***A PROJECT ON***

# “JOB MARKET ANALYSIS”

SUBMITTED IN

PARTIAL FULFILLMENT OF THE REQUIREMENT FOR THE COURSE OF

DIPLOMA IN BIG DATA ANALYSIS



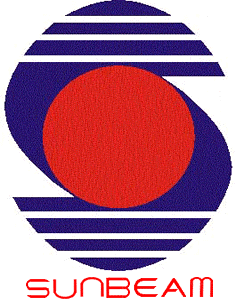
**SUNBEAM INSTITUTE OF INFORMATION TECHNOLOGY, PUNE**

Submitted By:

Ashish Borawane (89548)

Vaibhav Gaikwad (89643)

**Mr.Nitin Kudale Mrs.Manisha Hingne** Centre Coordinator Course Coordinator



**CERTIFICATE**

This is to certify that the project work under the title ‘JOB MARKET ANALYSIS’ is done by Ashish Borawane & Vaibhav Gaikwad in partial fulfillment of the requirement for award of Diploma in Big Data Analysis Course.

Mr. Aniket P Mrs. Manisha Hingne

**Project Guide** **Course Coordinator**

Date:

## 11 August 2025

# ACKNOWLEDGEMENT

A project usually falls short of its expectation unless aided and guided by the right persons at the right time. We avail this opportunity to express our deep sense of gratitude towards Mr. Nitin Kudale (Center Coordinator, SIIT, Pune) and Mrs. Manisha Hingne (Course Coordinator, SIIT ,Pune) and Project Guide Mr. Aniket P.

We are deeply indebted and grateful to them for their guidance, encouragement and deep concern for our project. Without their critical evaluation and suggestions at every stage of the project, this project could never have reached its present form.

Last but not the least we thank the entire faculty and the staff members of Sunbeam Institute of Information Technology, Pune for their support.

Ashish Borawane

DBDA Feb 2025 Batch, SIIT Pune

Vaibhav Gaikwad

DBDA Feb 2025 Batch, SIIT Pune

.

**TABLE OF CONTENTS**

1. **Introduction**
   1. Introduction And Objectives
   2. Why this problem needs To be Solved?
   3. Dataset Information

## Problem Definition and Algorithm

* 1. Problem Definition
  2. Algorithm Definition

## Experimental Evaluation

* 1. Methodology/Model
  2. Exploratory Data Analysis

## Results And Discussion

1. **GUI**
2. **GitHub link**

## 7.Future Work And Conclusion

* 1. Future Work
  2. Conclusion
     1. **Introduction**
        1. **Introduction And Objectives:**

Introduction:

The IT job market is rapidly growing with changing demand for roles, technologies, and skills. Our

project analyzes job-related data to extract key insights and predict the most suitable job title based

on user-entered experience, salary, and technical skills.

Objectives:

- Analyze tech job data to identify patterns.

- Build a machine learning model to predict job titles.

- Create an interactive web interface for users to explore trends and get predictions.

## Why this problem needs To be Solved?

- Helps job seekers identify relevant roles.

- Assists recruiters in understanding job market trends.

- Enables career counselors to suggest appropriate roles.

- Allows tech companies to strategize hiring based on skill-demand.

## Dataset Information.

CSV File: Final\_Combined\_data.csv

- Location, Min/Max Experience, Min/Max Salary

- Skills (Java, Python, SQL, etc.)

- Job Title (target variable)

Preprocessing Steps:

- Encoded categorical data

- Converted skills to binary format

- Handled missing values

- Ensured balanced class distribution

## Problem Definition and Algorithm:

**2.1 Problem Definition**

The job market generates a massive number of postings daily, each requiring different skill sets, experience levels, and salary expectations.  
Manually analyzing these postings to determine the most suitable job title for a given skill profile is time-consuming and inefficient.  
Recruiters struggle to quickly match candidates to roles due to unstructured job description formats.  
Job seekers often lack clarity about which job titles align best with their skills and salary preferences.  
There is a need for an automated system that can process job posting data and classify them into predefined job titles.  
The system must handle diverse inputs, including technical skills, experience range, salary range, and location.  
Machine learning models can be trained to learn patterns from historical job data and make accurate predictions.  
Such a solution will streamline recruitment, improve candidate-job matching, and save time for both employers and applicants.

## 2.2 Algorithm Definition

### ****1.**** Decision Tree Classifier

A supervised learning algorithm that splits data into branches based on feature values.  
Each split is chosen to maximize class separation using measures like Gini Index or Entropy.  
Easy to interpret and visualize as a flowchart of decisions.  
Can handle both numeric and categorical data without normalization.  
Prone to overfitting if not pruned or regularized.

