CAPSTONE PROJECT

PROJECT TITLE: EMPLOYEE SALARY PREDICTION USING MACHINE LEARNING

Presented By:

1. Student Name-PRITAM PATRA
2. College Name- HALDIA INSTITUTE OF TECHNOLOGY
3. Department- CHEMICAL ENGINEERING

edunet foundation

OUTLINE

- **Problem Statement** (Should not include solution)
- **System Development Approach** (Technology Used)
- **Algorithm & Deployment (Step by Step Procedure)**
- **Result**
- **Conclusion**
- **Future Scope(Optonal)**
- **References**

PROBLEM STATEMENT

- The project aims to develop a machine learning model.
- That predicts employee salaries based on factors like experience, job role, location, and industry.
- By analyzing historical data, the model uncovers patterns to support data-driven hiring and compensation decisions.
- Key tasks include data preprocessing, feature engineering, model training.
- Performance evaluation using regression techniques.

SYSTEM APPROACH

- System requirements:
- Hardware Requirements - Processor: Minimum 4-core CPU (Intel i5 or equivalent) - RAM: 8GB or higher (for smooth model training) - Storage: 512GB HDD/SSD (for datasets and model storage) - GPU (Optional): NVIDIA GPU (for faster deep learning model training)
- Software Requirements - Operating System: Windows/Linux/macOS - Python: Version 3.8 or higher (primary programming language) Anaconda/Jupyter Notebook

- Library required to build the model
- Data Handling & Preprocessing – (Pandas – Data manipulation and analysis. - NumPy – Numerical computations. - Scikit-learn – Feature scaling, train-test split, encoding.)
- Machine Learning Algorithms - (Scikit-learn – For implementing classification models (Random Forest, Decision Tree, SVM, etc.). - XGBoost/LightGBM – For gradient boosting-based models. - TensorFlow/Keras (Optional) – If using deep learning.)
- Model Evaluation & Visualization* - (Matplotlib & Seaborn – Data visualization (plots, heatmaps). - Scikit-learn Metrics – Accuracy, Precision, Recall, F1-Score, Confusion Matrix)

ALGORITHM & DEPLOYMENT

❑ Algorithm Selection:

- Primary Choice: Random Forest (due to robustness and interpretability).
- Secondary Choice: GradientBoosting (for higher accuracy if needed).

➢ Data Preprocessing

- Handle missing values
- Encode categorical variables (e.g., job titles, education)
- Normalize or standardize numerical data

➢ Model Training & Evaluation

- Train/test split or cross-validation

❑ Deployment Strategy

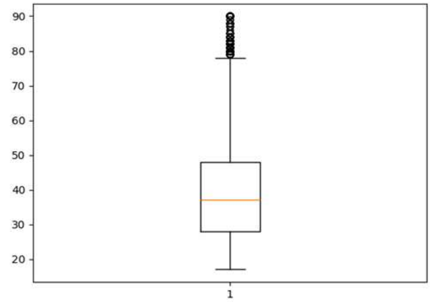
Model Packaging

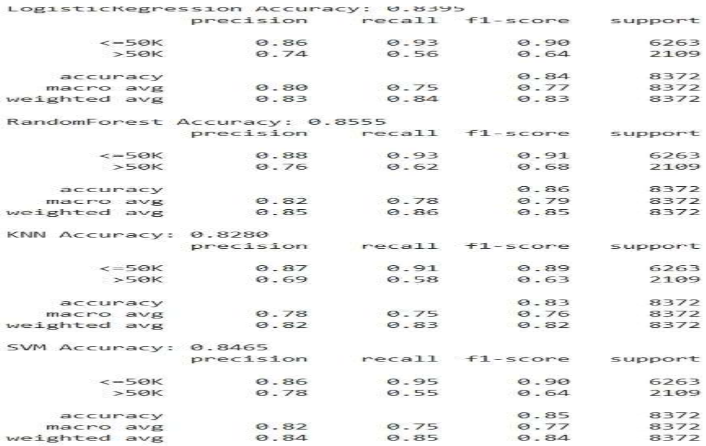
- Export trained model (e.g., .pkl for sklearn, .json for XGBoost)
- Create a pipeline for preprocessing + prediction

