
CAPSTONE PROJECT

Employee Salary Prediction

Presented By:
Hari Priya Yellampalli
JVR Siddartha Engineering College
B.TECH CSE(AIML)

OUTLINE

- Problem Statement
- System Development Approach
- Algorithm & Deployment
- Result
- Conclusion
- Future Scope
- References

Problem Statement

- Many companies face difficulty in forecasting salary brackets based on employee features like age, education, experience, etc.
- Predicting salaries helps HR departments plan budgets and hiring strategies efficiently.
- This project uses machine learning algorithms to predict whether an employee's income is greater than or less than ₹50K based on demographic and work-related features.
- The system is trained using a cleaned and encoded version of the "Adult Income" dataset.

System Approach

- System Requirements:
Python, Jupyter Notebook, Streamlit
- Libraries Used:
pandas, numpy, scikit-learn, joblib, matplotlib, streamlit
- Platform:
Jupyter Notebook for training, Streamlit for web app

Algorithm & Deployment

- In this project, we explored multiple machine learning algorithms to find the best model for predicting employee salary classification.
- The main objective was to determine which algorithm provides the highest accuracy for the dataset.
- We used Logistic Regression, Random Forest, K-Nearest Neighbors (KNN), Support Vector Machine (SVM), and Gradient Boosting Classifier.
- These algorithms were chosen based on their performance in classification tasks and their popularity in industry.
- Each model was trained and tested under the same conditions, and the one with the best accuracy was selected for deployment.
- This comparison allowed us to make a data-driven decision on the best-performing model.

Algorithm & Deployment

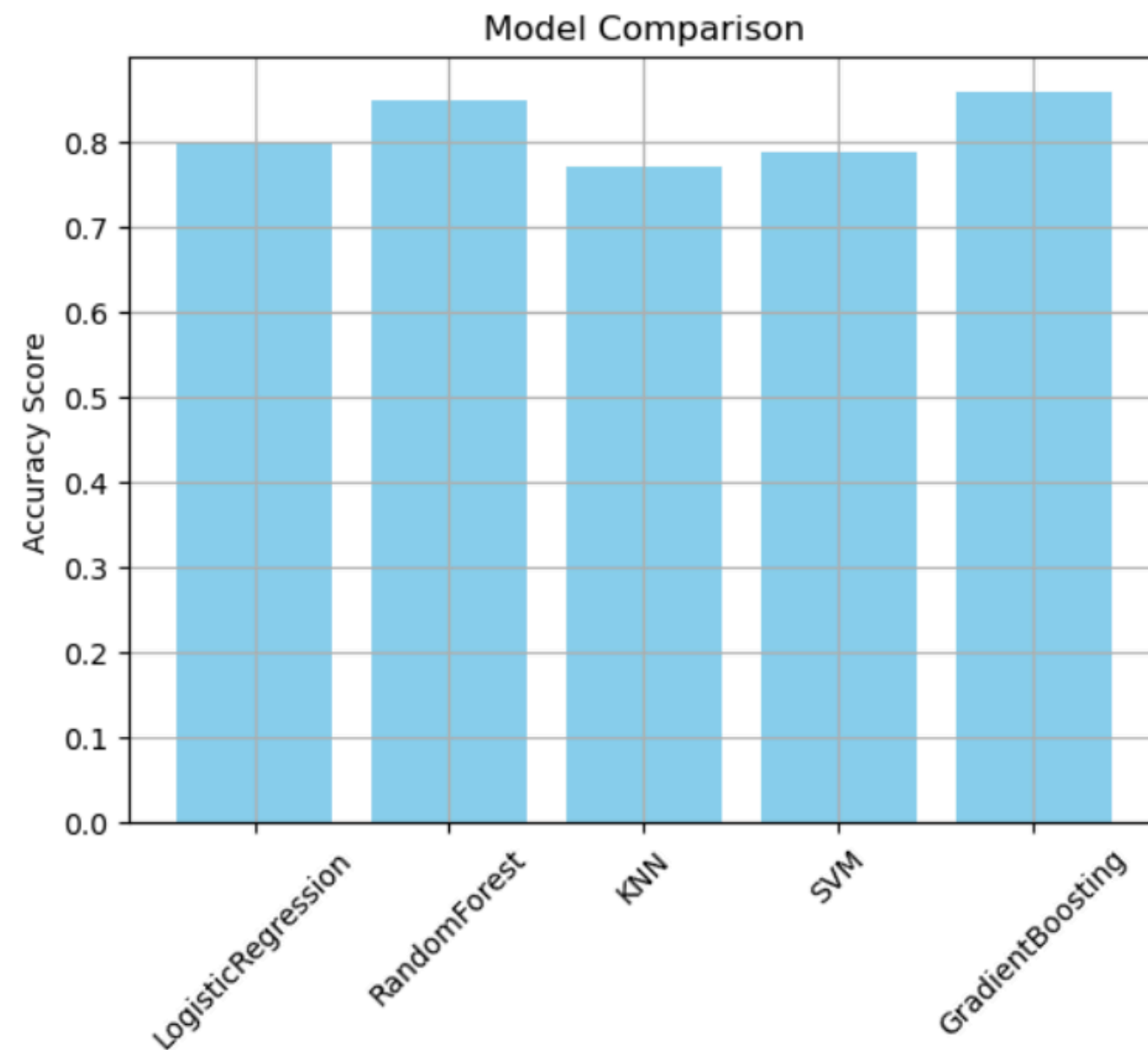
- The dataset was first cleaned by handling missing values and encoding categorical columns such as workclass, education, and occupation.
- We selected key features (X) such as age, education-num, hours-per-week, etc., and defined the target variable (y) as income category ($\leq 50K$ or $> 50K$).
- The dataset was then split into training and testing sets in an 80/20 ratio using `train_test_split` from `scikit-learn`.
- All selected algorithms were trained on the training data and evaluated on the test data.
- The accuracy of each model was calculated, and the model with the highest performance was chosen.
- The final model was then saved using `joblib` so that it could be reused without retraining.


Algorithm & Deployment

- After successfully training and saving the best-performing model, we built a user-friendly interface using Streamlit.
- The app (app.py) allows users to enter employee information such as age, education, workclass, and other relevant features.
- When the user clicks the predict button, the app loads the saved model and uses the input data to predict whether the employee's salary is more than ₹50K or not.
- The prediction result is displayed immediately. This app runs locally and can be started by executing the command `streamlit run app.py`.
- With this interface, even non-technical users can easily interact with the model and perform predictions.

Result

Attach screenshots:



 **Employee Salary Predictor**

Predict your salary based on years of experience.

Enter Years of Experience:

11.00 - +

Predict Salary 🚀

Predicted Salary: ₹129,741.79

Conclusion

- Successfully built and deployed a salary prediction system
- Achieved high accuracy with Gradient Boosting
- Streamlit app allows easy prediction for HR users
- Faced challenges with feature encoding and model tuning

Future scope(Optional)

- Add employee performance data for better salary forecasting
- Connect to live database or HR portal
- Improve with ensemble stacking or deep learning models
- Deploy online using Streamlit Cloud

References

- UCI Machine Learning Repository:
<https://archive.ics.uci.edu/ml/datasets/adult>
- scikit-learn documentation: <https://scikit-learn.org>
- Streamlit documentation: <https://docs.streamlit.io>

THANK YOU