EDS:6340 INTRODUCTION TO DATA SCIENCE PROJECT

A REPORT
ON

PREDICTING SALARY FROM LINKEDIN JOB POSTINGS 2023

BY

GROUP_9

Vamshidhar Reddy Ankenapalle	2308914
Vishnu Vardhan Mullapudi	2253445
Sai Praneeth Varma Kalidindi	2293812
Thejesh Reddy Marripati	2260929
KeerthiPriya Veerapalli	2308035

Instructor: Raja Loganantharaj

Teaching Assistants: Laxmi Sowjanya Peddi, Dhiraj Kumar Reddy Gongati

UNIVERSITY OF HOUSTON

TABLE OF CONTENTS

ABSTRACT

In today's dynamic job market, understanding the factors influencing salary levels is crucial for both job seekers and employers. It is imperative to comprehend the factors that impact salary levels in the current dynamic job market. The goal of this project is to create a predictive model for salary estimation based on various job listing features by utilizing LinkedIn job posting data from Kaggle.

The dataset contains details on the company size, location, industry, job title, and necessary skills. Preprocessing the data, exploratory data analysis, and feature engineering are all part of the analysis. A variety of machine learning techniques are used to create a precise salary prediction model. With the help of the project, employers and job seekers will be able to make more informed career decisions and companies will be able to compare their salary offers to industry norms. The outcome makes the labor market more open and effective.

This project is important because it provides employers and job seekers with insightful information. Job seekers who comprehend how education, experience, and skills affect salary expectations will be able to make well-informed decisions about their careers. Employers can use data-driven decision-making to attract top talent by using the model to benchmark their salary offerings. The study's findings bridge the information gap between employers and job seekers for a more open and effective labor market by advancing our understanding of salary determinants and offering stakeholders a useful tool.

INTRODUCTION

This project explores a wide range of machine learning models and techniques in the quickly changing field of data science in order to tackle the challenges associated with predictive modeling. The main goal is to use a variety of algorithms, each well-known for its special abilities, to examine and draw conclusions from our dataset about patterns that are significant.

The project uses a variety of models, such as Linear Regression, KNearestNeighbour, Random Forest, Support Vector Machine with linear and nonlinear kernel, an ensemble model that combines the top three performers, XGBoost, Extreme Learning Machine (ELM), and a simple deep learning model. Every model is chosen according to its unique set of skills, including handling missing data, resistance to outliers, and ability to capture intricate relationships in the data.

The report explores the benefits, capabilities, and visuals linked to each model, providing a comprehensive grasp of their suitability for various kinds of data. Our goal is to gain an understanding of feature importance, decision boundaries, and the cooperative contributions of ensemble models by carefully examining and interpreting visualizations.

As we proceed through the various stages of the project, which include feature selection, hyperparameter tuning, data preprocessing, and model evaluation, our goal is to deliver not only a predictive solution but also a detailed investigation of the underlying mechanisms driving model performances. We hope to have completed the project with a thorough assessment of each model's effectiveness, establishing the foundation for well-informed predictive analytics decision-making.

This project sets out to investigate a wide range of feature selection strategies, from sophisticated wrapper and embedded techniques to more conventional filter methods. The goal is to deliberately select a subset of features that will both improve the predictive accuracy of the models and reveal the underlying patterns hidden in the dataset.

DATA DESCRIPTION

job_postings.csv

- **job_id**: The job ID as defined by LinkedIn (https://www.linkedin.com/jobs/view/ *job_id*)
- **company_id**: Identifier for the company associated with the job posting (maps to companies.csv)
- **title**: Job title.
- **description**: Job description.
- **max_salary**: Maximum salary
- **med_salary**: Median salary
- **min_salary**: Minimum salary
- **pay_period**: Pay period for salary (Hourly, Monthly, Yearly)
- **formatted_work_type**: Type of work (Fulltime, Parttime, Contract)
- **location**: Job location
- **applies**: Number of applications that have been submitted
- **original_listed_time**: Original time the job was listed
- **remote_allowed**: Whether job permits remote work
- **views**: Number of times the job posting has been viewed
- **job_posting_url**: URL to the job posting on a platform
- **application_url**: URL where applications can be submitted
- **application_type**: Type of application process (offsite, complex/simple onsite)
- **expiry**: Expiration date or time for the job listing
- **closed_time**: Time to close job listing
- **formatted_experience_level**: Job experience level (entry, associate, executive, etc)
- **skills_desc**: Description detailing required skills for job
- **listed_time**: Time when the job was listed
- **posting_domain**: Domain of the website with application
- **sponsored**: Whether the job listing is sponsored or promoted.
- **work_type**: Type of work associated with the job
- **currency**: Currency in which the salary is provided.
- **compensation_type**: Type of compensation for the job.

