data.
- Adaptability: Quickly adjusts to changes in the job market or company policies.

2. Automated Data Processing:

- Description: Automated data collection and preprocessing to ensure accurate and timely updates.

Advantages:

- Reduces manual errors.
- Streamlines data management, allowing HR to focus on strategic tasks.

3. Predictive Modeling:

- Description: Utilizes advanced algorithms (e.g., regression models, decision trees) to analyze trends and predict future salaries.

Advantages:

- Can handle large datasets with multiple variables.
- Identifies patterns and correlations that traditional methods might miss.

4. User-Friendly Interface:

- Description: A web-based UI for HR professionals to easily input data and obtain predictions.

Advantages:

- Intuitive design improves accessibility for non-technical users.
- Facilitates quick decision-making through visualized results and reports.

5. Reporting and Visualization:

- Description: Generates comprehensive reports on salary predictions, trends, and model performance.

Advantages:

- Supports strategic planning and budgeting.

3.3 FEASIBILITY STUDY

Feasibility study is a preliminary exploration of a proposed project or undertaking to determine its merits and viability. A feasibility study aims to provide an independent assessment that examines all aspects of a proposed project, including technical, economic, financial, legal, and environmental considerations. This information then helps decision-makers determine whether or not to proceed with the project. Without a feasibility study, it cannot be easy to know whether or not a proposed project is worth pursuing.

There are aspects in the feasibility study portion of the preliminary investigation:

- Technical Feasibility
- Operational Feasibility
- Economic Feasibility

3.3.1 TECHNICAL FEASIBILITY |

The technical issue usually raised during the feasibility stage of the investigation includes the following:

- Does the necessary technology exist to do what is suggested?
- Do the proposed equipment have the technical capacity to hold the data required to Use the new system?
- Will the proposed system provide adequate response to inquiries, regardless of the Number or location of users?
- Can the system be upgraded if developed?
- Are there technical guarantees of accuracy, reliability, ease of access and data security?

Earlier no system existed to cater to the needs of 'Secure Infrastructure Implementation

System. The current system developed is technically feasible. It is a web-based user interface for audit workflow at NIC-CSD. Thus, it provides easy access to the users. The database's purpose is to create, establish and maintain a workflow among various entities in order to facilitate all concerned users in their various capacities or roles. Permission to the users would be granted based on rules. Therefore, it provides the technical guarantee of accuracy, reliability and security. The software and hard requirements for the development of this project are not Many and are already available in-house at NIC or are available as free as open source.

3.3.2 OPERATIONAL FEASIBILITY

Proposed projects are beneficial only if they can be turned into an information system. at will meet the organization Operating requirements. Operational feasibility studies also examine how a project plan satisfies the requirements identified in the requirements analysis phase of system development. Operational feasibility of the project aspects are to be taken as an important part of the project implementation. Some of the important issues raised are to test the operational feasibility of a project includes the following:

- Is there sufficient support for the management from the users?
- will the system be used and work properly if it is being developed and implemented?
- Will there be any resistance from the user that will undermine the possible application benefits?

The system 15 targeted to be in accordance with the above-mentioned issues. Beforehand, the management issues and user requirements have been taken into consideration. So, there is no question of resistance from the users that can undermine the possible application benefits. The well-planned design would ensure the optimal utilization of the computer resources and would help in the improvement of performance status

3.3.3 ECONOMICAL FEASIBILITY

This assessment typically involves a cost/ benefits analysis of the project, helping organizations determine the viability, cost, and benefits associated with a project before financial resources are allocated. A system can be developed technically and that will be used if installed must still be a good investment for the organization. In the economic feasibility, the development cost in creating the system is evaluated against the ultimate benefit derived from the new systems. Financial benefits must equal or exceed the costs.

The system is economically feasible. It does not require any additional hardware or software. Since the interface for this system is developed using the existing resources and technologies available at NIC, There is nominal expenditure and economic feasibility for certain.

3.4 SYSTEM REQUIREMENTS:

System Requirements

RAM	: 4GB and Higher
Processor	: Intel i3 and above
Hard Disk	: 500GB: Minimum

Software Requirements

OS: Windows	: or Linux
Python IDE	: python 2.7.x and above PyCharm IDE Required
	Setup tools and pip to be installed for 3.6 and above
Language	: python Scripting

3.4.1 FUNCTIONAL REQUIREMENTS

Functional requirements describe what the software should do. They define the functions or features that the system must have. All these functionalities need to be necessarily incorporated into the system as a part of the contract. These are represented or stated in the form of input to be given to the system, the operation performed and the output expected. They are the requirements stated by the user which one can see directly in the final product, unlike the non-functional requirements.

Functional requirements define the specific behaviors and functionalities that a system must support. For an Employee Salary Prediction system using machine learning, here are key functional requirements to consider:

1. User Authentication

- **Login/Logout:** Users must be able to securely log in and log out of the system.
- **Role Management:** The system should support different user roles (e.g., HR Manager, Employee) with appropriate access permissions.

2. Data Input

- **Employee Data Entry:** Users should be able to input employee data including:
- Years of experience
- Education level
- Department (optional)
- Location (optional)
- **Data Validation:** The system should validate input data (e.g., ensuring numeric values for years of experience).

3. Salary Prediction

- **Model Prediction:** The system must utilize a trained machine learning model to predict the salary based on the input features.
- **Display Prediction Results:** After processing, the system should display the predicted salary to the user.

4. Model Training and Management

- **Train Model:** Users with appropriate permissions should be able to upload historical salary data to train or retrain the model.
- **View Model Performance:** Users should be able to view metrics such as Mean Squared Error (MSE) and R^2 score to evaluate the model's performance.

5. Reporting

- **Generate Reports:** The system should allow users to generate reports summarizing salary predictions and trends.
- **Export Data:** Users should be able to export prediction results and reports in common formats (e.g., CSV, PDF).

6. User Interface

- **Dashboard:** The system should provide a user-friendly dashboard that summarizes key functionalities, including input fields, prediction results, and model performance metrics.
- **Help and Support:** The interface should include help sections or tooltips to assist users in understanding how to use the system.

7. Data Storage

- **Database Management:** The system must store employee data, predictions, and model parameters in a secure database.
- **Data Backup:** The system should implement regular data backups to prevent data loss.

8. Notifications

- **User Alerts:** The system should notify users of important events (e.g., successful model training, prediction completion) via in-app notifications or emails.

9. Security

- **Data Encryption:** The system must ensure that sensitive employee data is encrypted during transmission and at rest.
- **Access Control:** Implement role-based access control to ensure that users can only access features and data appropriate to their role.

10. Integration

- **API Access:** The system should provide APIs for integration with other HR systems or applications for seamless data exchange.
- **Data Import/Export:** Users should be able to import historical data from various formats (e.g., Excel, CSV) into the system for model training.

Outputs from computer systems are required primarily to communicate the results of accessing 10 users. They are also used to provide a permanent copy of the results for later consultation- The various types of outputs in general are:

- External Outputs, whose destination is outside the organization,
- Internal Outputs whose destination is within organization and they are the
- User's main interface with the computer.
- Operational outputs whose use is purely within the computer department.

- Interface outputs, which involve the user in communicating directly.
- Understanding user’s preferences, expertise level and his business requirements through a friendly questionnaire.
- Input data can be in four different forms - Relational DB, text files, .xls and
- To produce a cost-effective method of input.
- To achieve the highest possible level of accuracy.
- To ensure that the input is acceptable and understood by the user

INPUT STAGES

The main input stages can be listed as below:

- Data recording
- Data transcription
- Data conversion
- Data verification
- Data control
- Data transmission
- Data validation

INPUT TYPES

It is necessary to determine the various types of inputs, Inputs can be categorized as follows:

- External inputs, which are prime inputs for the system
- Internal inputs, which are user communications with the system.
- Operational, which are the computer department's communications to the system?
- Interactive, which are inputs entered during a dialogue

3.4.2 NON-FUNCTIONAL REQUIREMENTS

Non-functional requirements describe how the software performs a task rather than what it should do. They define the quality attributes, performance criteria, and constraints.These are basically the quality constraints that the system must satisfy according to the project contract. Nonfunctional requirements, not related to the system functionality, rather define how the system should perform The priority or extent to which these factors are implemented varies from one project to other. They are also called non-behavioral requirements.

Non-functional requirements (NFRs) specify the quality attributes of a system rather than specific behaviors or functions. For an Employee Salary Prediction system using machine learning, here are some key non-functional requirements to consider:

1. Performance

- **Response Time:** The system should provide salary predictions within 2 seconds for user inputs.
- **Throughput:** The system should handle at least 100 concurrent users without degradation in performance.

2. Scalability

- **Horizontal Scalability:** The system should be able to accommodate increased loads by adding more instances of the model as needed.
- **Data Scalability:** The system should be able to handle larger datasets (e.g., up to 1 million records) as the organization grows.

3. Reliability

- **Availability**: The system should maintain 99.9% uptime, ensuring users can access it consistently.
- **Fault Tolerance:** The system should be able to recover gracefully from failures (e.g., model errors, input errors) without crashing.

4. Usability

- **User Interface:** The user interface should be intuitive and easy to navigate, requiring minimal training for users.

- **Accessibility:** The application should meet accessibility standards (e.g., WCAG) to accommodate users with disabilities.

5. Security

- **Data Protection:** User and salary data must be encrypted both in transit and at rest to protect sensitive information.
- **Authentication and Authorization:** The system should implement role-based access control to ensure that only authorized users can access specific features.

6. Maintainability

- **Code Quality:** The codebase should follow best practices, including clear documentation and modular design to facilitate future updates and maintenance.
- **Monitoring:** The system should include monitoring tools to track model performance and detect anomalies in real-time.

7. Portability

- **Cross-Platform Compatibility:** The application should be compatible with major operating systems (Windows, macOS, Linux) if it's a desktop application, or accessible across different web browsers if it's a web application.

8. Compliance

- **Regulatory Compliance:** The system must comply with relevant data protection regulations (e.g., GDPR, HIPAA) regarding the handling of personal and sensitive information.

9. Maintainability

- **Documentation:** Comprehensive documentation should be provided for both the system and the machine learning model to facilitate onboarding and maintenance.
- **Version Control:** The system should utilize version control to manage changes and maintain historical versions of the model and codebase.

10. Efficiency

- **Resource Utilization:** The system should efficiently use computational resources, optimizing CPU and memory usage during model training and prediction.

PERFORMANCE REQUIREMENTS

Performance is measured in terms of the output provided by the application.requirement specification plays an important part in the analysis of a system.only when the requirement specifications are properly given,it is possible to design a system,which will fit into the required environment.It rests largely on the part of the users of the existing system to give the requirement specifications because they are the people who finally use the system.This is because the requirements have to beknowm during the intial stages so that the system can be designed according to those requirements.It is very difficult to change the system once it has been designed and on the other hand designing a system,which does not cater to the user,is of no use.

- The requirement specification for any system can be broadly stated as given below:
- The system should be able to interface with the existing system
- The system should be accurate
- The system should be better than the existing system

3.5 SYSTEM DESIGN

3.5.1 SYSTEM ARCHITECTURE

An Architectural Diagram or a pipeline is used to help automate machine learning workflows. They operate by enabling a sequence of data to be transformed and correlated together in a model that can be tested and evaluated to achieve an outcome, whether positive or negative. The pipeline/ Diagram consists of several steps to train a model. Machine learning pipelines are iterative as every step is repeated to continuously improve the accuracy of the model and achieve 2 successful algorithms. To build better machine learning models, and get the most value from them, accessible, scalable and durable storage solutions are imperative, paving the way for on premises object storage. The steps include:

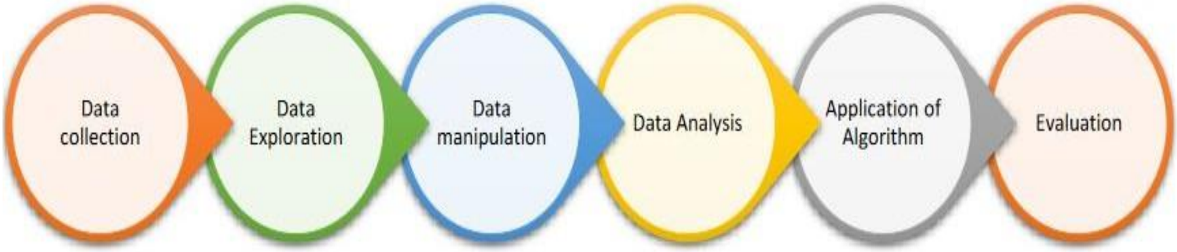
Data Collection: Collecting raw data from billions of datasets available.

Data Exploration: Exploring the data & the features related and being familiar with the data-types.

Data Manipulation: Includes Cleaning of data, treating missing, repetitive values that are present.

Data Analysis: Analyzing the data to increase efficiency while applying the best Algorithm & feature selection according to our preferences.

Application of Algorithm: Applying the algorithm to the model. Evaluation: Using evaluation metrics to calculate the least error and following the above to make further changes.



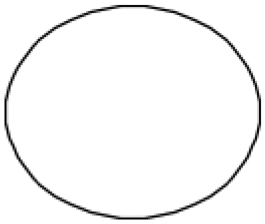
3.5.2 DATA FLOW DIAGRAM

A graphical tool used to describe and analyze the moment of data through a system manual or automated including the process, stores of data, and delays in the system. Data Flow Diagrams are the central tool and the basis from which other components are developed. The transformation of data from input to output, through processes, may be described logically and independently of the physical components associated with the system. The DFD is also known as a data flow graph or a bubble chart.

DFDs are the model of the proposed system. They clearly should show the requirements on which the new system should be built. Later during design activity this is taken as the basis for drawing the system's structure charts. The Basic Notation used to create a DFD's are as follows:

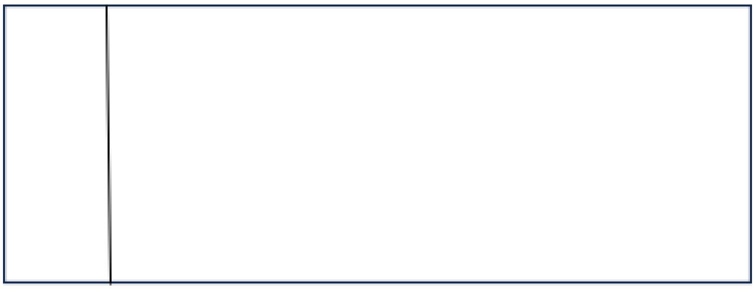
1. **Dataflow:** Data moves in a specific direction from an origin to a destination.

2. **Process:** People, procedures, or devices that use or produce (Transform) Data. The physical component is not identified.



3. **Source**: External sources or destination of data, which may People, programs, Organization other entity

4. Data Store: Here data is stored or referenced by a process in the System



3.5 UML DIAGRAM

UML (Unified Modeling Language) is a general-purpose, graphical modeling language in the field of Software Engineering. UML is used to specify, visualize, construct, and document the artifacts (major elements) of the software system. It was initially developed by Grady Booch, Ivar Jacobson, and James Rumbaugh in 1994-95 at Rational software, and its further development was carried out through 1996. In 1997, it got adopted as a standard by the Object Management Group.