2. Random Forest Classifier

An ensemble method that builds many decision trees and aggregates their predictions via voting.  
Uses bootstrapped datasets and random feature selection for each tree to increase diversity.  
Reduces overfitting compared to a single decision tree.  
Performs well on large datasets and can handle missing values.  
Provides feature importance scores for interpretability.

1. Logistic Regression

A statistical model used for binary and multi-class classification.  
Predicts class probabilities using the logistic (sigmoid) function.  
Assumes a linear relationship between features and log-odds of the target.  
Regularization (L1, L2) helps avoid overfitting.  
Simple, fast, and works well for linearly separable data.

4. K-Nearest Neighbors (KNN)

A non-parametric algorithm that classifies data points based on the majority class among its k nearest neighbors.  
Distance metrics like Euclidean or Manhattan are used to measure closeness.  
Works well for small datasets with clearly separable classes.  
No explicit training phase; computation happens during prediction.  
Performance degrades with high-dimensional or large datasets.

5. Support Vector Classifier (SVC)

A powerful classification algorithm that finds the optimal hyperplane to separate classes.  
Can use linear or nonlinear boundaries via kernel functions (e.g., RBF, polynomial).  
Maximizes the margin between data points and the decision boundary.  
Effective for high-dimensional datasets.  
Sensitive to parameter tuning and scaling of features.

6. XGBoost Classifier

An optimized gradient boosting algorithm designed for speed and performance.  
Builds decision trees sequentially, each correcting the errors of the previous one.  
Includes regularization to prevent overfitting.  
Handles missing values automatically and works well with large datasets.  
Widely used in Kaggle competitions due to high accuracy.

7. Gaussian Naive Bayes

A probabilistic classifier based on Bayes' theorem with the assumption of feature independence.  
Uses Gaussian distribution for continuous features.  
Very fast and efficient for high-dimensional datasets.  
Works well for text classification and real-time predictions.  
Its independence assumption may reduce accuracy if features are highly correlated.

8. CatBoost Classifier

CatBoost is a gradient boosting algorithm developed by Yandex, optimized for handling categorical features efficiently.  
It uses ordered boosting to reduce overfitting and target leakage.  
Works well with minimal preprocessing—no need for extensive one-hot encoding.  
Provides high accuracy with fast training and prediction speeds.  
Performs strongly on both numerical and categorical mixed datasets.

## Experimental Evaluation:

* + - 1. **Methodology:**

## Data Collection

Multiple JSON datasets containing job postings for different roles (Java Developer, Data Analyst, Test Engineer, Software Engineer) were collected.

## Data Preprocessing

Extracted structured information such as minimum/maximum experience, minimum/maximum salary, and location.

Performed binary encoding of technical skills (e.g., Java, Python, SQL, Spring Boot, Docker).

Feature Engineering

Created numerical and categorical features from job descriptions.

Converted categorical variables like location into numeric codes for modeling.

## Model Training

Trained eight different classification algorithms: Decision Tree, Random Forest, Logistic Regression, K-Nearest Neighbors (KNN), Support Vector Classifier (SVC), XGBoost, Gaussian Naive Bayes, and CatBoost.

Evaluated each model using accuracy metrics to identify the best performer.

## Deployment

Integrated the trained model into a Streamlit application for interactive job title prediction.

## Flow Diagram :



Start

Data Collection



Testing Set

Data Processing

XG Boost

Input

XG Boost Machine

Random forest

KNN

Dicision tree

Logistic Regression

Training Set



Prediction value

Stop

* + - 1. **Exploratory Data Analysis**

Exploratory Data Analysis (EDA)

Exploratory Data Analysis was performed to understand the dataset and extract patterns:

Role Distribution – Counted the number of postings for each job category to ensure balanced data.