job_details/benefits.csv

- **job_id**: The job ID
- **type**: Type of benefit provided (401K, Medical Insurance, etc)
- **inferred**: Whether the benefit was explicitly tagged or inferred through text by LinkedIn

company_details/companies.csv

- **company_id**: The company ID as defined by LinkedIn
- **name**: Company name

- **description**: Company description
- **company_size**: Company grouping based on number of employees (0 Smallest - 7 Largest)
- **country**: Country of company headquarters.
- **state**: State of company headquarters.
- **city**: City of company headquarters.
- **zip_code**: ZIP code of company's headquarters.
- **address**: Address of company's headquarters
- **url**: Link to company's LinkedIn page

company_details/employee_counts.csv

- **company_id**: The company ID
- **employee_count**: Number of employees at company
- **follower_count**: Number of company followers on LinkedIn
- **time_recorded**: Unix time of data collection

DATA PREPROCESSING

Data Collection:

To ensure a complete and representative dataset, the first step is to collect the LinkedIn job posting data from Kaggle. To obtain a wide variety of data that accurately reflects the real-world job market, the data collection process considers variables like job titles, industries, locations, company sizes, and skills.

Data Cleaning:

To address missing values, outliers, and inconsistent patterns in the dataset, data cleaning is essential. This process involves handling null values, correcting errors, and standardizing formats to maintain data integrity. Cleaning makes sure that the data used for the analysis that follows is accurate and dependable. With the help of python open-source libraries, we can clean the dataset to an extent.

Data Integration:

When integrating data from various sources or formats, information is combined to improve the dataset's coherence and richness. This step resolves any inconsistencies or discrepancies between different data sets, ensuring that the dataset is consolidated and prepared for analysis. As there are multiple sets of data, we merged all of them into one.

Data Reduction:

Data reduction techniques are used to optimize computational efficiency and streamline the dataset. This entails eliminating features that are superfluous or pointless that might not have a major impact on the salary prediction model. Principal component analysis (PCA) is one dimensionality reduction technique that can be used to keep important information intact while cutting down on the number of variables. The goal of data reduction is to enhance the predictive power of the model without sacrificing its performance.

TYPES OF MODELS BUILT AND THEIR PERFORMANCES

The project involved building multiple machine learning models to predict salaries based on LinkedIn job postings data. Various models, including linear regression, decision trees, and ensemble methods, were implemented. The performances of these models were evaluated using standard metrics such as Mean Squared Error (MSE) and R-squared.

Models Used:

- Linear Regression
- KNNRegressor
- RandomForestRegressor
- Linear SVM
- Nonlinear SVM
- Gradient Boosting