UML (Unified Modeling Language) is a general-purpose, graphical modeling language in the field of Software Engineering. UML is used to specify, visualize, construct, and document the artifacts (major elements) of the software system. It was initially developed by Grady Booch, Ivar Jacobson, and James Rumbaugh in 1994-95 at Rational software, and its further development was carried out through 1996. In 1997, it got adopted as a standard by the Object Management Group.

Software design serves as a foundation for all the software engineering steps that follow. Without a strong design we risk building an unstable system — one that will be difficult to test, one whose quality cannot be assessed until the last stage. The purpose of the design phase is to plan a solution of the problem specified by the requirement document. This phase is the first step in moving from the problem domain to the solution domain. In other words, starting with what is needed, design takes us toward how to satisfy the needs. The design of a system is perhaps the most critical factor affecting the quality of the software; it has a major impact on the later phase, particularly testing, maintenance. The output of this phase is the design document. This document is similar to a blueprint for the solution and is used later during implementation, testing and maintenance. The design activity is often divided into two separate phases: System Design and Detailed Design.

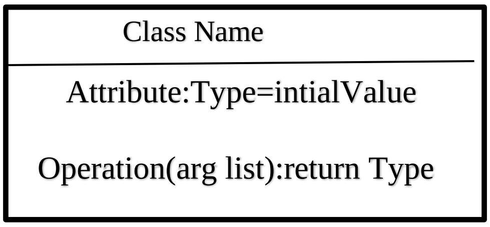
System Design, also called top-level design, aims to identify the modules that should be in the system, the specifications of these modules, and how they interact with each other to produce the desired results. At the end of the system design all the major data structures, file formats, output formats, and the major modules in the system and their specifications are decided. During Detailed Design, the internal logic of each of the modules specified in system design is decided. During this phase, the details of the data of a module is usually specified in a high-level design description language, which is independent of the target language in which the Software will eventually be implemented.

3.5.1 CLASS DIAGRAM

Class diagrams are the backbone of almost every object-oriented method including UML. They describe the static structure of a system.

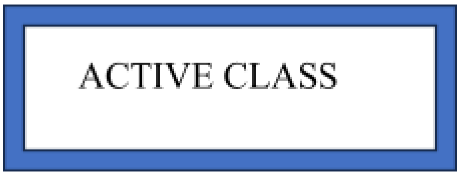
Basic Class Diagram Symbols and Notations

Classes represent abstraction of entities with common characteristics. Associations represent the relationships between classes. Illustrate classes with rectangles divided into compartments. Place the name of the class in the first partition (centered, bolded, and capitalized), list the attributes in the second partition, and write



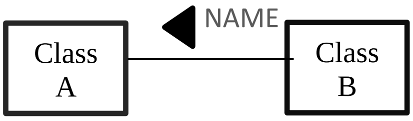
Active class

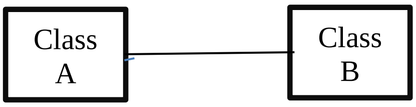
Active classes initiate and control the flow of activity, while passive classes store data and serve other classes. Illustrate active classes with a thicker border.



Associations

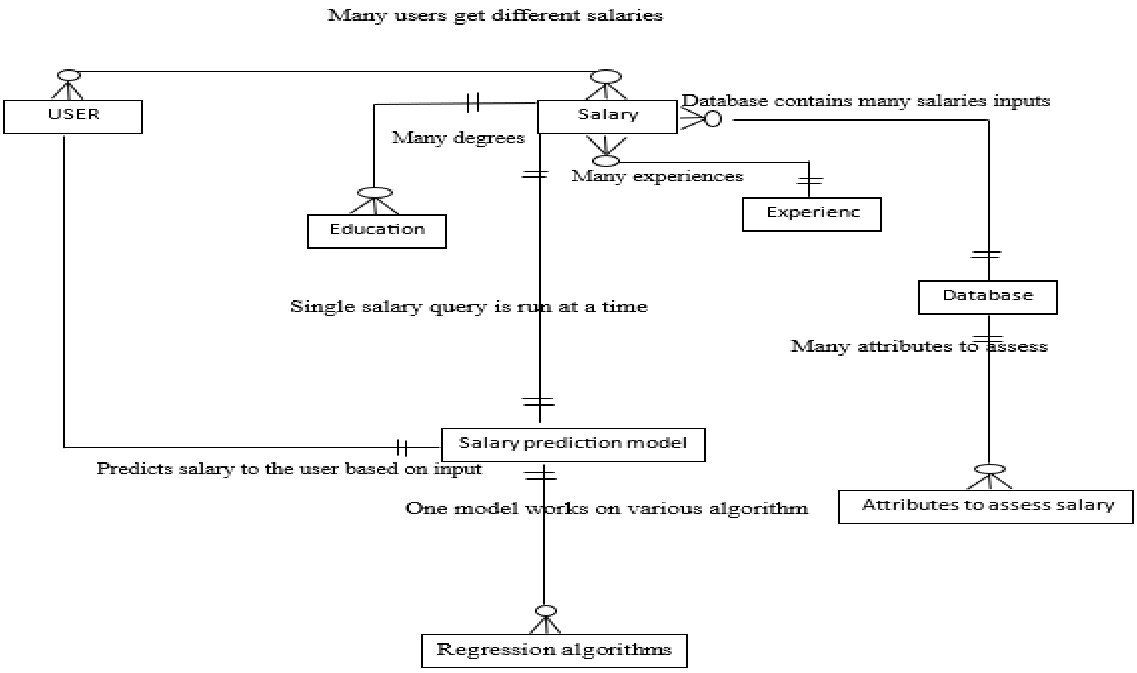
Associations represent static relationships between classes. Place association names above, on, or below the association line. Use a filled arrow to indicate the direction of the relationship. Place roles near the end of an association. Roles represent the way the two classes see each other. Note: It's uncommon to name both the association and the class roles.





Composition and Aggregation

Composition isa special type of aggregation that denotes a strong ownership between Class A hole, and Class B, 1ts part, illustrate Composition with a filled diamond. Use a hollow diamond to represent a simple aggregation relationship, in which the "whole" class plays a more important role than the “part” class, but the two classes are not dependent on each other. The diamond end in both a composition and aggregation relationship points toward the “whole" class or the aggregate.


Many users get different salaries
USER
Salary
Database contains many salaries inputs
Many degrees
Many experiences
Education
Experienc
Single salary query is run at a time
Database
Many attributes to assess
Salary prediction model
Predicts salary to the user based on input
One model works on various algorithm
Attributes to assess salary
Regression algorithms

Class diagram on credit card fraud Detection

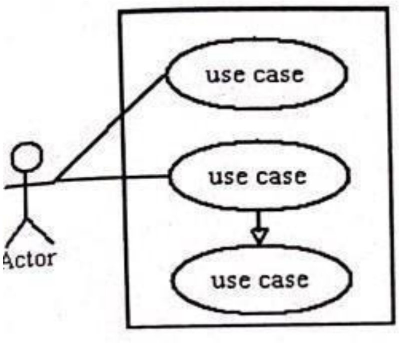
3.5.2 USE CASE DIAGRAM

Use case diagrams model the functionality of a system using actors and use cases. Use cases are services of functions provided by the system to its users.

Basic Use Case Diagram Symbols and Notations:

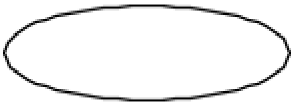
System

Draw your system's boundaries using a rectangle that contains use cases. Place actors outside the system's boundaries.

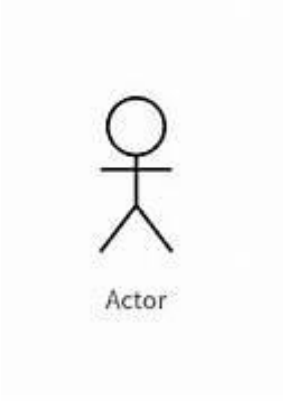


Use Case:

Draw use cases using ovals. Label with ovals with verbs that represent the system's functions.

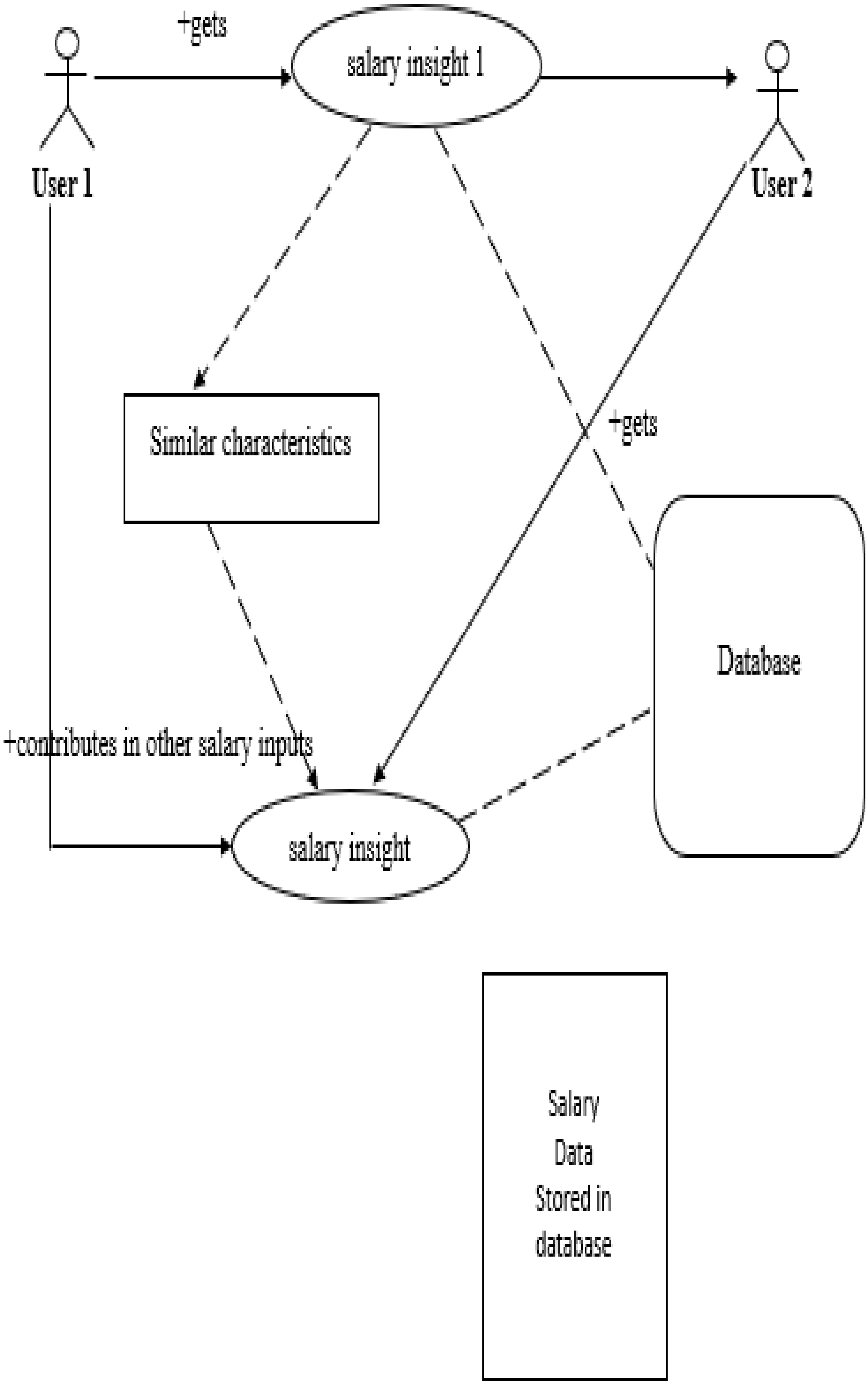


Actor:



Relationships:

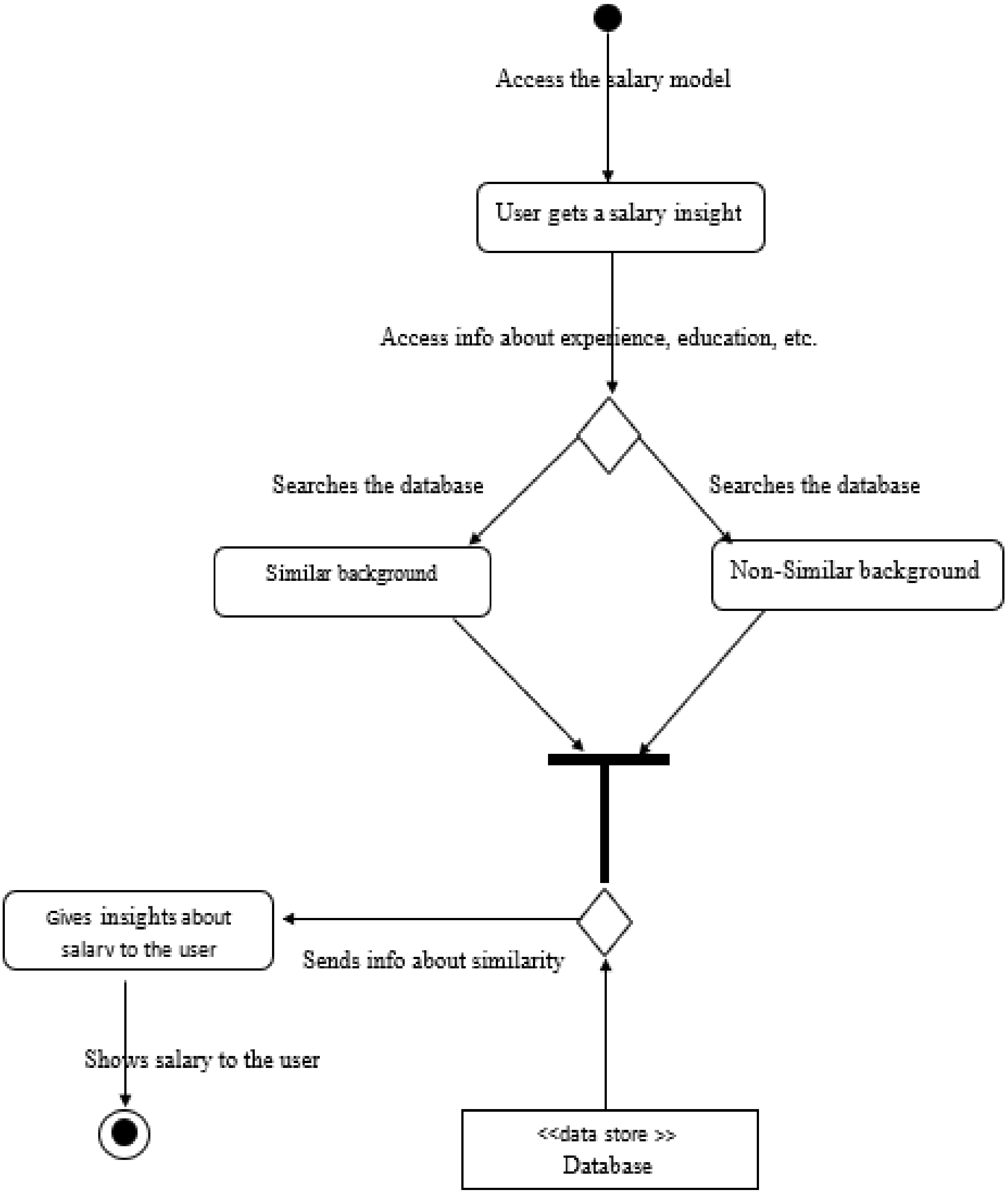
Illustrate relationships between an actor and a use case with a simple line. For relationships among use cases, use arrows labeled either "uses" or "extends." A "uses" relationship indicates that one use case is needed by another in order to perform a task. An "extends" relationship indicates alternative options under a certain use case.