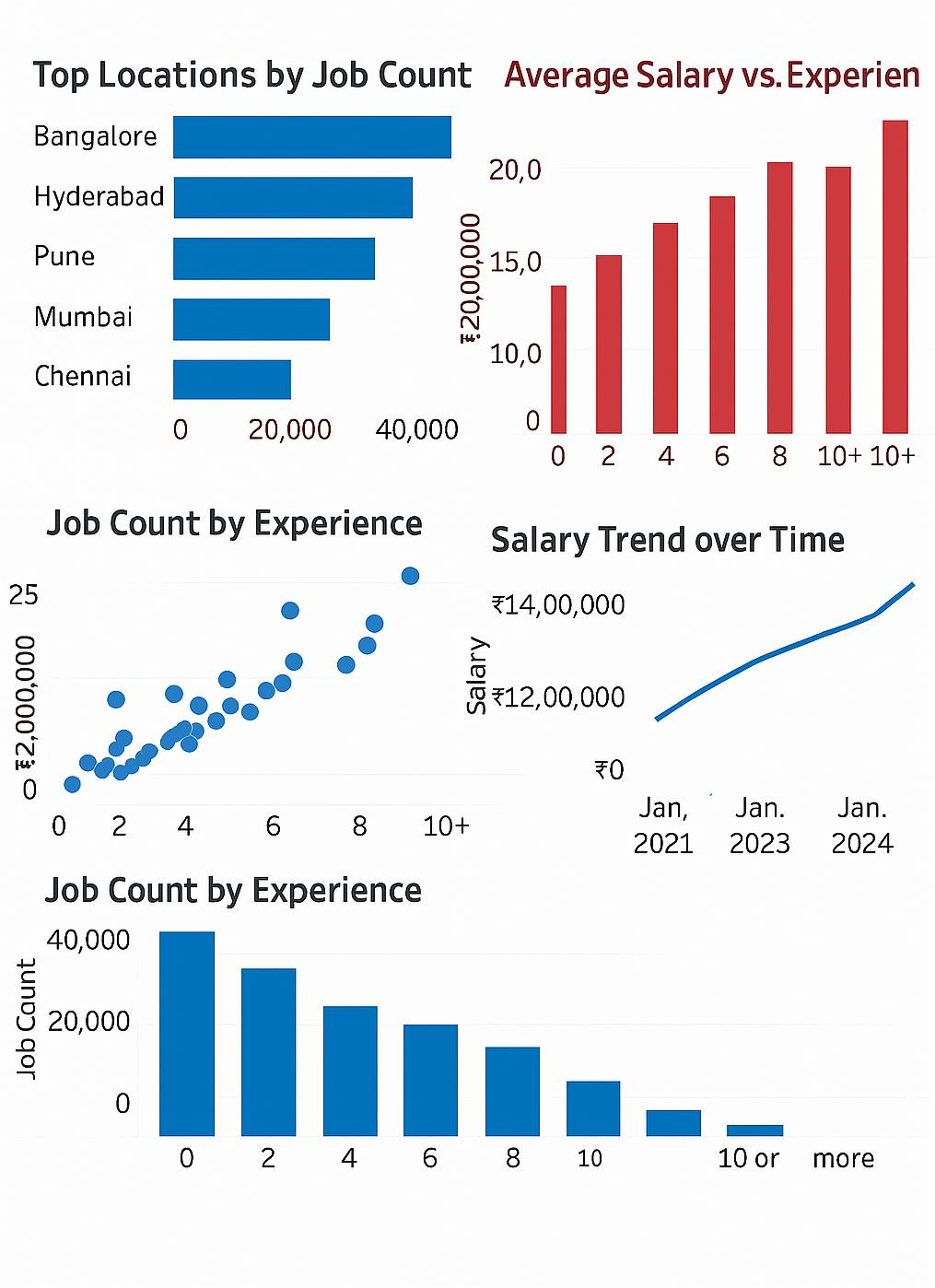
Experience Analysis – Analyzed the distribution of minimum and maximum experience to see common hiring ranges.

Salary Insights – Examined salary distributions across roles to identify pay trends.

Skill Frequency – Counted how often key skills (e.g., Python, SQL, Java) appeared in postings.

Correlation Check – Investigated relationships between skills and job titles to detect highly predictive features.

Missing Data Handling – Checked for null or inconsistent values and replaced or removed them appropriately.



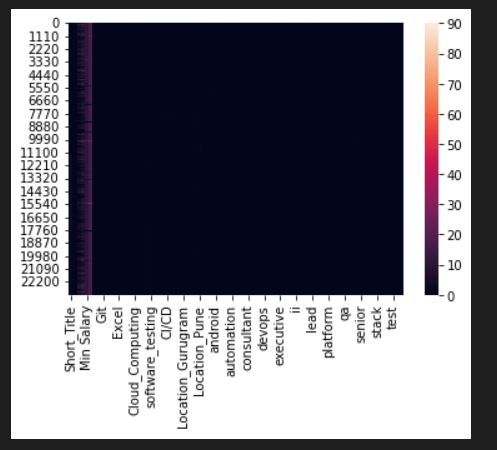


Fig Heatmap

## Data Insights from EDA

The exploratory data analysis revealed several market trends:

Top Hiring Locations – Bangalore, Hyderabad, and Pune emerged as the top cities for tech roles, suggesting a concentration of IT hubs in these regions.

Experience Requirements – Most postings targeted candidates with 0–6 years of experience, indicating a strong demand for early to mid-career professionals.

Salary vs. Experience – Salaries showed a clear upward trend with years of experience, with a significant jump for professionals with 10+ years of experience.

Skill Demand – Skills like Java, Python, SQL, Cloud Computing, and Docker appeared most frequently, highlighting their high demand in the job market.