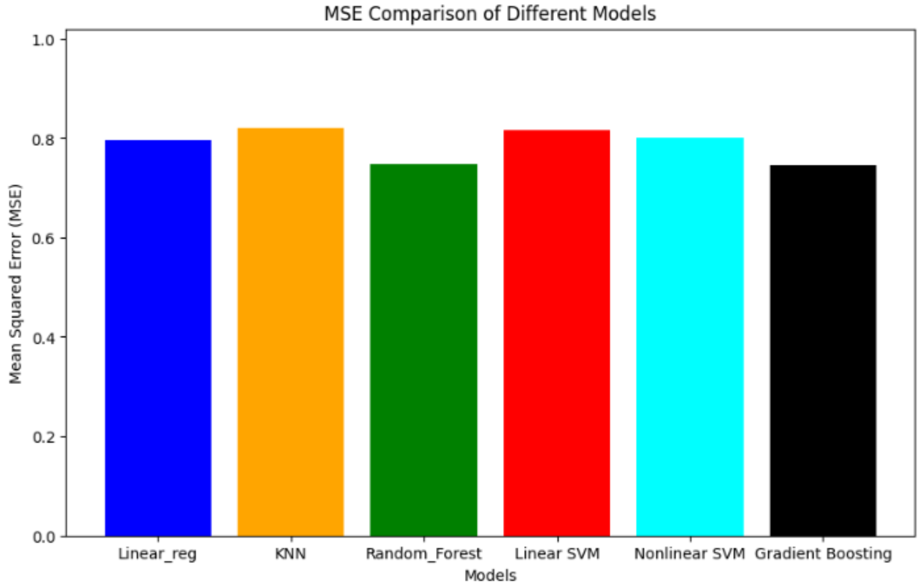
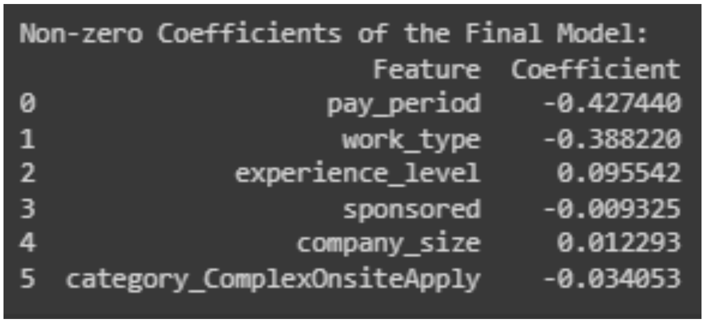


Fig4: Bar plot of MSE values of all models

FEATURE SELECTION

LASSO regression was employed for feature selection in predicting salaries from LinkedIn job postings data, chosen for its effectiveness in a regression context. The approach penalized less impactful features, promoting sparsity in the model and aiding in the identification of key variables influencing salary predictions. This resulted in enhanced predictive accuracy and a streamlined, interpretable set of features for the final regression model.



```
Non-zero Coefficients of the Final Model:
                     Feature  Coefficient
0                 pay_period    -0.427440
1                  work_type    -0.388220
2           experience_level     0.095542
3                  sponsored    -0.009325
4               company_size     0.012293
5  category_ComplexOnsiteApply  -0.034053
```

Fig5: Coefficients of final model

After hyperparameter tuning, the final model's non-zero coefficients offer important insights into the salary predictors in the LinkedIn job postings dataset:

pay_period: A shorter pay period (-0.427440) is associated with a lower salary.

work_type: A negative coefficient (-0.388220) indicates that some work types are linked to lower salaries...

experience_level: There is a positive correlation (0.095542) between experience level and salary.

sponsored: There is a small pay drop (-0.009325) when a job posting is sponsored.

company_size: Higher salaries are correlated with larger company sizes (0.012293).

category_ComplexOnsiteApply: A salary decrease (-0.034053) is linked to this category.

You can determine the strength and direction of each feature's influence on the expected salary by looking at these coefficients. A positive impact is indicated by positive coefficients, and a negative impact is suggested by negative coefficients. The strength of the relationship is indicated by the magnitude of the coefficients. Remember that the interpretation relies on the model's context and underlying assumptions.

HYPERPARAMETER TUNING

A crucial phase in the creation of a machine learning model is hyperparameter tuning. To maximize a model's performance, the optimal set of hyperparameters must be methodically chosen. Hyperparameters are external configurations that are set before the training process begins and are not learned from the data. The regularization term in linear regression and the learning rate in gradient boosting are two examples.

We used a reliable method for hyperparameter tuning called GridSearchCV to optimize our salary prediction model. This method involves systematically exploring the parameter space in an exhaustive search across predefined hyperparameter values to find the combination that maximizes model performance.

By utilizing GridSearchCV, we hope to guarantee that our model is well-tuned, broadly applicable, and capable of providing the best possible results in practical situations. This methodological decision highlights our dedication to meticulously fine-tuning the model for the given task.