Use diagram on credit card fraud Detection

3.5.3ACTIVITY DIAGRAM

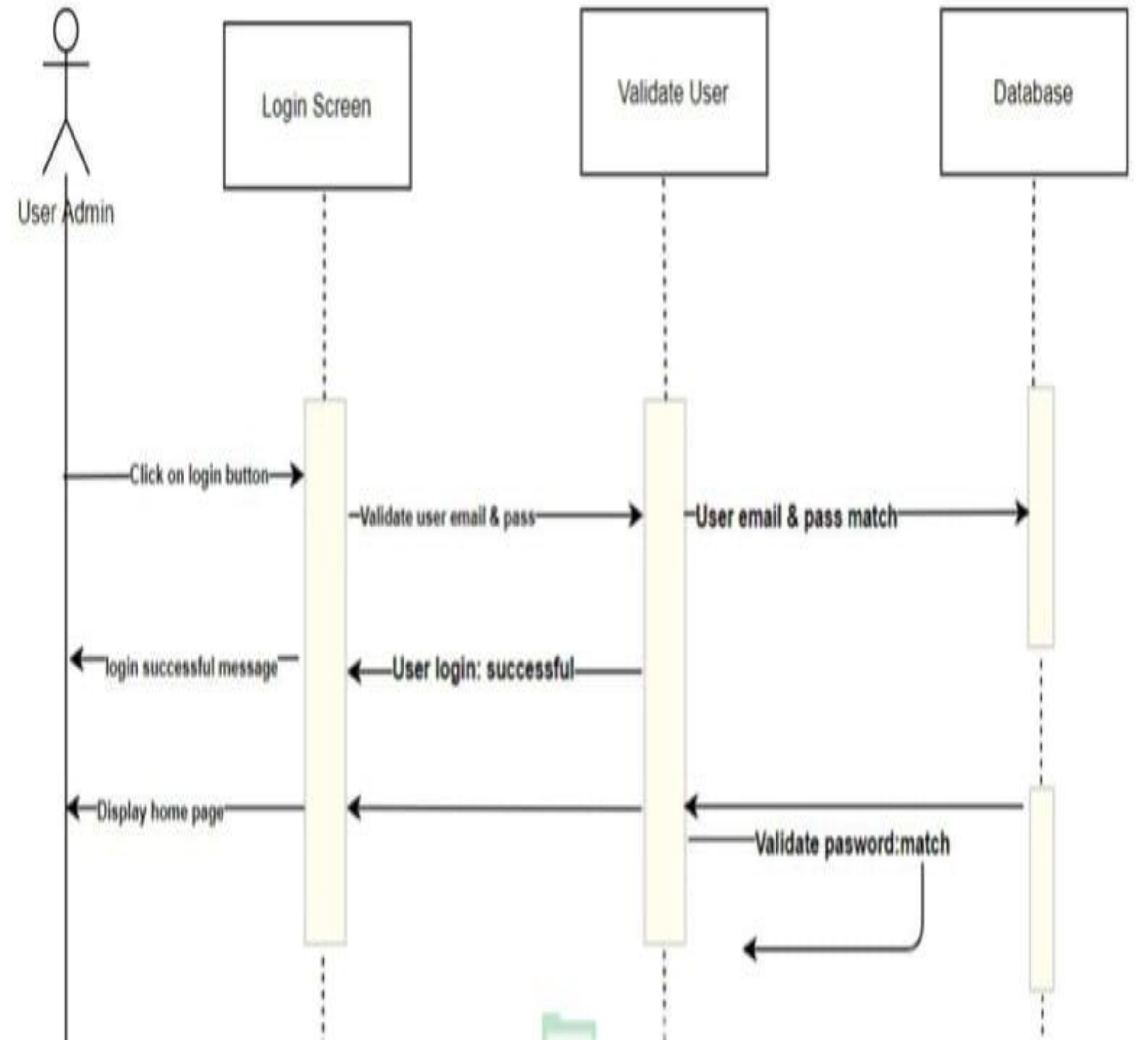
Activity diagrams are graphical representations of Workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.



Activity diagram on credit card fraud Detection

3.5.4 SEQUENCE DIAGRAM

Sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram That shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. A sequence diagram shows, as parallel vertical lines ("lifelines") different processes or objects that live simultaneously, and, as horizontal arrows, the messages exchanged between them, in the order in which they occur. This allows the specification of runtime scenarios in a graphical manner.



Sequence diagram on credit card fraud Detection

3.7 INPUT DESIGN

In an information system, input is the raw data that is processed to produce output. During the input design, the developers must consider the input devices. It should be easy to fill and straightforward.It should focus on user’s attention, consistency,and simplicity.The input design is the link between the information system and the user.

It comprises the developing specification and procedures for data preparation and those steps necessary to put transaction data into a usable form for processing can be achieved by inspecting the computer to read data from a written or printed document or it can occur by having people keying the data directly into the system. The design of input focuses on controlling the amount of input required, controlling the errors, avoiding delay, avoiding extra steps and keeping Process simple. The input is designed in such a way so that it provides security and ease of use With retaining privacy.

Designing an effective input interface for an Employee Salary Prediction model is crucial for usability and user experience. Here’s a structured approach for creating the input design, whether it’s a web application, desktop application, or command-line interface.

Input Design Components

1.Title and Instructions

- **Title:"**Employee Salary Prediction Input"
- **Instructions:** A brief explanation of what the user needs to provide.

2. Input Fields

- Years of Experience
- **Type:** Numeric Input (Slider or Text Box)
- **Description:** "Enter years of experience (e.g., 1 to 10)."
- Education Level
- **Type:** Dropdown Menu or Radio Buttons

Options:

- Bachelors
- Masters
- PhD
- **Description:** "Select the highest education level attained."

3. Additional Features (Optional)

- Department
- **Type:** Dropdown Menu
- **Options:** Engineering, Marketing, Sales, HR, etc.
- **Description:** "Select the department."
- Location
- Type: Text Box or Dropdown Menu

- Description: "Enter the city or select from the list."

4. Submit Button

- **Label:** "Predict Salary"
- **Action:** When clicked, the model processes the input and returns predictions.

5. Clear/Reset Button (Optional)

- **Label:** "Clear Inputs"
- **Action:** Resets all fields to their default values.

1. Years of Experience:

[_____] (Numeric Input)

(Enter a number between 1 and 10)

2. Education Level:

[Dropdown Menu]

- Bachelors (1)
- Masters (2)
- PhD (3)

3. Department:

[Dropdown Menu]

- Engineering
- Marketing
- Sales
- HR

4. Location:

[_____] (Text Input)

(e.g., New York)

Implementation Tips

- User-Friendly Design: Use clear labels and placeholders to guide users on what to enter.
- Validation: Implement input validation to ensure users enter valid data (e.g., years of experience must be a number).
- Responsive Design: Ensure that the interface is mobile-friendly if applicable.
- Feedback: Provide instant feedback (e.g., loading spinners) while predictions are being processed.
- Accessibility: Ensure that the design is accessible to users with disabilities by adhering to web accessibility standards (e.g., proper labels for screen readers).

This structured input design will help users easily provide the necessary information for predicting salaries and enhance their overall experience.

OBJECTIVES

- Input Design is the process of converting a user-oriented description of the input into a based system. This design is important to avoid errors in the data input process and show the Correct direction to the management for getting correct information from the computerized system.
- Iris achieved by creating user-friendly screens for the data entry to handle large volumes of data. The goal of designing input is to make data entry easier and to be free from errors. The data entry screen is designed in such a way that all the data can be performed. It also provides record viewing facilities.
- When the data is entered it will check for its validity. Data can be entered with the help of screens. Appropriate messages are provided as when needed so that the user will not be in maize instant. Thus the objective of input design is to create an input layout that is easy to follow

3.8 OUTPUT DESIGN

The design of output is the most important task of any system. During output design, developers identify the type of outputs needed, and consider the necessary output controls and prototype report layouts. A quality output is one, which meets the requirements of the end user and presents the Information clearly.In output design it is determined how the information is to be displaced for immediate need and also the hard copy output. It is the most important and direct source of information to the user.Efficient and intelligent output design improves the system's relationship to help user decision-making.

1. Designing computer output should proceed in an organized, well thought out manner; the right output must be developed while ensuring that each output element is designed so that people will find the system can use easily and effectively. When analysis design computer output, they should Identify the specific output that is needed to meet the requirements.

2. Select methods for presenting information.

3. Create documents, reports, or other formats that contain information produced by the System.

- Convey information about Past activities, current status or projections of the Future.
- Signal important events, Opportunities, problems, or warnings.
- Trigger an action.

Creating an effective output design for an Employee Salary Prediction model involves clearly presenting the results, predictions, and relevant metrics. Here's a structured approach to designing the output:

Output Design Components

1. Title and Introduction

- Title: "Employee Salary Prediction Results"
- Brief description of the purpose of the model and dataset.

2. Input Features

- Display the features used for predictions (e.g., Years of Experience, Education Level).

Example:

```
Input Features:
- Years of Experience: 5
- Education Level: Masters (2)
```

3. Predicted Salary

- Present the predicted salary based on the input features.

Example:

```
Predicted Salary: $75,000
```

4. Actual Salary (if applicable)

- Show the actual salary if you have it (for comparison).

Example:

```
Actual Salary: $78,000
```

5. Performance Metrics

- Display key evaluation metrics of the model.

Example:

```
Model Performance:
- Mean Squared Error (MSE): 5,000,000
- R^2 Score: 0.85
```

6. Visualizations

- Include graphs that compare actual vs. predicted salaries.
- Example: A scatter plot with a diagonal line representing perfect predictions.

7. Summary and Conclusion

- Briefly summarize the results and any insights gained from the predictions.

The model demonstrates a strong correlation between experience, education, and salary. Further enhancements can be made by incorporating more features.

Input Features:

- Years of Experience: 5

- Education Level: M)
- Predicted Salary:$75,000
- Actual Salary:$78,000(if available)

Model Performance:

- Mean Squared Error (MSE):5,000,000
- R² Score: 0.85

Visual Comparison:

The model demonstrates a strong correlation between experience, education, and salary. Further enhancements can be made by incorporating more features and tuning the model.

Implementation Tips

- Use a clean and professional layout, possibly leveraging HTML/CSS if this is a web-based output.
- Consider using Jupyter Notebooks for an interactive experience, where outputs and visualizations can be easily presented together.
- If deploying as an application, ensure that the interface allows users to input features and view predictions seamlessly.

This structured approach will help users understand the predictions and model performance clearly and effectively.

IMPLEMENTATION

4.1 MODULES

1. Data Collection Module

- Data Sources: Identify and gather data from various sources (e.g., internal HR databases, public datasets).
- Features: Include relevant features such as employee age, education, years of experience, job role, location, and industry.

2. Data Preprocessing Module

- Data Cleaning: Handle missing values, remove duplicates, and correct inconsistencies.
- Feature Engineering: Create new features based on existing data (e.g., salary per year of experience).
- Encoding Categorical Variables: Use techniques like one-hot encoding or label encoding for categorical data.
- Normalization/Scaling: Standardize or normalize numerical features to improve model performance.

3. Exploratory Data Analysis (EDA) Module

- Data Visualization: Use plots (histograms, box plots, scatter plots) to understand distributions and relationships.
- Correlation Analysis: Assess relationships between features and the target variable (salary).

4. Model Selection Module

- Algorithm Selection: Choose appropriate machine learning algorithms (e.g., Linear Regression, Decision Trees, Random Forest, XGBoost).
- Model Comparison: Compare models based on performance metrics (e.g., RMSE, MAE).

5. Model Training Module

- Training the Model: Use the training dataset to fit the selected model.
- Hyperparameter Tuning: Optimize model parameters using techniques like Grid Search or Random Search.

6. Model Evaluation Module

- Validation: Use cross-validation techniques to assess model performance.
- Performance Metrics: Evaluate using metrics such as R^2 score, RMSE, MAE, and RMAE.

7. Deployment Module

- Integration: Integrate the model into an application or system for end-users.
- API Development: Create APIs for model prediction services.

8. Monitoring and Maintenance Module

- Model Performance Monitoring: Continuously track the model’s performance over time.
- Retraining: Implement a strategy for retraining the model with new data.

9. User Interface Module

- Dashboard: Develop a user-friendly interface for users to input data and view predictions.
- Visualization: Provide insights and visualizations for users to understand salary predictions.

10. Documentation Module

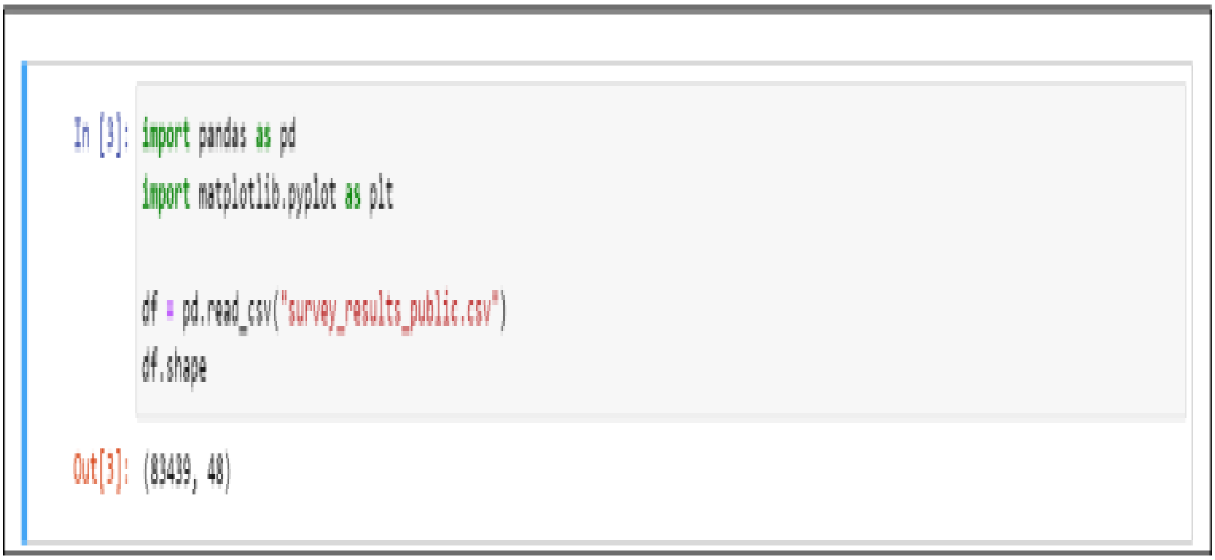
- User Guides: Create documentation for users and developers.
- Technical Documentation: Maintain detailed records of the development process, algorithms used, and data sources.