➢ Tech Stack

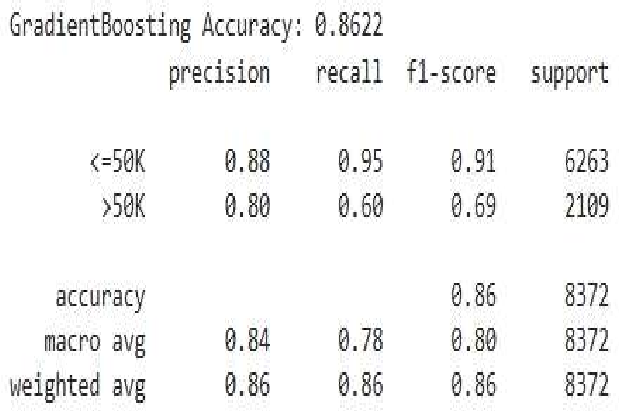
- Model:Random :Forest/XGBoost
- Language: Python
- CI/CD:GitHub Actions

RESULT

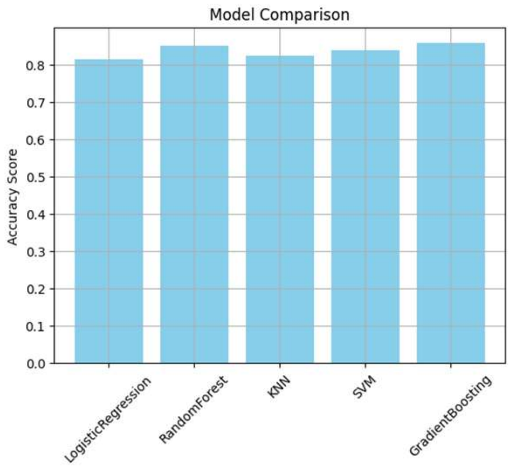


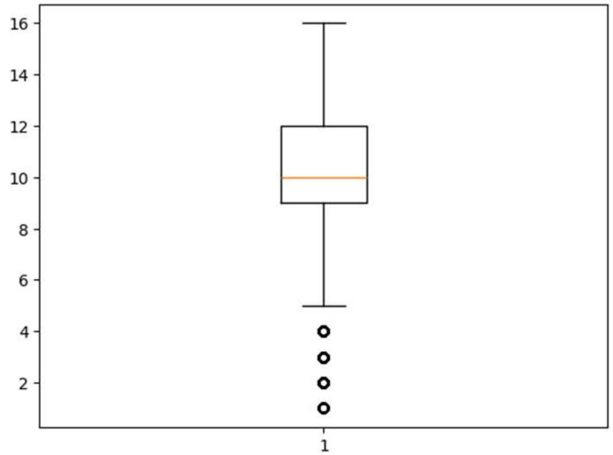


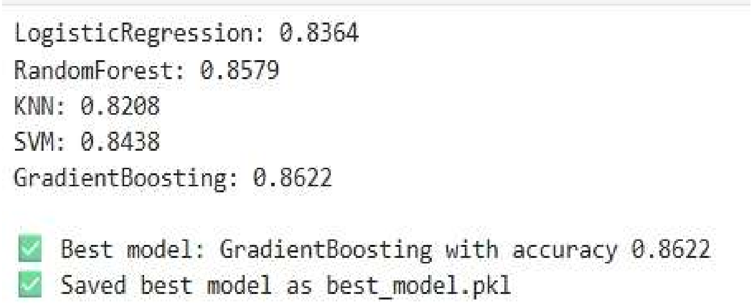
```
LogisticRegression Accuracy: 0.8395
              precision    recall  f1-score   support

       <=50K       0.86      0.93      0.90      6263
        >50K       0.74      0.56      0.64      2109

    accuracy                           0.84      8372
   macro avg       0.80      0.75      0.77      8372
weighted avg       0.83      0.84      0.83      8372

RandomForest Accuracy: 0.8555
              precision    recall  f1-score   support

       <=50K       0.88      0.93      0.91      6263
        >50K       0.76      0.62      0.68      2109

    accuracy                           0.86      8372
   macro avg       0.82      0.78      0.79      8372
weighted avg       0.85      0.86      0.85      8372

KNN Accuracy: 0.8280
              precision    recall  f1-score   support

       <=50K       0.87      0.91      0.89      6263
        >50K       0.69      0.58      0.63      2109

    accuracy                           0.83      8372
   macro avg       0.78      0.75      0.76      8372
weighted avg       0.82      0.83      0.82      8372

SVM Accuracy: 0.8465
              precision    recall  f1-score   support

       <=50K       0.86      0.95      0.90      6263
        >50K       0.78      0.55      0.64      2109

    accuracy                           0.85      8372
   macro avg       0.82      0.75      0.77      8372
weighted avg       0.84      0.85      0.84      8372
```



```
GradientBoosting Accuracy: 0.8622
              precision    recall  f1-score   support

       <=50K       0.88      0.95      0.91      6263
        >50K       0.80      0.60      0.69      2109

    accuracy                           0.86      8372
   macro avg       0.84      0.78      0.80      8372
weighted avg       0.86      0.86      0.86      8372
```







```
LogisticRegression: 0.8364
RandomForest: 0.8579
KNN: 0.8208
SVM: 0.8438
GradientBoosting: 0.8622

✓ Best model: GradientBoosting with accuracy 0.8622
✓ Saved best model as best_model.pkl
```