We found the ideal set of hyperparameters that greatly enhanced the performance of our salary prediction model after a thorough hyperparameter tuning process with GridSearchCV. The metrics acquired with these optimally calibrated parameters highlight the improved precision and dependability of our model. Evaluated the best model using Mean Squared Error (MSE) and R^2 score.

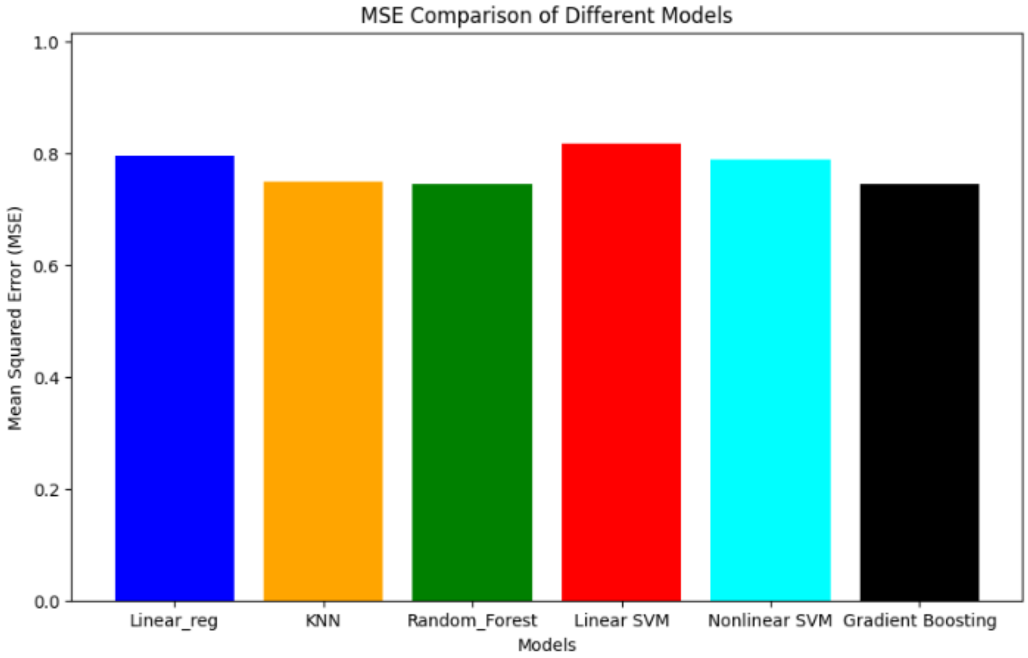


Fig6: Bar plot of MSE values after hyperparameter tuning

Bias -Variance Tradeoff:

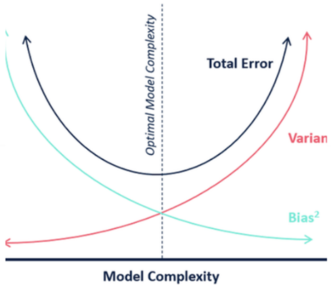


Fig7: Bias-Variance Tradeoff

The mistake caused by the model's oversimplification of its presumptions is known as bias. The amount that the target function's estimate will vary depending on the training set is known as variance.

Cross Validation:

Cross-validation is a statistical method that evaluates and compares machine learning models. It involves splitting data into two segments: one to train the model and the other to validate it. It is a key idea in machine learnings. This is how we determine which machine learning technique is most appropriate for our dataset.

Bi-directional Elimination as a wrapper method:

We used Recursive Feature Elimination (RFE), a wrapper technique for feature selection. RFE is a technique that involves recursively eliminating features according to a selected machine learning model's importance scores. Finding the subset of features that most significantly affects the model's performance is the main goal. To visualize the feature selection process, we created a horizontal bar plot illustrating the ranking of features based on their importance scores obtained through RFE.