IMPLEMENTATION OF THE MODEL

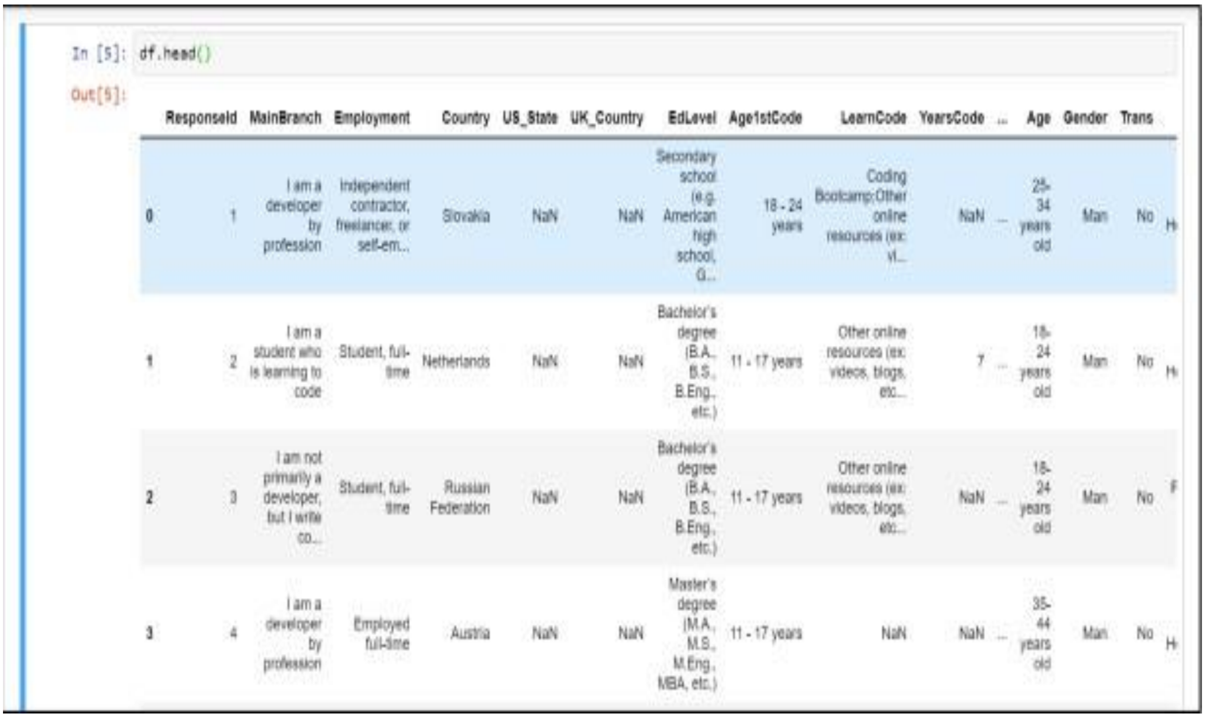
The Data collection is the first real step towards the real development of a machine learning model, collecting data. This is a critical step that will cascade in how good the model will be the more and better data that we get, the better our model will perform. Our data set named “Survey_results_public" is a raw dataset. It means that a lot of preprocessing is required so that it becomes useful for evaluation. Our dataset consists of 83439 rows and 48 features that will help us to predict the sales of the house and is a fairly big dataset.

Loading the Data: We load the dataset into our notebook using the panda’s data frame Data Pre-Processing: Our Next step is to convert our data set into the best possible format so that we can extract what all features are required to predict the price of the house. This is where all cleaning of our data takes place, be it treating the missing values, treating repetitive values, or addition of different features according to our needs. Once they are identified, there are several ways to deal with them:

Eliminating the samples or features with missing values. (we risk to delete relevant information or too many samples).Imputing the missing values, with some pre-built estimators such as the Imputer class from learn.

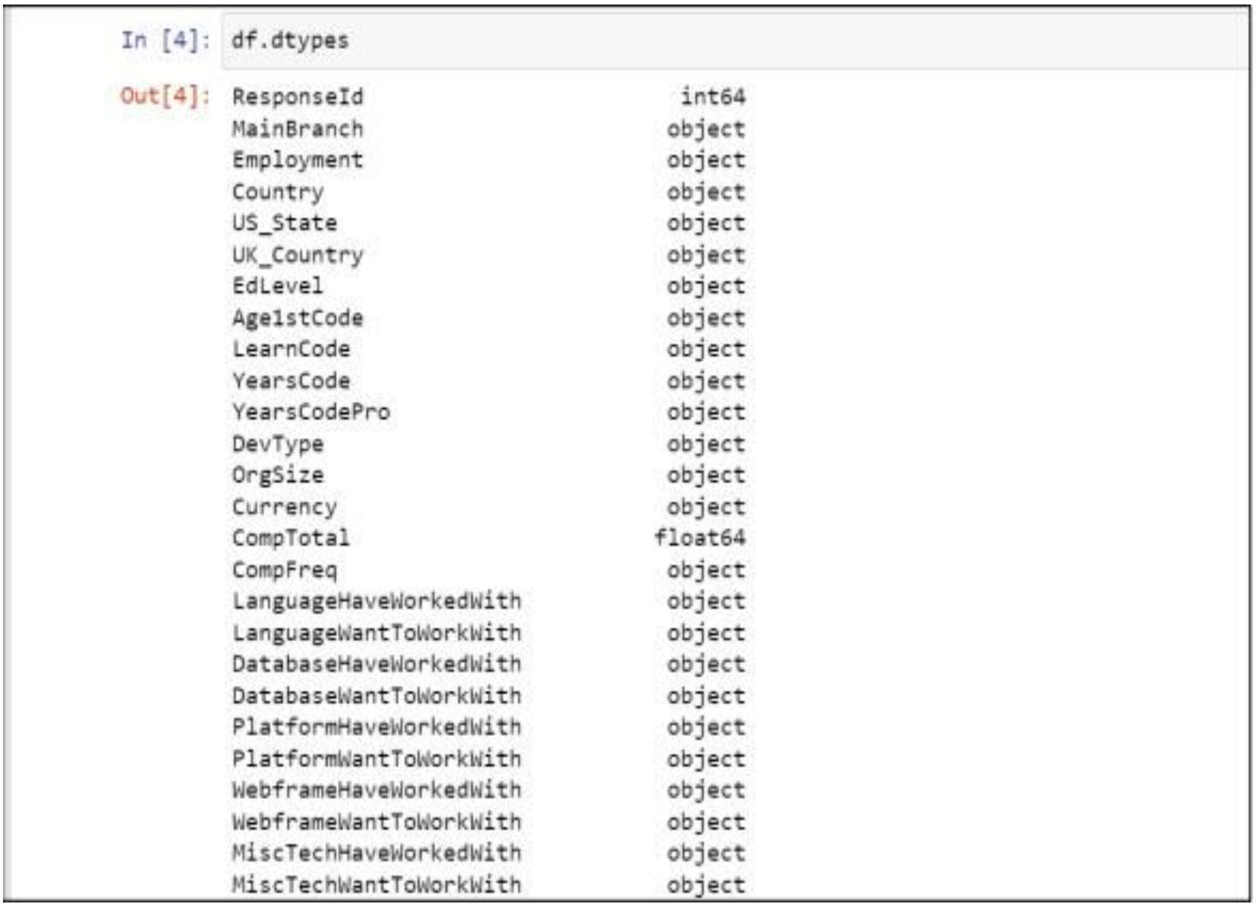


```
In [3]: import pandas as pd
        import matplotlib.pyplot as plt

        df = pd.read_csv("survey_results_public.csv")
        df.shape

Out[3]: (83439, 48)
```



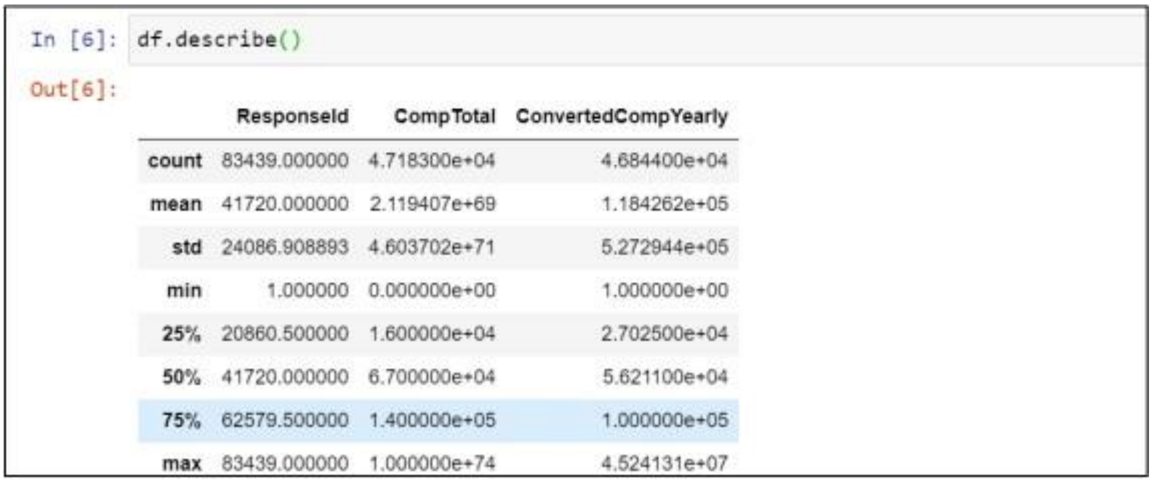
```
In [5]: df.head()
Out[5]:
```

	ResponseId	MainBranch	Employment	Country	US_State	UK_Country	EdLevel	Age1stCode	LearnCode	YearsCode	...	Age	Gender	Trans	
0	1	I am a developer by profession	Independent contractor, freelancer, or self-em...	Slovakia	NaN	NaN	Secondary school (e.g. American high school, G...	18 - 24 years	Coding Bootcamp;Other online resources (ex: vi...	NaN	...	25-34 years old	Man	No	H
1	2	I am a student who is learning to code	Student, full-time	Netherlands	NaN	NaN	Bachelor's degree (B.A., B.S., B.Eng., etc.)	11 - 17 years	Other online resources (ex: videos, blogs, etc...	7	...	18-24 years old	Man	No	H
2	3	I am not primarily a developer, but I write co...	Student, full-time	Russian Federation	NaN	NaN	Bachelor's degree (B.A., B.S., B.Eng., etc.)	11 - 17 years	Other online resources (ex: videos, blogs, etc...	NaN	...	18-24 years old	Man	No	F
3	4	I am a developer by profession	Employed full-time	Austria	NaN	NaN	Master's degree (M.A., M.S., M.Eng., MBA, etc.)	11 - 17 years	NaN	NaN	...	35-44 years old	Man	No	H



```
In [4]:  df.dtypes

Out[4]:  ResponseId                       int64
         MainBranch                      object
         Employment                      object
         Country                         object
         US_State                        object
         UK_Country                      object
         EdLevel                         object
         Age1stCode                      object
         LearnCode                       object
         YearsCode                       object
         YearsCodePro                    object
         DevType                         object
         OrgSize                         object
         Currency                        object
         CompTotal                      float64
         CompFreq                        object
         LanguageHaveWorkedWith          object
         LanguageWantToWorkWith          object
         DatabaseHaveWorkedWith          object
         DatabaseWantToWorkWith          object
         PlatformHaveWorkedWith          object
         PlatformWantToWorkWith          object
         WebframeHaveWorkedWith          object
         WebframeWantToWorkWith          object
         MiscTechHaveWorkedWith          object
         MiscTechWantToWorkWith          object
```

Data Exploration: Further, we explore our data as much as possible to know the features very standard deviation, min and well. We get to know the count of each feature, their mean values, Smax value etc.

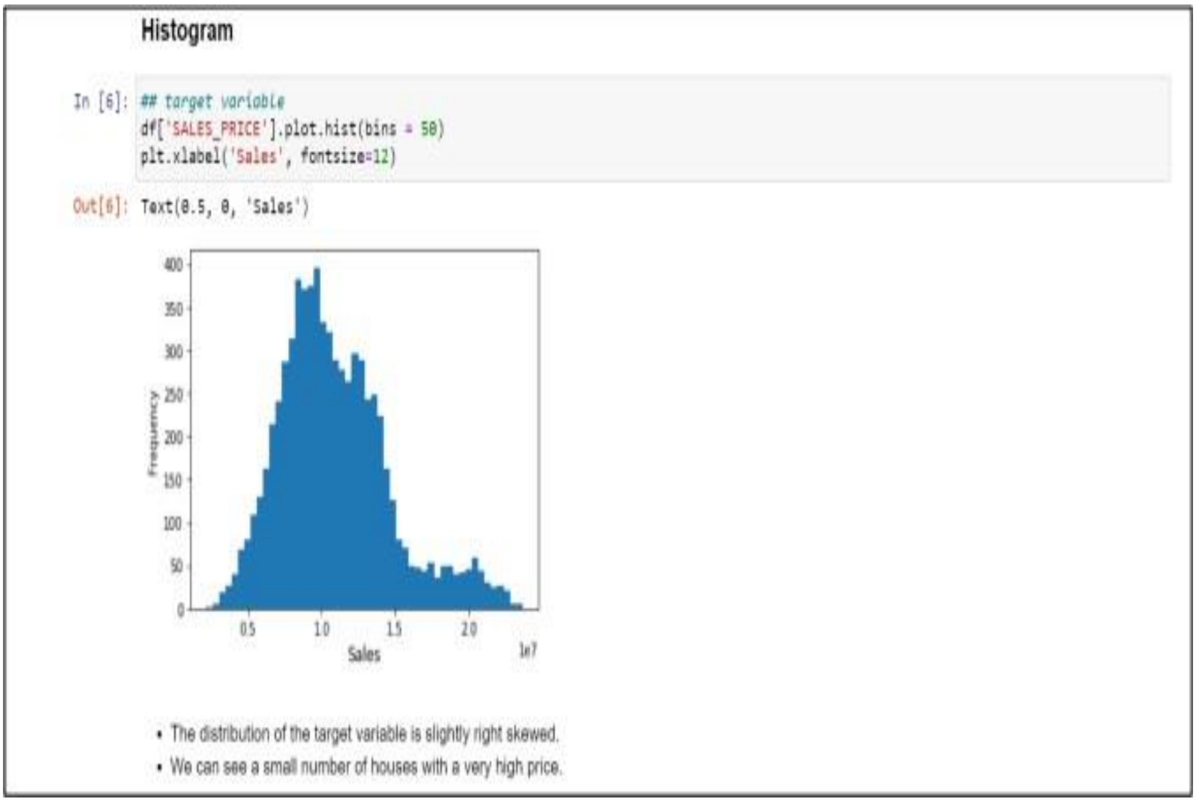


```
In [6]: df.describe()

Out[6]:
```