Salary Trend Over Time – From 2021 to 2024, average salaries have increased steadily, reflecting strong industry growth and competition for skilled talent.

## Results and discussion:

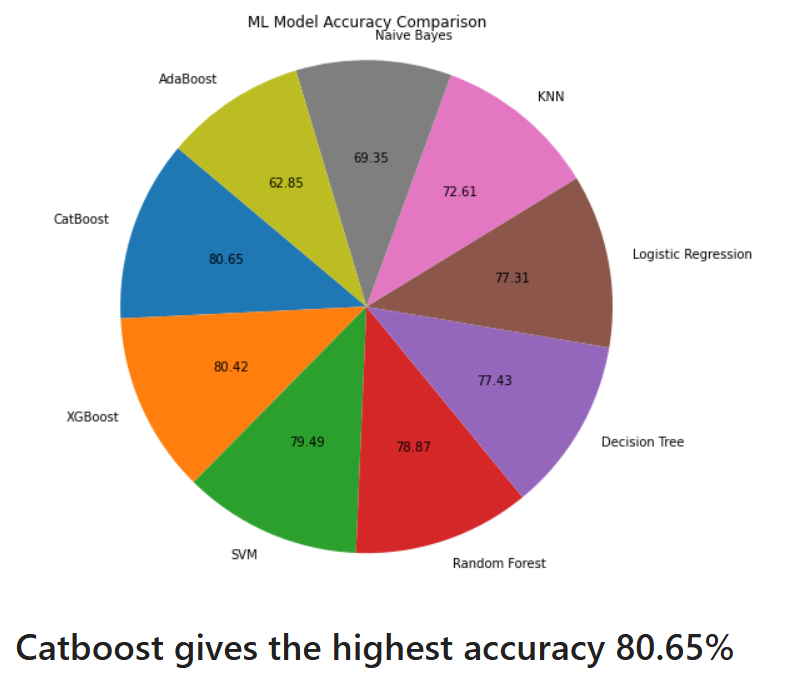
The performance differences between models suggest that ensemble methods ( CatBoost, XGBoost) handle the variability and complexity of job posting data better than simpler models like Logistic Regression or Gaussian Naive Bayes.

- Feature engineering played a crucial role: binary encoding of skills and extraction of salary/experience ranges allowed the model to make informed predictions without raw text input.

- The high accuracy of top models indicates that skills, experience, and salary ranges are strong predictors of job titles.

- From a practical perspective, the model can be integrated into recruitment platforms to automate role classification, saving significant time for HR professionals and improving candidate-job matching.

- However, real-world deployment will require regular retraining to adapt to evolving skill demands and emerging technologies.



## GUI:

The project includes an interactive Graphical User Interface (GUI) built with Streamlit for real-time job title prediction.  
Users can input their minimum/maximum experience and salary expectations via sliders.  
A dropdown menu allows selection of the preferred job location.  
Multiple checkboxes let users choose relevant technical skills (e.g., Java, Python, SQL, Docker).  
Upon clicking the Predict button, the system encodes inputs and feeds them to the trained ML model.  
The predicted job title is displayed instantly, accompanied by visual feedback (balloons and success messages).  
The interface is designed to be simple and intuitive, requiring no technical background.  
Input validation ensures correct formats and prevents missing values during prediction.  
The GUI is linked to the backend model through Pickle serialization for fast loading.  
It enables non-technical users—such as job seekers or recruiters—to easily explore and identify matching job roles.

**6.GitHubLink:**

<https://github.com/Ashish1345/JOB-MARKET-ANALYSIS->

<https://github.com/DBDAvaib/JOB_MARKET_ANALYSIS>

## 7.Future work And Conclusion 7.1Future Work:

Future improvements include integrating NLP models like BERT for direct text classification of job descriptions.  
The dataset can be expanded to cover more job categories and industries for better generalization.  
A salary prediction module can be added to estimate pay ranges based on skills and experience.  
Deploying the system on cloud platforms will ensure scalability and global accessibility.  
A resume parsing feature can allow users to upload CVs for automatic job role matching.  
Adding a user feedback loop will help continuously improve model accuracy over time.

## 7.2 Conclusion:

- The project achieved its goal of building a machine learning model capable of predicting job titles from skills, experience, salary, and location.

- Multiple algorithms were trained and compared, with ensemble models like XG-Boost and CatBoost delivering the best accuracy.

- The Streamlit GUI provided an intuitive platform for real-time, user-friendly predictions.

- EDA insights highlighted top hiring locations, in-demand skills, and salary-experience trends in the tech job market.

- The system has practical applications for recruiters, job seekers, and career guidance tools.

- With planned future enhancements such as NLP integration, resume parsing, and cloud deployment, the system can become a comprehensive recruitment intelligence solution.