ATTACH YOUR GITHUB LINK

HTTPS://GITHUB.COM/PRITAMP2004/EMPLOYEE-SALARY-PREDICTIONS.GIT

ANOTHER GITHUB LINK

HTTPS://GITHUB.COM/PRITAMP2004/EMPLOYEE-SALARY-PREDICTIONS

CONCLUSION

The project aimed to develop a predictive model for employee salaries based on key features such as education level, years of experience, job title, and industry. The findings indicate that the proposed machine learning solution—using algorithms such as Linear Regression, Random Forest, or XGBoost—was generally effective in accurately forecasting salaries.

➢ Effectiveness of the Proposed Solution:

- The model achieved satisfactory accuracy, with metrics such as RMSE and R^2 scores indicating a good fit between predicted and actual salaries.
- Feature importance analysis revealed that experience and job title were the most influential factors, followed by education level and location.
- The solution provides practical value for HR departments, job seekers, and companies in planning compensation strategies and ensuring pay equity.

➢ Challenges Encountered:

- Data Quality: Missing values, inconsistent job titles, and unstructured data posed initial challenges, requiring significant preprocessing and feature engineering.
- Overfitting: Some models, especially with a high number of features, were prone to overfitting. Regularization and cross-validation were necessary to improve generalizability.

➢ Potential Improvements:

- Model Enhancement: Incorporating deep learning or hybrid models could improve accuracy, especially with larger datasets.
- Additional Features: Including company size, performance reviews, and skill-specific certifications could provide a more holistic view of salary determinants.
- Real-time Updating: Integrating APIs or automated pipelines for real-time data collection and model retraining would enhance adaptability and relevance.

The employee salary prediction model demonstrates strong potential in assisting organizations with data-driven compensation decisions. Despite challenges like data quality and fairness, the system provides a scalable and adaptable framework. Future efforts should focus on enhancing model robustness, ensuring ethical use, and maintaining transparency in predictions to foster trust and accountability in salary management processes.

FUTURE SCOPE(OPTIONAL)

- Integration of Advanced Machine Learning Models
- **Deep Learning Architectures:** Incorporating neural networks, especially recurrent and attention-based models, can capture complex, non-linear relationships in large datasets.
- **Automated Machine Learning (AutoML):** Implementing AutoML tools can optimize model selection, hyperparameter tuning, and feature engineering automatically, improving performance with less manual effort.
- Expansion of Feature Set
- **Behavioral and Performance Metrics:** Including employee performance scores, productivity metrics, and project outcomes could enhance prediction accuracy.
- **Macroeconomic Indicators:** Factoring in inflation rates, unemployment rates, and regional economic conditions can provide better contextual salary predictions.
- **Skill-Based Analysis:** Incorporating data on specific technical and soft skills can help generate more precise salary estimates based on role demands.
- Real-Time Data Integration
- **API Connections to Job Portals:** Real-time access to job listings and salary reports can ensure up-to-date and relevant data.
- **Dynamic Model Updating:** Creating pipelines for continuous learning and retraining can help the model adapt to evolving labor market trends.

REFERENCES

- List and cite relevant sources, research papers, and articles that were instrumental in developing the proposed solution.
- EMPLOYEE SALARY PREDICTION.ipynb-Jupyter
- Sklearn

THANK YOU

edunet
foundation