	ResponseId	CompTotal	ConvertedCompYearly
count	83439.000000	4.718300e+04	4.684400e+04
mean	41720.000000	2.119407e+69	1.184262e+05
std	24086.908893	4.603702e+71	5.272944e+05
min	1.000000	0.000000e+00	1.000000e+00
25%	20860.500000	1.600000e+04	2.702500e+04
50%	41720.000000	6.700000e+04	5.621100e+04
75%	62579.500000	1.400000e+05	1.000000e+05
max	83439.000000	1.000000e+74	4.524131e+07

univariate Analysis:

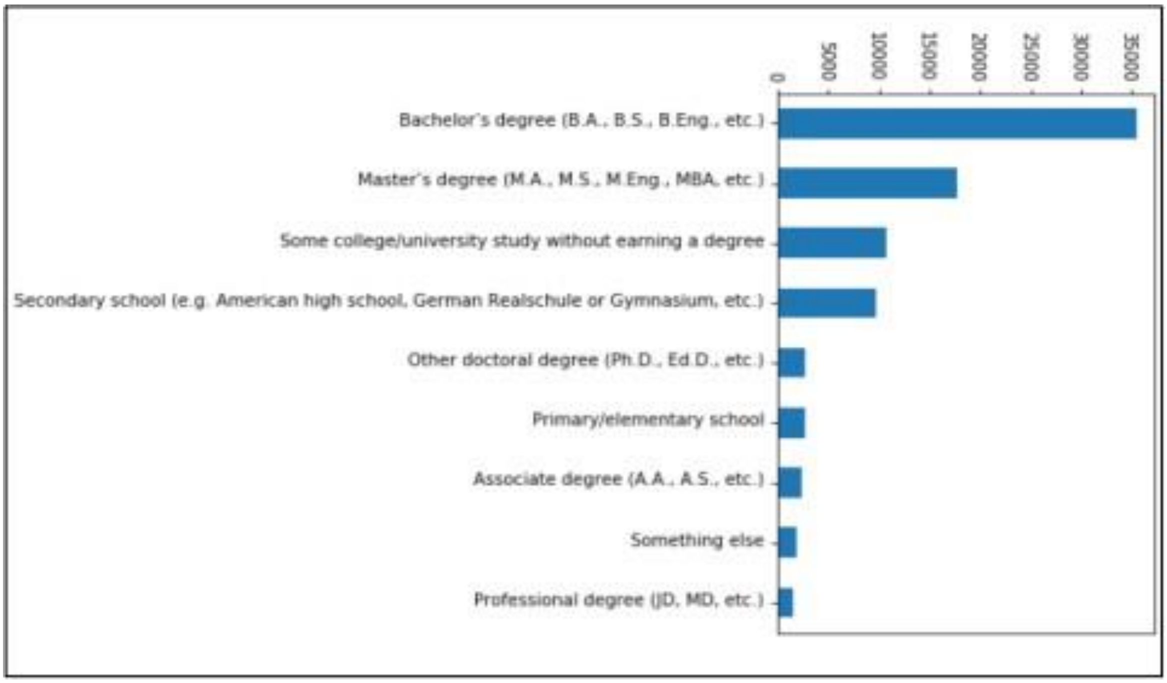
Uni means one and variate means variable, so in univariate analysis, there is only one dependable variable. The objective of univariate analysis is to derive the data, define and summarize It, and analyze the pattern present in it. In a dataset, it explores each variable separately.

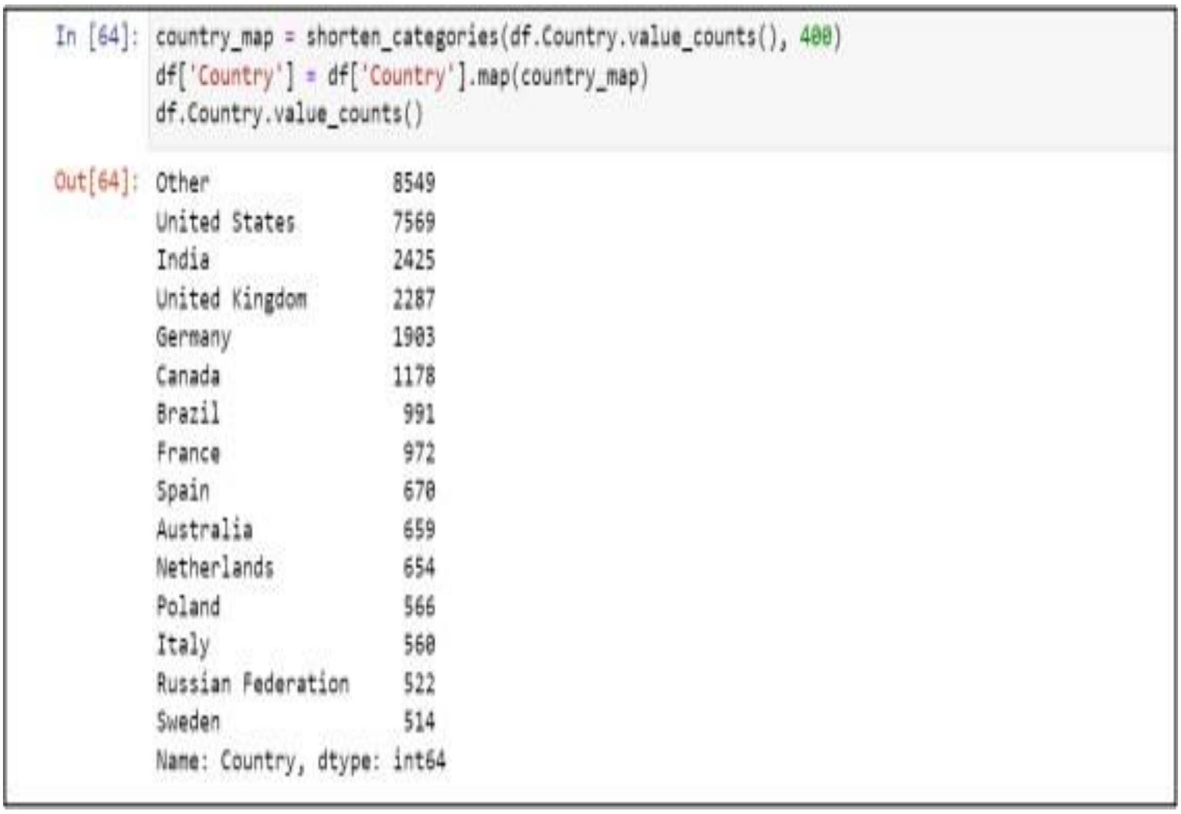


Histogram

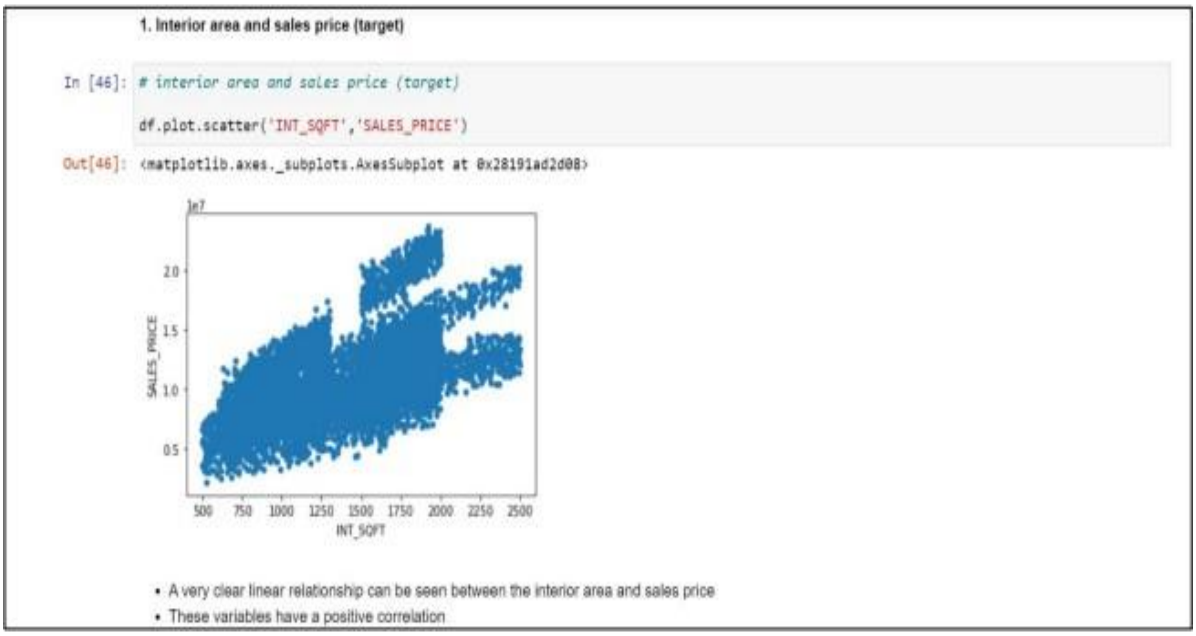
```
In [6]: ## target variable
        df['SALES_PRICE'].plot.hist(bins = 50)
        plt.xlabel('Sales', fontsize=12)

Out[6]: Text(0.5, 0, 'Sales')
```

- The distribution of the target variable is slightly right skewed.
- We can see a small number of houses with a very high price.





```
In [64]: country_map = shorten_categories(df.Country.value_counts(), 400)
         df['Country'] = df['Country'].map(country_map)
         df.Country.value_counts()

Out[64]: Other                 8549
         United States         7569
         India                 2425
         United Kingdom        2287
         Germany               1903
         Canada                1178
         Brazil                 991
         France                 972
         Spain                  670
         Australia              659
         Netherlands            654
         Poland                 566
         Italy                  560
         Russian Federation     522
         Sweden                 514
         Name: Country, dtype: int64
```

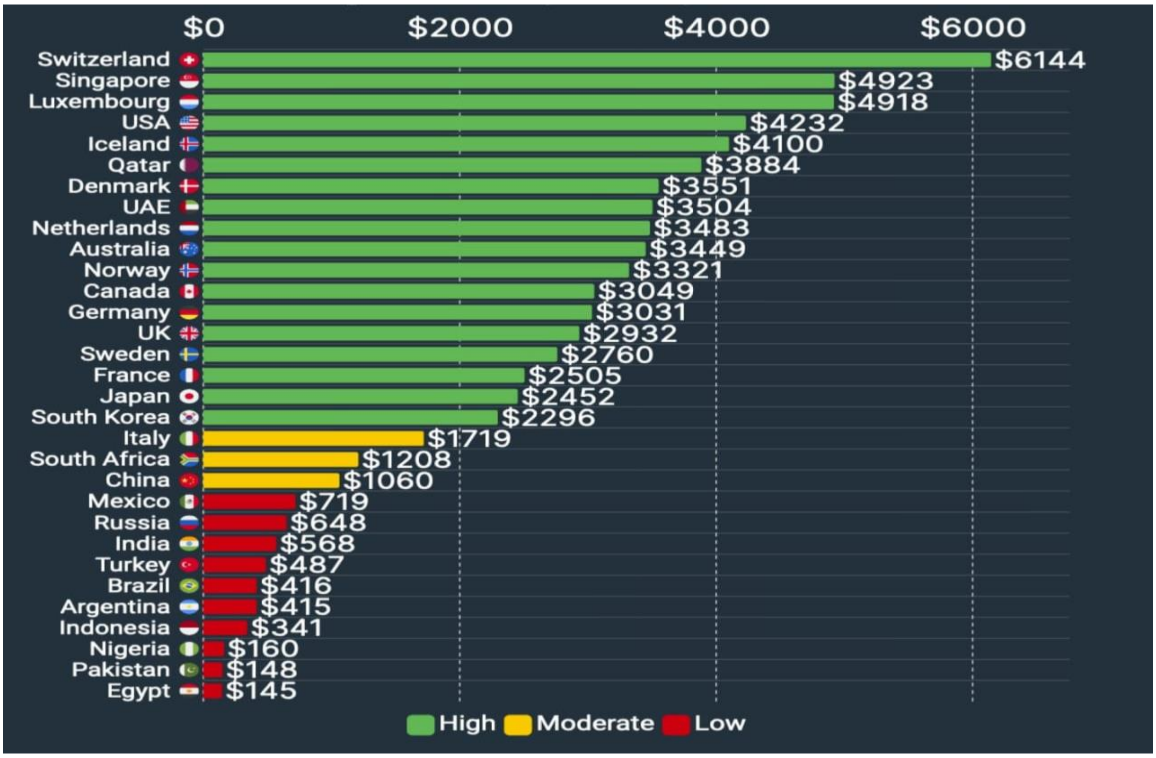


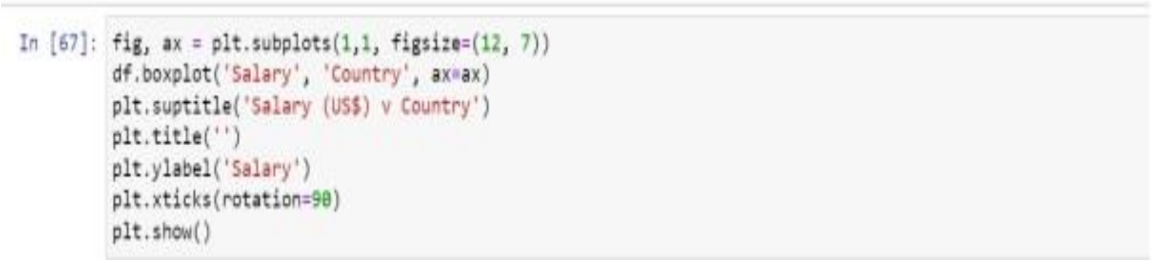
1. Interior area and sales price (target)

```
In [46]: # interior area and sales price (target)

         df.plot.scatter('INT_SQFT','SALES_PRICE')
```

```
Out[46]: <matplotlib.axes._subplots.AxesSubplot at 0x28191ad2d08>
```

- A very clear linear relationship can be seen between the interior area and sales price
- These variables have a positive correlation





```
In [67]: fig, ax = plt.subplots(1,1, figsize=(12, 7))
         df.boxplot('Salary', 'Country', ax=ax)
         plt.suptitle('Salary (US$) v Country')
         plt.title('')
         plt.ylabel('Salary')
         plt.xticks(rotation=90)
         plt.show()
```

feature Scaling:

This is a crucial step in the preprocessing phase as the majority of Chinese learning algorithms perform much better when dealing with features that are on the same scale. The most common techniques are:

Normalization: it refers to rescaling the features to a range of [0,1], which is a special case of min-max scaling. To normalize our data we'll simply need to apply the min-max scaling method to each feature column.

1. **Standardization**: it consists in centering the feature columns at mean 0 with standard deviation 1. so that the feature columns have the same parameters as a standard normal distribution (zero mean and unit variance).

This makes it much easier for the learning algorithms to learn the weights of the parameters. In addition, it keeps useful information about outliers and makes the algorithms less sensitive to them.