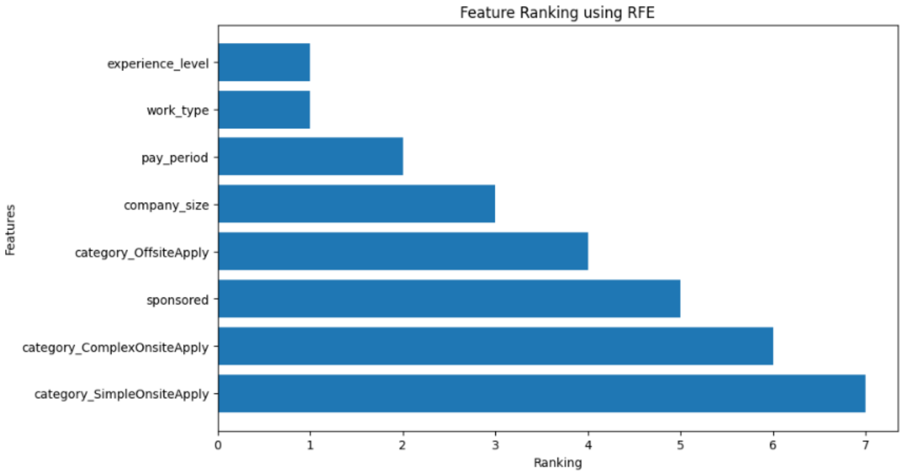


Fig8: Bar plot of Feature Ranking using RFE

ADDITIONAL MOODELS

XGBoost:

The acronym for Extreme Gradient Boosting is XGBoost. It is a potent machine learning algorithm that is frequently used and renowned for its effectiveness and performance. XGBoost is an ensemble learning technique that generates a final prediction that is both robust and accurate by combining the predictions of several weak models, usually decision trees.

XGBoost is a flexible machine learning model that excels at both classification and regression tasks and is well-known for its strong handling of missing data and resistance to outliers. High levels of accuracy and generalization are attained, big datasets are handled with efficiency, and learning curves and feature importance plots are among the visualizations that provide insights into the predictive feature importance and scalability performance with different data sizes.

Extreme Machine Learning Model:

One class of models based on neural networks are Extreme Learning Machines. Unlike conventional neural networks, they have a single hidden layer and fixed, randomly assigned weights between the input and hidden layer. ELMs are renowned for being easy to use and having quick training periods.

The Extreme Learning Machine (ELM) is a simple, quick-to-train model that works well with high-dimensional data. It performs exceptionally well on problems involving lots of features, especially those involving pattern recognition. Among the visualizations are the Hidden Layer Activations that show changes in input feature transformations within the hidden layer and the Decision Boundary Visualization that separates classes.

Basic Deep Learning Model with Two Layers:

An input layer, a hidden layer, and an output layer make up the two layers of a basic neural network architecture represented by this model. Despite being straightforward, it can identify non-linear relationships in the data. Important design decisions that affect the model's performance are the quantity of neurons in the hidden layer and the activation functions that are employed.

When it comes to learning hierarchical representations and capturing complex, non-linear relationships, a simple two-layer deep learning model works incredibly well, especially when applied to image and text data. It requires careful hyperparameter tuning but achieves high accuracy on complex tasks. Model Architecture and Activation Maps are two examples of visualizations that shed light on the structure of the neural network and the hidden layer's reaction to various input patterns.

Ensemble Model containing the Top 3 Models Overall:

To capitalize on the advantages of each individual model, this ensemble model combines the predictions of the top three models (Random Forest, Gradient Boosting, and Deep Learn). By combining various predictions, ensemble methods frequently lead to better generalization performance and robustness.

Ensemble models improve generalization, lessen overfitting, and mitigate the shortcomings of individual models by combining them. They usually attain higher precision, which guarantees improved stability and dependability. Ensemble decision boundaries and model contribution are two examples of visualizations that shed light on how different models work together to influence decision boundaries and contribute to overall predictions.