2.Implementing the Model: Here comes the part where the actual machine learning algorithms are being implemented. As stated above, we are using Linear Regression Machine Learning Algorithm to predict the house price under the Chennai house price Prediction model.

3.Segregating Dependent and Independent Variables: Independent variables (also referred to as Features) are the input for a process that is being analyzed. Dependent variables are the output of the process.

4.Splitting the Data Set into Train and Test Dataset: We will split our data in three parts: training, testing and validating sets. We train our model with training data, evaluate it on validation data and finally, once it is ready to use, test it one last time on test data. The ultimate goal is that the model can generalize well on unseen data, in other words, predict accurate results from new data, based on its internal parameters adjusted while it was trained and validated.

Implementing Linear Regression:

Learning Phase: In Linear regression we are given a number of predictor variables and a continuous response variable, and we try to find a relationship between those variables that allows us to predict a continuous outcome.

SYSTEM SECURITY

The protection of computer-based resources that include hardware, software, data, procedures and people against unauthorized use or natural Disaster is known as System Security system security can be divided into four related issues:

- Security
- Integrity
- Privacy
- Confidentiality

SYSTEM SECURITY: refers to the technical innovations and procedures applied to the hardware and operation systems to protect against deliberate or accidental damage from a defined threat.

DATA SECURITY: is the protection of data from loss, disclosure, modification and destruction.

SYSTEM INTEGRITY: refers to the power functioning of hardware and programs, | security and safety against external threats such appropriate physical as eavesdropping and wiretapping.

PRIVACY: defines the rights of the user or organizations to determine what information they are willing to share with or n can be protected except from others and how the organization against unwelcome, unfair or excessive dissemination of information about it.

CONFIDENTIALITY: is a special status given to sensitive information in a database to minimize the possible invasion of privacy. It is an attribute of information that characterizes its need for protection.

4.2 SOFTWARE ENVIRONMENT

USED LANGUAGES

In the implementation of the proposed system, we used python as a programming language. Python is a beginner's language, which provides various applications. In recent years, python has set the new trend because it is an easy to use, interpreted, object-oriented, high-level, scripting language. Python is one of the best languages for implementing machine learning. It provides rich packages and libraries that are used in machine learning.

4.2.1 PYTHON

Below are some facts about Python.

Python is currently the most widely used multi-purpose, high-level programming language.

Python allows programming in Object-Oriented and Procedural paradigms.

Python

programs generally are smaller than other programming languages like Java.

Programmers have to type relatively less and the indentation requirement of the language makes them readable all the time.

Python language is being used by almost all tech-giant companies like — Google, Amazon, Facebook, Instagram, Dropbox, Uber... etc.

The biggest strength of Python is huge collection of standard library which can used for the following —

- Machine Learning
- GUI Applications (like Kivy, Tkinter, PyQt5 etc.)
- Web frameworks like Django (used by YouTube, Instagram, Dropbox)
- Image processing (like Opencv, Pillow)
- Web scraping (like Scrapy, BeautifulSoup, Selenium)
- Test frameworks
- Multimedia

History of python

The programming language Python was conceived in the late 1980s,[1] and its implementation was started in December 1989[2] by Guido van Rossum at CWI in the Netherlands as a successor to ABC capable of exception handling and interfacing with the Amoeba operating system.[3] Van Rossum is Python's principal author, and his continuing central role in deciding the direction of Python is reflected in the title given to him by the Python community, Benevolent Dictator for Life (BDFL).[4][5] (However, Van Rossum stepped down as leader on July 12, 2018.[6]). Python was named after the BBC TV show Monty Python's Flying Circus.[7]

Python 2.0 was released on October 16, 2000, with many major new features, including a cycle- detecting garbage collector (in addition to reference counting) for memory management and support for Unicode. However, the most important change was to the development process itself, with a shift to a more transparent and community-backed process.[8]

Python 3.0, a major, backwards-incompatible release, was released on December 3, 2008[9] after a long period of testing. Many of its major features have also been backported to the backwards-compatible, though now-unsupported, Python 2.6 and 2.7.

4.2.2 PYTHON FEATURES

1. Free and Open Source: Python language is freely available at the official website and you can download it from the given download link below click on the Download Python keyword. Download Python Since it is open-source, this means that source code is also available to the public. So you can download it, use it as well as share it.

2. Easy to code: Python is a high-level programming language. Python is very easy to learn the language as compared to other languages like C, C#, JavaScript, Java, etc. It is very easy to code in the Python language and anybody can learn Python basics in a few hours or days. It is also a developer-friendly language.

3. Easy to Read: As you will see, learning Python is quite simple. As was already established, Python’s syntax is really straightforward. The code block is defined by the indentations rather than by semicolons or brackets.

4. Object-Oriented Language: One of the key features of Python is Object-Oriented programming. Python supports object-oriented language and concepts of classes, object encapsulation, etc.

5. GUI Programming Support: Graphical User interfaces can be made using a module such as PyQt5, PyQt4, wxPython, or Tk in Python. PyQt5 is the most popular option for creating graphical apps with Python.

6. High-Level Language: Python is a high-level language. When we write programs in Python, we do not need to remember the system architecture, nor do we need to manage the memory.

7. **Large Community Support:** Python has gained popularity over the years. Our questions are constantly answered by the enormous Stack Overflow community. These websites have already provided answers to many questions about Python, so Python users can consult them as needed.

8. Easy to Debug: Excellent information for mistake tracing. You will be able to quickly identify and correct the majority of your program's issues once you understand how to interpret Python's error traces. Simply by glancing at the code, you can determine what it is designed to perform.

Advantages Of Python Over Other Programming Languages

Python is a language embraced by a large community of coders for its many advantages and features. Many businesses choose python as the main programming language. Let's find out these **advantages of Python** and the reasons why most developers love Python?

Python is a Simple Language

Python is an easily readable and simple to understand language for developers who have never written a code in it. As a result, the community of Python users are continuously evolving and growing. There are many scholars and professors among the Python users' community. So that when a problem occurs, the developer can focus on it and take help from others in the community without any worries about language complexity.

Python is Free

Python is a programming language which is free of charge and open. The OSI- approved open source license under which Python is developed makes it a language free to use and distribute, including for commercial purposes. It will reduce your cost for maintenance. While the developers can share, copy, and change it. As for the Python community, it provides an opportunity to share knowledge with junior specialists.

Easy to Use

Programmers say that Python is easy to use. Although when constructing mobile applications or games C++ or any other typical scripting language might be easier to use, Python is better for easily building server-side applications, automating build systems, and collecting test data.

Compatible with Various Platforms

Python is highly compatible and offers compatibility with various platforms. It is one of the major issues developers commonly face when they use other languages. The well supported platforms on Python 3.7 and 2.7 include:

- Linux
- Windows Vista and newer for Python 3.7, Windows XP and newer for Python 2.7
- FreeBSD 10 and newer
- macOS Snow Leopard (macOS 10.6, 2008) and newer
- This feature of Python makes it a favorable language for most users.

Object-Oriented

Python supports object-oriented programming, and it is procedure-oriented. Object-Oriented Programming, in the sense, utilizes objects that are based on data and functionality. The Procedure Oriented feature offers to apply reusable pieces of code.

NUMPY

In Python we have lists that serve the purpose of arrays, but they are slow to process. NumPy stands for Numerical Python. The array object in NumPy is called ndarray, it provides a lot of supporting functions that make working with ndarray very easy.

PANDAS

Pandas is a Python library used for working with data sets. It has functions for analyzing, cleaning, exploring, and manipulating data. Pandas allows us to analyze big data and make conclusions based on statistical theories. Pandas can clean messy data sets, and make them readable and relevant. Relevant data is very important in data science.

Keras Tutorial

Keras is an open-source high-level Neural Network library, which is written in Python and is capable enough to run on Theano, TensorFlow, or CNTK. It was developed by one of the Google engineers, Francois Chollet. It is made user- friendly, extensible, and modular for facilitating faster experimentation with deep neural networks. It not only supports Convolutional Networks and Recurrent Networks individually but also their combination.

MySQL

MySQL is an open-source RDBMS developed by MySQL AB and owned by Oracle Corporation. It is a combination of two words – ‘My’ and ‘SQL” A relational database organizes data into one or more data tables in which data may be related to each other; these relations help structure the data. MySQL is written in C and C++ programming languages. It supports Linux, Solaris, macOS, Windows and FreeBSD operating systems.

MODULES

Dataset

Dataset is essentially the backbone for all operations, techniques or models used by developers to interpret them. Datasets involve a large amount of data points grouped into one table**.** Datasets are used in almost all industries today for various reasons. In this day and age, to train the younger generation to interact effectively with Datasets, many Universities publicly release their Datasets for example UCI and websites like Kaggle and even GitHub release datasets which developers can work with to get the necessary outputs.

DATA PREPROCESSING

Data preprocessing is a technique that is used to convert the raw data into a clean data set. It is the first and crucial step while creating a machine learning model. The major goal of data preprocessing is to eliminate data issues such as missing values, improve data quality, and make the data useful for machine learning purposes.

FEATURE SELECTION

Feature selection methods are used to remove unnecessary, irrelevant, and redundant attributes from a dataset that do not contribute to the accuracy of a predictive model or which might reduce the accuracy of the model. It is a process of automatically or manually selecting the subset of most appropriate and relevant features to be used in model building." Feature selection is performed by either including the important features or excluding the irrelevant features in the dataset without changing them.

FEATURE IMPORTANCE

In machine learning, feature importance scores are used to determine the relative importance of each feature in a dataset when building a predictive model. These scores are calculated using a variety of techniques, such as decision trees, random forests, linear models. It reduces the number of input features. In this paper, feature importance is implemented using an extra tree classifier from the decision tree. Understand how a model works internally, Identify features that can be removed to improve accuracy.

4.2.3 SAMPLE CODE

```
import pandas as pd
data = pd.read_csv('data.csv')
print("First 5 rows of the dataset:")
print(data.head())
print("\nSummary statistics:")
print(data.describe())
print("\nMissing values in each column:")
print(data.isnull().sum())
data['Salary'].fillna(data['Salary'].mean(), inplace=True)
age_filtered_data = data[data['Age'] > 30]
print("\nFiltered data (Age > 30):")
print(age_filtered_data)
avg_salary_by_gender = data.groupby('Gender')['Salary'].mean()
print("\nAverage salary by gender:")
print(avg_salary_by_gender)
sorted_data = data.sort_values(by='Salary', ascending=False)
print("\nData sorted by Salary (descending order):")
print(sorted_data.head())
data.to_csv('cleaned_data.csv', index=False)
```

Step 1: Install Required Libraries

```
pip install pandas numpy scikit-learn matplotlib
```

Step 2: Create a Sample Dataset

```
import pandas as pd
import numpy as np
# Creating a synthetic dataset
np.random.seed(0)
```

```
data_size = 100
data = {
    'YearsExperience': np.random.uniform(1, 10, data_size),  # Random years of experience
    'EducationLevel': np.random.randint(1, 4, data_size),    # Education level: 1 (Bachelors), 2 (Masters), 3 (PhD)
    'Salary': np.random.uniform(30000, 120000, data_size)     # Random salaries
}
df = pd.DataFrame(data)
# Display the first few rows of the dataset
print(df.head())
```

Step 3: Prepare the Data

```
from sklearn.model_selection import train_test_split
# Features and target variable
X = df[['YearsExperience', 'EducationLevel']]
y = df['Salary']
# Split the dataset into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=0)
```

Step 4: Train a Machine Learning Model

```
from sklearn.linear_model import LinearRegression
# Create and train the model
model = LinearRegression()
model.fit(X_train, y_train)
```

Step 5: Make Predictions

```
# Make predictions on the test set
predictions = model.predict(X_test)
# Display predictions
print("Predicted Salaries:", predictions)
```

Step 6: Evaluate the Model

```
from sklearn.metrics import mean_squared_error, r2_score
# Calculate performance metrics
mse = mean_squared_error(y_test, predictions)
r2 = r2_score(y_test, predictions)
print("Mean Squared Error:", mse)
print("R^2 Score:", r2)
```

Step 7: Visualize the Results

```
import matplotlib.pyplot as plt
# Plotting actual vs predicted salaries
plt.scatter(y_test, predictions)
plt.xlabel('Actual Salaries')
plt.ylabel('Predicted Salaries')
plt.title('Actual vs Predicted Salaries')
plt.plot([min(y_test), max(y_test)], [min(y_test), max(y_test)], color='red', linestyle='--')  # Diagonal line
plt.show()
```

ALGORITHMS

5. ALGORITHMS

List of algorithm used in implementation of our experiment are:

- support Vector Machine
- K-nearest algorithm
- Artificial Neural Network
- Random forest algorithm
- Logistic Regression

5.1Support Vector Machine:

pseudocode:

- Importing the necessary packages Example: import pandas as pd
- def SVM

Step 1: START

Step 2: Reading the dataset. pd.read.csv (file name) # reads the dataset file

Step 3: Data cleaning and preprocessing of data Resampling the data as normal and fraud class i.e. normal = 0 and fraud =1 under

- Under sampling of data is done * Data is scaled (if any null value then eliminated) and normalized. + Dataset is splitted into two sets as train data and test data using split () on training data is used to split the data.

Step 4: Training the data using the SVM algorithm

- SVM classifier is called as classifier. predict () # which predicts whether transaction fraud or non fraud.

Step 5: Calculating the fraud transactions and valid transactions, then calculating the recall, precision and accuracy and stored in the respective locations

Step 6: STOP

5.2 K-Nearest Neighbour

Pseudo code

step1: START

step 2 :Loading of dataset pd.read.csy (csv file) # reads the file and loads

step 3 :Cleaning and normalization of data

- Normal=0
- Fraud=1# resampling + Data is scaled and normalized
- Train_test_split() # splitting of dataset into train and test data

Step 4: Train the model then fit the trained model

```
    • Trained the data using Knn classifier
    • K-NeighborsClassifier() classifier which does classification of transactions
Step 5: Calculating the number of fraud, valid transactions and recall, precision and accuracy calculated.
Step 6: STOP
```

5.3Artificial Neural Network (ANN):

Pseudocode:

The ANN algorithm has two parts: Training part and testing part. Training part: Def ANN:

```
Step 1: START
Step2: Loading and observing the dataset
    • pd.read.csv(.csv) # reads the dataset
    • resampling of data
    • StandardScaler() #scaling and normalization of data
Step 3: Data pre-processing
        Train_test_split() #Splitting of data
Step 4: Training the model
         Dense() #Adding data to activation function
Step 5: Analyzing the model «+ Prediction of fraud is made and this trained data is stored .     it can used to test (training the model takes longer time so it is stored)
Step 6: STOP
```

Testing part

Def ANN is carried out in a similar way; the only difference is that the stored trained model is used to test the data and classify it.

Random forest tree algorithm:

Random Forest algorithm is a powerful tree learning technique in Machine Learning. It works by creating a number of Decision Trees during the training phase. Each tree is constructed using a random subset of the data set to measure a random subset of features in each partition. This randomness introduces variability among individual trees, reducing the risk of overfitting and improving overall prediction performance.. For regression tasks, the mean or average prediction of the individual trees is returned. Random decision forests correct for decision trees' habit of overfitting to their training set. Random forests generally outperform decision trees, but their accuracy is lower than gradient boosted trees. However, data characteristics can affect their performance.

The first algorithm for random decision forests was created in 1995 by Tin Kam Ho[l] using the random subspace method, which, in Ho's formulation, is a way to implement the "stochastic discrimination" approach to classification proposed by Eugene Kleinberg.

An extension of the algorithm was developed by Leo Breiman and Adele Cutler, who registered "Random Forests" as a trademark in 2006 (as of 2019, owned by Minitab, Inc.). The extension combines Breiman's "bagging" idea and random selection of features, introduced first by Ho [1] and later independently by Amit and Geman ([13] in order to construct a collection of decision trees with controlled variance Random forests are frequently used as "black box" models businesses, as they generate reasonable predictions across a wide range of data while requiring little configuration.

5.4 Logistic Regression Algorithm

Logistic regression is the appropriate regression analysis to conduct when the dependent variable is dichotomous (binary).

- Like all regression analyses, logistic regression is a predictive analysis. Logistic regression is used to describe data and to explain the relationship between one dependent binary variable and one or more nominal, ordinal, interval or ratio-level independent variables,
- By using sigmoid function it calculates the probability and it compares that probability to the threshold probability and predicts the class of the observation

Logistic egression is a powerful statistical method commonly used in machine learning for binary classification problems. While salary prediction is typically a regression task (predicting continuous values), logistic regression can be applied in specific scenarios related to salary, such as predicting whether an employee will earn above or below a certain salary threshold. Here’s how logistic regression can be utilized in the context of employee salary prediction:

1. Problem Definition

- Binary Classification: Instead of predicting exact salaries, you can frame the problem as predicting whether an employee's salary is above or below a defined threshold (e.g., $70,000).
- Outcome Variable: The target variable is binary, where 1 indicates salaries above the threshold and 0 indicates salaries below.

2. Data Preparation

- Feature Selection: Identify relevant features that may influence salary, such as:
- Years of experience
- Education level
- Job role

- Location
- Performance ratings
- Data Encoding: Convert categorical variables (e.g., job role, education level) into numerical format using techniques like one-hot encoding.

3. Model Training

- Splitting Data: Divide the dataset into training and testing sets to evaluate model performance.
- Training the Model: Fit a logistic regression model using the training data. This involves estimating the coefficients for each feature that maximize the likelihood of the observed data.

4. Model Evaluation

- Prediction: Use the model to predict probabilities of the outcome for the test set.
- Thresholding: Convert predicted probabilities into binary outcomes using a threshold (commonly 0.5).
- Performance Metrics: Assess the model using metrics like accuracy, precision, recall, F1-score, and the ROC-AUC curve to understand its effectiveness.

5. Interpretability

- Coefficient Analysis: Logistic regression provides interpretable coefficients, allowing you to understand the influence of each feature on the salary outcome. Positive coefficients indicate a higher likelihood of earning above the threshold, while negative coefficients indicate the opposite.

6. Limitations and Considerations

- Linear Relationship: Logistic regression assumes a linear relationship between the log-odds of the outcome and the predictor variables. Non-linear relationships may require transformation or alternative algorithms.
- Imbalance in Classes: If the classes are imbalanced (e.g., many more employees earning below the threshold), consider techniques like resampling or using class weights to address this.

7. Extensions and Improvements

- Regularization: Implement techniques like L1 (Lasso) or L2 (Ridge) regularization to prevent overfitting, especially when dealing with a large number of features.
- Ensemble Methods: Combine logistic regression with other models in ensemble methods (e.g., stacking or boosting) to improve prediction accuracy.

SYSTEM TESTING AND SCREEN SHOTS

6. SYSTEM TESTING AND SCREEN SHOTS

6.1 TESINGS

6.1.1 UNIT TESTING

Unit tests are very low level and close to the source of an application.They consist in testing individual methods and functions of the classes, components, or modules used by your software. Unit tests are generally quite cheap to automate and can run very quickly by a continuous integration server.

6.1.2 SYSTEM TESTING

System testing for employee salary prediction involves verifying the functionality, performance, and reliability of a system that predicts the salaries of employees based on various factors such as job role, experience, education, and performance. Here are some steps involved in system testing for employee salary prediction

6.1.3 INTEGRATION TESTING

Integration testing is a type of software testing that verifies how different components or modules of a system work together as a whole. In the context of employee salary prediction, integration testing can be performed to ensure that the various components of the system, such as data input, data processing, and output generation, function correctly when integrated into a single system.

6.1.4 FUNCTIONAL TESTING

Functional testing is a type of software testing that verifies whether the software application fulfills its intended purpose and meets the specified requirements. In the context of employee salary prediction, functional testing can be performed to ensure that the software accurately predicts salaries based on the input data provided by the user.

6.1.5 WHITE BOX TESTING

This type of testing ensures that

- All independent paths have been exercised at least once
- All logical decisions have been exercised on their true and false sides
- All loops are executed at their boundaries and within their operational bounds All internal data structures have been exercised to assure their validity.

6.1.6 BLOCK BOX TESTING

Black-box testing is a type of software testing in which the tester is not concerned with the software’s internal knowledge or implementation details but rather focuses on validating the functionality based on the provided specifications or requirements.

6.1.7 ACCEPTANCE TESTING

Acceptance testing is a type of software testing that evaluates whether a software application meets the specified requirements and expectations of the end-users or stakeholders. In the context of employee salary prediction, acceptance testing ensures that the system accurately predicts salaries based on the input data provided by the users.

BASIC PATH TESTING

Established technique of flow graph with Cyclomatic complexity was used to derive test cases overall the functions. The main steps in deriving test cases were:

Use the design of the code and draw a correspondent flow graph.

Determine the Cyclomatic complexity of the resultant flow graph, using formula: V(G)=E-N+2 or y(G)=Pt! or

v(G) =Number Of Regions

Where V(G) is Cyclomatic complexity, is the number of edges,

N is the number of flow graph nodes; P is the number of predicate nodes.

Determine the basis of a set of linearly independent paths.

CONDITIONAL TESTING

In this part of the testing each of the conditions were tested to both true and false aspects. And all the resulting paths were tested. So that each path that may be generated on a particular condition is traced to uncover any possible errors.

DATA FLOW TESTING

This type of testing selects the path of the program according to the location of definition and use of variables. This kind of testing was used only when some local variables were declared. The definition-use chain method was used in this type of testing. These were particularly useful in nested statements

Loop Testing

In this type of software testing type that is performed to validate the loops. It is one type of Control Structure Testing. Loop testing is a white box testing technique and is used to test loops in the program.

Objectives of Loop Testing:

The objective of Loop Testing is:

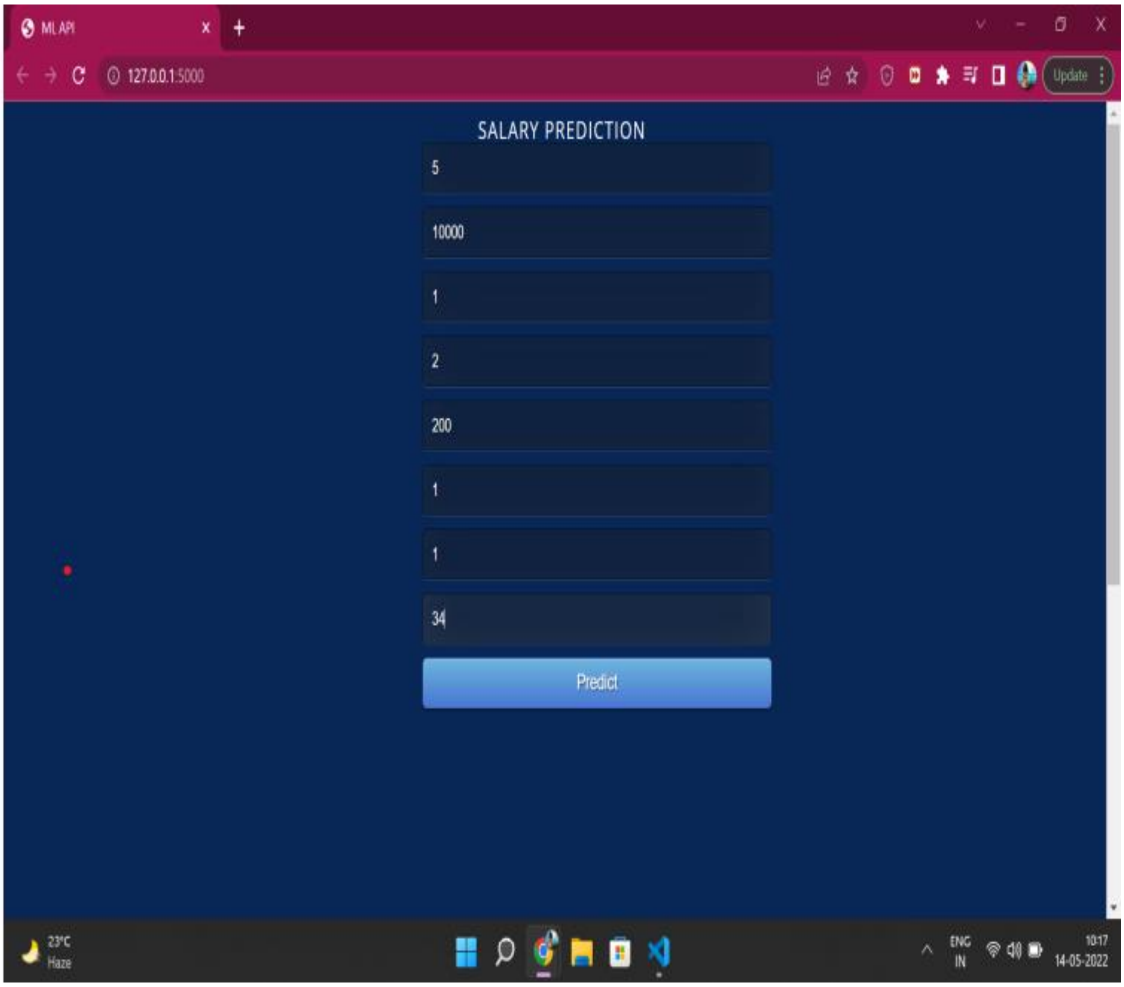
- To fix the infinite loop repetition problem.
- To know the performance.

6.2 TEST CASE

TCID	Condition Being Tested	Expected Result	Result
1	Check if dataset is taken as input	If taken process next step else throw error	Passed
2	Check for null values in the dataset	If null values drop the null records	Passed
3	Check for future extraction and save the file in .pkl format	If file not saved throw the error	Passed

6.3 SCREEN SHOTS

HOME SCREEN



SALARY PREDICTION

5

10000

1

2

200

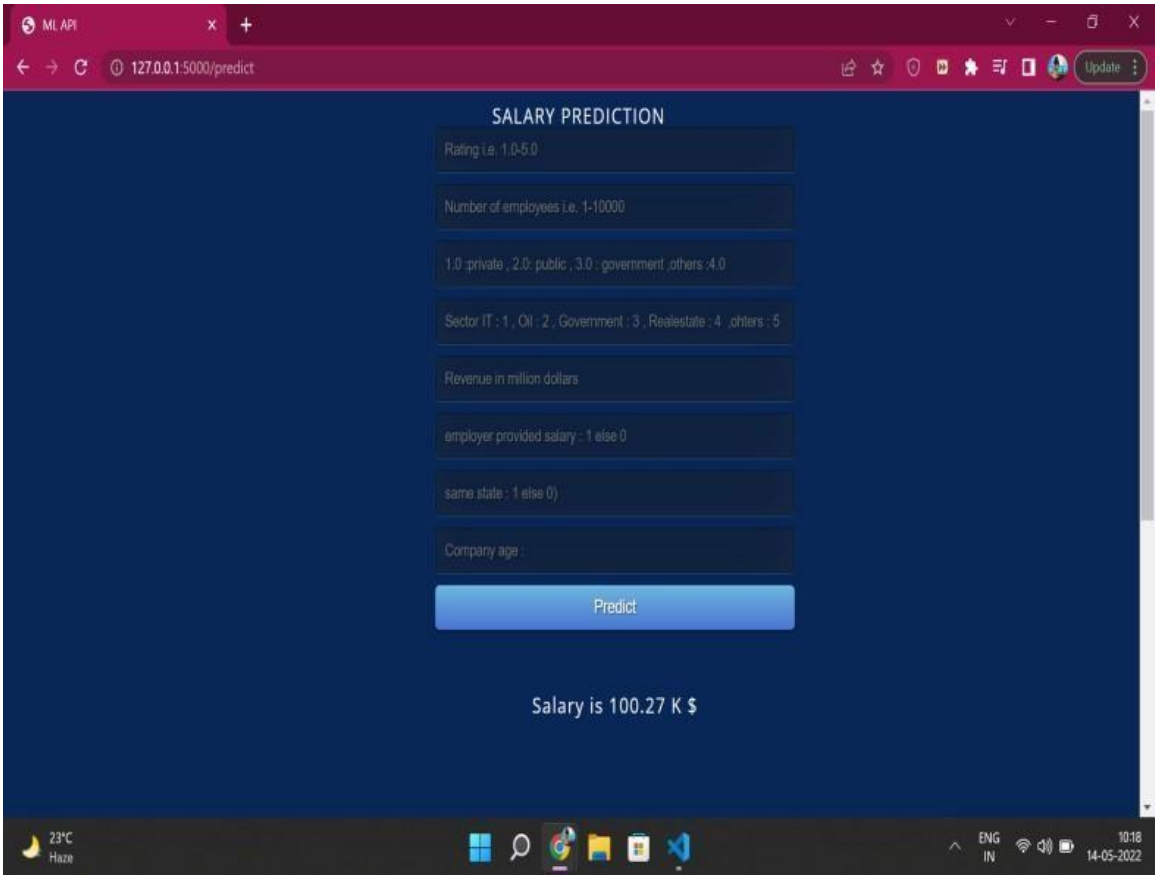
1

1

34

Predict

OUTPUT


ML API
127.0.0.1:5000/predict
Update
Rating i.e. 1.0-5.0
Number of employees i.e. 1-10000
1.0 :private , 2.0: public , 3.0 : government ,others :4.0
Sector IT : 1 , Oil : 2 , Government : 3 , Realestate : 4 ,ohters : 5
Revenue in million dollars
employer provided salary : 1 else 0
same state : 1 else 0)
Company age :
Predict
23°C
Haze
ENG
IN
10:18
14-05-2022

SALARY PREDICTION

Salary is 100.27 K $

CONCLUSION AND FUTURE SCOPE

7.1FUTURE SCOPE:

Since nothing in this universe can be termed as “perfect”, thus a lot of features can be added to make the system more widely acceptable and more user friendly. This will not only help to predict rates of other areas in the city but also will be more user beneficial. In the upcoming phase of our project we will be able to connect an even larger dataset to this model so that the training can be even better. This model should check for new data, once in a month, and incorporate them to expand the dataset and produce better results.