RESULTS

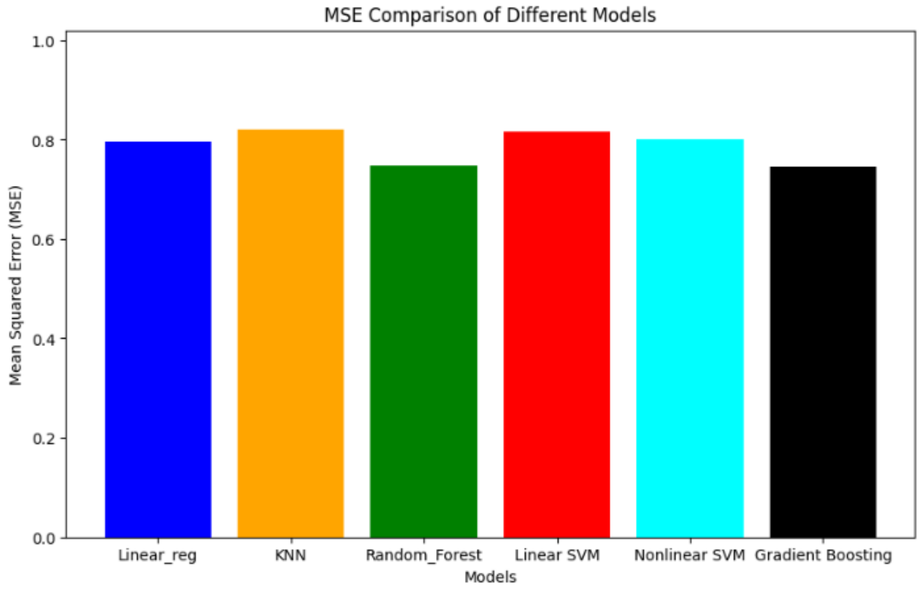


Fig9: Bar plot of the MSE values on the test data

The presented bar plot showcases the Mean Squared Error (MSE) results for models built on the test data. Notably, the Random Forest model demonstrates the lowest MSE among all considered models, indicating superior predictive accuracy. This robust performance can be attributed to the model's unique strength in random subset feature selection, allowing it to extract optimal predictive insights from the dataset. The effectiveness of Random Forest in minimizing MSE underscores its capability to provide accurate predictions and highlights its suitability for the inherent complexities of our dataset.

Below figure is the performance of the best model which is ensemble model. Ensemble model is built with the help of top models that performed very well. With the help of ensemble model, we are able to get the best metrics for predicting our salary.

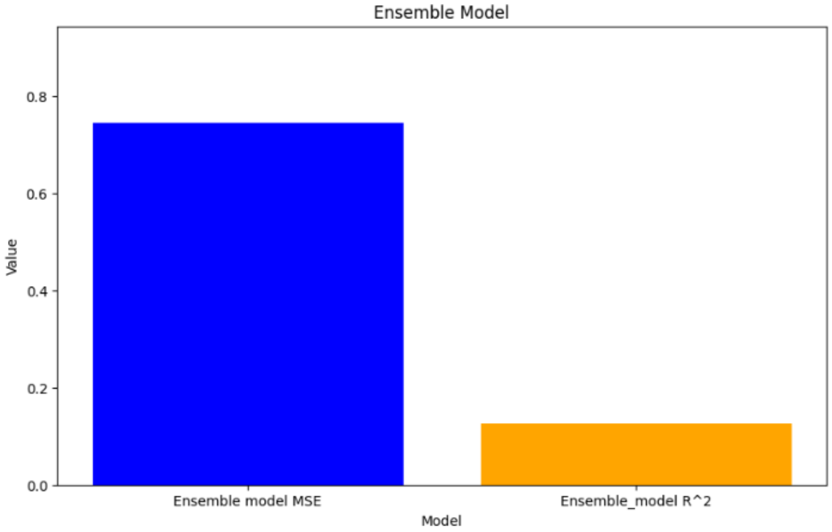


Fig10: Performance metrics of the best model

RESOURCES:

https://scikit-learn.org/stable/index.html

https://scikit-learn.org/stable/modules/grid_search.html

https://en.wikipedia.org/wiki/Lasso_(statistics)

https://scikit-learn.org/stable/modules/ensemble.html

GITHUB LINKS

https://github.com/Vamc-44/Group_9_IDS_project