We can try out other dimensionality reduction techniques like Univariate Feature Selection and Recursive feature elimination in the initial stages. Another major future scope that can be added is providing the model with an estate database of more cities which will provide the user to explore more graduates and reach an accurate decision. More factors like training period that affect the job salary of a graduate shall be added. In-depth details of every individual will be added to provide ample details of a desired estate. This will help the system to run on a larger level.

Employee salary prediction using machine learning has significant potential and relevance in today's data-driven workplace. Here are some future scopes and considerations in this area:

1. Improved Accuracy with Advanced Models

- Deep Learning Techniques: Utilizing neural networks can enhance the accuracy of predictions by capturing complex relationships in data.
- Ensemble Methods: Combining multiple models can lead to better performance and reduce overfitting.

2. Real-Time Salary Insights

- Dynamic Modeling: Developing models that adapt to changing market conditions, economic factors, or company performance metrics in real time.

3. Incorporating External Data Sources

- Market Trends: Integrating data from job market trends, industry reports, and economic indicators can provide context for salary predictions.
- Geographical Variations: Accounting for location-specific factors, such as cost of living and regional demand for skills.

4. Fairness and Bias Mitigation

- Bias Detection: Implementing techniques to identify and mitigate bias in salary predictions, ensuring fair compensation practices.
- Transparency: Developing explainable AI models that clarify how salary predictions are made, which can foster trust and accountability.

5. Employee Performance and Development

- Linking Performance Data: Analyzing how employee performance metrics relate to salary progression can help in creating more informed compensation strategies.
- Career Pathing: Using predictive models to help employees understand potential salary trajectories based on career development choices.

6. Integration with HR Systems

- Automated Decision Support: Integrating salary prediction models into HR software for automated compensation adjustments and forecasting.
- Strategic Workforce Planning: Utilizing predictions to inform hiring strategies and workforce allocation.

7. Customization for Organizations

- Tailored Solutions: Developing models that are specific to an organization's unique structure, culture, and compensation philosophy.
- Role-Specific Predictions: Focusing on niche roles or industries that may require specialized salary forecasting.

8. Ethical Considerations

- Privacy and Data Security: Ensuring that the data used for predictions adheres to privacy regulations and ethical standards.
- Employee Consent: Involving employees in data usage discussions to maintain transparency and build trust.

9. Continuous Learning Models

- Feedback Loops: Creating systems that learn from past predictions and outcomes to improve future accuracy.
- Adaptive Algorithms: Developing algorithms that can adjust to new data patterns without needing complete retraining.

10. Predictive Analytics for Talent Acquisition

- Salary Benchmarking: Using models to benchmark salaries during recruitment processes to ensure competitive offers.
- Retention Strategies: Analyzing factors leading to employee turnover and their impact on salary trends.

By focusing on these areas, the future of employee salary prediction using machine learning can enhance fairness, accuracy, and strategic decision-making in compensation practices.

7.2 CONCULUSION

In conclusion, employee salary prediction using machine learning represents a transformative approach to understanding and managing compensation within organizations. By leveraging advanced algorithms and data analytics, businesses can gain valuable insights into salary structures, identify fair compensation practices, and make informed decisions that align with market trends.

The integration of diverse data sources, coupled with techniques to ensure fairness and transparency, can lead to more equitable salary determinations. Furthermore, the ability to continuously refine predictive models allows organizations to adapt to changing economic conditions and workforce dynamics.

As companies increasingly prioritize data-driven strategies, the potential for machine learning in salary prediction not only enhances operational efficiency but also fosters a more engaged and satisfied workforce. Ultimately, embracing these technologies can lead to better talent acquisition, retention, and overall organizational performance.

4o mini

In today's real world, it has become tough to store such huge data and extract them for one's own requirement. the extracted data should be useful. The linear regression algorithm helps to fulfill customers by increasing the accuracy of estate choice and reducing the risk of investing in an estate.

The major goal of this project is to determine an applicant's appropriate future wage based on various domain-specific parameters. To train and perform predictions using the ML model, Multiple linear regression, Lasso regression and Random forest . The classification report is used as a comparison criteria to evaluate the overall efficiency of both algorithms once they have been imported. Random Forest outperforms the other THREE algorithms. The best prediction can then be picked. It can be improved by doing the following:

- It can provide more advanced software for tallying salary mediums
- It will host the platform on web servers
- It can handle larger databases and curves than the sample above.

REFERENCES & BIBLIOGRAPHY

8.1BIBLIOGRAPHY

Creating 8 bibliographies for a research project on employee salary prediction using machine learning involves the various academic papers, books, articles, and online resources that provide relevant information and insights on the topic. Here's a sample bibliography to get you startled .

- Title: Employee Salary Prediction using Machine Learning" Author: Boudreau, J. W., & Cascio, W. F.
- publication Year: 2017 source Type: Book SBN: 978-1
- 13891706
- Title: "Employee Salary Prediction with Machine Learning: A Comparative Study”
- Author: Sharma, R., & Smith, J. publication Year: 2020
- source Type: Conference Paper
- URL: [Link to the conference paper if applicable]
- Title: "Feature Engineering for Predictive Modeling of Employee Salaries" Author: Chen, J., & Li, H.
- Publication Year: 2018 Source Type: Journal Article
- Journal: Journal of Machine Learning Research, 19(1), 514-548
- Title: "A Survey of Machine Learning Techniques for Salary Prediction”

8.2 REFERENCES

- Mangui Wu, Shunmin Shu," Top Management Salary, Stock Ratio and Firm Performance: A Comparative Study of State owned and Private Listed Companies in China
- Navyashree M, Navyashree M K, Neetu M, Pooja G R, Arun Biradar "Salary Prediction in It Job Market", International Journal of Computer Sciences and Engineering Vol.-7, May 2019.
- Pornthep Khongchai, Pokpong Songmuang, "Improving Students' Motivation to Study using Salary Prediction System" 2016 13th International Joint Conference on Computer Science and Software Engineering (JCSSE)
- Phuwadol Viroonluecha, Thongchai Kaewkiriya," Salary Predictor System for Thailand Labour Workforce using Deep Learning" The 18th International Symposium on Communications and Information Technologies (ISCIT 2018)
- D.M Lothe,Prakash Tiwari, Nikhil Patil, Sanjana Patil, Vishwajeet Patil, "SALARY PREDICTION USING MACHINE LEARNING" 20216 International Journal Of Advanced Scientific Research and Engineering Trends (IJASRET
- B. K. Han, C. K. Rigsby, and J. Leipsic, "... . An expert consensus document of the Society of Cardiovascular Computed Tomography (SCCT): endorsed by the Society of Pediatric Radiology (SPR) and the ...," Journal of, 2015.
- AI, IoT and the Fourth Industrial Revolution ReviewVOLUME 10, ISSUE 2Page | 9[2]A. K. Jones, D. E. Hintenlang, and W. E. Bolch, "Tissue-equivalent materials for construction of tomographic dosimetry phantoms in pediatric radiology," Med. Phys., vol. 30, no. 8, pp. 2072–2081, Aug. 2003.
- U. M. Reddy, A. Z. Abuhamad, D. Levine, and G. R. Saade, "... , American College of obstetricians and Gynecologists, American College of radiology, Society for pediatric radiology, and society of radiologists in ultrasound fetal ...," American journal of, 2014.
- R. A. Pugliesi, "Recent Developments in AI Algorithms for Pediatric Radiology: Advancements in Detection, Diagnosis, and Management," International Journal of Applied Health Care Analytics, vol. 3, no. 10, pp. 1–20, 2018.
- M. M. Moore, E. Slonimsky, A. D. Long, R. W. Sze, and R. S. Iyer, "Machine learning concepts, concerns and opportunities for a pediatric radiologist," Pediatr. Radiol., vol. 49, no. 4, pp. 509–516, Apr. 2019.

- T. J. Greenwood et al., “CT Dose Optimization in Pediatric Radiology: A Multiyear Effort to Preserve the Benefits of Imaging While Reducing the Risks,” Radiographics, vol. 35, no. 5, pp. 1539–1554, Aug. 2015.
- Aljarbouh, “Accelerated Simulation of Hybrid Systems: Method combining static analysis and run-time execution analysis.(Simulation Accélérée des Systèmes Hybrides: méthode combinant analyse statique et analyse à l’exécution).” University of Rennes 1, France, 2017.
- K. McGee, “The role of a child life specialist in a pediatric radiology department,” Pediatr. Radiol., vol. 33, no. 7, pp. 467–474, Jul. 2003.
- K. Ho, C. Morioka, N. J. Mankovich, B. Stewart, and H. K. Huang, “Image acquisition for the pediatric radiology PACS module,” in Medical imaging, inis.iaea.org, 1987.
- R. K. Taira, N. J. Mankovich, and H. K. Huang, “One-Year Experience With A PACS Module In Pediatric Radiology: System Viewpoint,” in Medical Imaging II, 1988, vol. 0914, pp. 1046–1056.
- Aljarbouh, Y. Zeng, A. Duracz, B. Caillaud, and W. Taha, “Chattering-free simulation for hybrid dynamical systems semantics and prototype implementation,” 2016, pp. 412–422.
- M. T. Jacobs, D. P. Frush, and L. F. Donnelly, “The right place at the wrong time: historical perspective of the relation of the thymus gland and pediatric radiology,” Radiology, vol. 210, no. 1, pp. 11–16, Jan. 1999.
- M. E. Arch and D. P. Frush, “Pediatric body MDCT: a 5-year follow-up survey of scanning parameters used by pediatric radiologists,” AJR Am. J. Roentgenol.,vol. 191, no. 2, pp. 611–617, Aug. 2008.
- S. W. Hilton, D. K. Edwards, and J. W. Hilton, “Practical pediatric radiology,” WB Saunder Comp, 1984.
- L. Merewitz and J. H. Sunshine, “A portrait of pediatric radiologists in the United States,” AJR Am. J. Roentgenol., vol. 186, no. 1, pp. 12–22, Jan. 2006.
- Aljarbouh and B. Caillaud, “Chattering-free simulation of hybrid dynamical systems withthe functional mock-up interface 2.0,” 2016, vol. 124, pp. 95–105.
- R. S. Ayyala, F. S. Ahmed, C. Ruzal-Shapiro, and G. A. Taylor, “Prevalence of Burnout Among Pediatric Radiologists,” J. Am. Coll. Radiol., vol. 16, no. 4 Pt A, pp. 518–522, Apr. 2019.
- J. Billinger, R. Nowotny, and P. Homolka, “Diagnostic reference levels in pediatric radiology in Austria,” Eur. Radiol., vol. 20, no. 7, pp. 1572–1579, Jul. 2010.

- L. Lança and A. Silva, "Digital Radiology and Picture Archiving and Communication System (PACS)," in Digital Imaging Systems for Plain Radiography, L. Lanca and A. Silva, Eds. New York, NY: Springer New York, 2013, pp. 137–158.
- K. Darge, F. Papadopoulou, A. Ntoulia, and D. I. Bulas, "Safety of contrast-enhanced ultrasound in children for non-cardiac applications: a review by the Society for Pediatric Radiology (SPR) and the International Contrast ...," Radiology, 2013.
- J. O. Haller, T. L. Slovis, and A. Joshi, Pediatric radiology, 3rd ed. Berlin, Germany: Springer, 2005.
- E. J. Hall, "Radiation biology for pediatric radiologists," Pediatr. Radiol., vol. 39 Suppl 1, pp. S57-64, Feb. 2009. AI, IoT and the Fourth Industrial Revolution ReviewVOLUME 10, ISSUE 2Page | 10
- Aljarbouh and B. Caillaud, "On the regularization of chattering executions in real time simulation of hybrid systems," 2015, p. 49.
- J. G. Blickman, B. R. Parker, and P. D. Barnes, "Pediatric radiology: the requisites e-book," 2009.
- R. W. Arnold, M. J. Goske, D. I. Bulas, E. C. Benya, J. Ying, and J. H. Sunshine, "Factors influencing subspecialty choice among radiology residents: a case study of pediatric radiology," J. Am. Coll. Radiol., vol. 6, no. 9, pp. 635–642, Sep. 2009.
- M. Alexander, "Managing patient stress in pediatric radiology," Radiol. Technol., vol. 83, no. 6, pp. 549–560, Jul-Aug 2012.
- Aljarbouh and B.Caillaud, "Robust simulation for hybrid systems: chattering path avoidance," arXiv preprint arXiv:1512.07818, 2015.
- G. Alzen and G. Benz-Bohm, "Radiation protection in pediatric radiology," Dtsch. Arztebl. Int., vol. 108, no. 24, pp. 407–414, Jun. 2011.
- J. D. Reyes, "Intestinal Transplantation: An Unexpected Journey," J. Pediatr. Surg., vol. 49, no. 1, pp. 13–18, Jan. 2014.
- F. Lacaille et al., "Intestinal Failure–Associated Liver Disease: A Position Paper of the ESPGHAN Working Group of Intestinal Failure and Intestinal Transplantation," J. Pediatr. Gastroenterol. Nutr., vol. 60, no. 2, p. 272, Feb. 2015.
- S. Gadde, S. Poda, S. Veeravalli, and L. Addala, "PREVALENCE OF KRAS CODON 12 MUTATION IN PATIENTS WITH ORAL SQUAMOUS CELL CARCINOMA (OSCC) FROM SOUTH INDIAN POPULATION," International Research Journal of Natural and Applied Sciences, vol. 11, no. 3, pp. 108–119, 2016.

- S. Gadde, S. Poda, S. Veeravilli, and L. Addala, “Lack of the brafv600e mutation in oral squamous cell carcinoma,” Journal of Medical Science And Clinical Research, vol. 4, p. 14912, 2016.
- Aljarbouh, “Accelerated simulation of hybrid systems: method combining static analysis and run-time execution analysis.” Rennes 1, 2017.
- Aljarbouh, “Non-standard zeno-free simulation semantics for hybrid dynamical systems,” 2019, pp. 16–31.
- Aljarbouh, A. Duracz, Y. Zeng, B. Caillaud, and W. Taha, “Chattering-free simulation for hybrid dynamical systems,” HAL, vol. 2016, 2016.
- D. Nelson-Gruel, Y. Chamaillard, and A. Aljarbouh, “Modeling and estimation of the pollutants emissions in the Compression Ignition diesel engine,” 2016, pp. 317–322.
- J. W. Millar, G. Gupte, and K. Sharif, “Intestinal transplantation for motility disorders,” Semin. Pediatr. Surg., vol. 18, no. 4, pp. 258–262, Nov. 2009.
- Martinez Rivera and P. W. Wales, “Intestinal transplantation in children: current status,” Pediatr. Surg. Int., vol. 32, no. 6, pp. 529–540, Jun. 2016.
- R. P. Harmel Jr, “A simplified technique of small intestinal transplantation in the rat,” J. Pediatr. Surg., vol. 19, no. 4, pp. 400–403, Aug. 1984.
- V. K. Raghu, J. L. Beaumont, M. J. Everly, R. S. Venick, F. Lacaille, and G. V. Mazariegos, “Pediatric intestinal transplantation: Analysis of the intestinal transplant registry,” Pediatr. Transplant., vol. 23, no. 8, p. e13580, Dec. 2019.