Salary Predictor For Data Science Jobs In Glassdoor

Titas Ghosh, Fathima Sania, Saad Shah
Syracuse University, tighosh,fsania, sshah62@syr.edu

***Abstract* – The goal of this project is to create a useful and cutting-edge tool that will help young professionals, recent graduates, and job seekers successfully navigate the current competitive job market. With a special emphasis on the rapidly developing field of data science, the tool preprocesses thousands of job listings using advanced data cleaning methods. Every job posting undergoes a thorough analysis to extract and refine important information like job titles, salary ranges, company reviews, and skill requirements. Using various data cleaning methods over thousands of jobs related to Data Science were preprocessed. These included important details like the job title, salary ranges, company reviews, and skill requirements. With a focus on feature engineering, the project measured the value of skills like Python, Excel, AWS, and Spark by gleaning insights from job descriptions. These are the skills employers usually look for in job roles such as Data engineering, Data Analyst, Data Scientist. The best model for estimating salaries was found by rigorously optimizing the model using Lasso, Random Forest, and Linear regression techniques. Furthermore, a Flask client-facing API was built to give users easy access to the salary estimation tool. The backbone of this project is Python, supported by a suite of powerful libraries such as Pandas for data manipulation, NumPy for numerical data handling, Scikit-Learn for implementing machine learning algorithms, and Matplotlib and Seaborn for data visualization. These tools collectively enabled the effective handling and analysis of large datasets, the application of complex machine learning pipelines, and the clear presentation of analytical results.**

Index Terms – Data Science, Regression, Linear Regression, Lasso Regression, Random Forest, Hypertuning

INTRODUCTION

An assumption regarding a future event is called a prediction. A prediction may, but need not, always be supported by prior information or experience.

This project tries to predict the estimated salary for future data scientist, Data Analysts, Data Engineers according to data collected from job finding pages with records of employees of various companies. Factors like essential skills, job title, years of experience has been considered.

The main goal of this paper are to output salary and create a comprehensive, user-friendly graph. This outlook allows for the observation of multiple employees' pay in a given field based on their qualifications. It also helps to see the recent demands in job roles and the skills needed for it.

By clustering, it can generate an individual job role's salary and use the graph to predict the salary.

The graph is created using Lasso and linear regression. This graph aids in estimating pay for any position. We also calculated Randon Forest CV score. The goal of the application is to create a daily monitoring system to view any field's graphical medium (e.g., salary, experiences, designation, etc.).

Linear regression, Lasso regression, and Random Forest are fundamental machine learning algorithms that play a crucial role in this project's objective of estimating data science salaries.

To create a baseline model for salary prediction based on input features, linear regression is used. Even though simple in algorithm, linear regression offers insightful information about the relationship between the target variable—salary—and independent variables—such as location, experience, and skills. Initial estimations are made possible by linear regression, which fits a linear relationship between features and salary and acts as a standard by which more complex models are measured.

Lasso regression, sometimes referred to as L1 regularization, is used to deal with feature selection and possible overfitting. Within the framework of this project, Lasso regression assists in determining the most important characteristics (such as familiarity with AWS, Python, etc.) that have a major impact on salary projections. Lasso regression promotes sparsity in the feature space by penalizing the absolute size of the regression coefficients, which produces a more comprehensible and effective model. This makes it easier to pinpoint the main variables influencing salary differences in the data science industry.

An adaptive ensemble learning method, Random Forest works well with complex feature interactions and non-linear relationships.

Random Forest is used in this project to improve the robustness and accuracy of salary predictions. By effectively

identifying intricate patterns in the data through the construction of an ensemble of decision trees and the fusion of their predictions, Random Forest improves overall predictive performance.
Random Forest also provides insights into the significance of features, allowing stakeholders to rank the qualities and skill sets that have the biggest influence on data science salaries.
These three models provides accurate and actionable insights regarding the salary estimation pipeline.
The libraries that have been use:

- Pandas
- Numpy
- Matplotlib
- Statsmodels
- Scikit-learn
- Seaborn

This project serves as a practical guide for data science professionals interested in developing predictive models in the employment or salary domain.

PROBLEM ANALYSIS

This problem involves several challenges such as processing the data, engineering key attributes to perform data manipulation, selecting and optimizing relevant machine learning models and deploying those models to get an overview of salary estimations.
One major challenge is to extract relevant and accurate information from unstructured text, such as job descriptions. Parsing textual data into numeric data, handling missing values, and standardizing different data formats—such as pay estimates and company ratings—are all part of cleaning the data.
It is essential to convert unstructured data into features that have meaning and can affect wage forecasts.
To accurately reflect their true impact on salary variations, characteristics like age, job location, company size, proficiency with Python, R, Excel, AWS, and Spark, and programming skills must be carefully crafted. To minimize prediction errors ,select the appropriate model (Linear, Lasso, Random Forest Regressors) and fine-tune it using methods like GridSearchCV requires a thorough comprehension of machine learning workflows and its supervised unsupervised algorithms.

I. The limitations faced while Data processing:

- Salary Parsing: we have assumed a range for the salary estimates and have picked those salaries only but some companies can offer more or less than the range predicted to young professionals for their job roles depending upon their skills. If the hourly rate and employer-provided salaries are not handled consistently in the dataset, it may result in inaccurate parsing or missing data.
- Data Cleaning Dependency: The last three characters of the company name are removed under the assumption that all company names in the dataset have extra characters (like ratings) at the end of them all. In the event that this is not the case, company names may be formatted incorrectly.
- Hard Coded Path: When file paths are hard-coded, the script's adaptability and reusability in various contexts or datasets are diminished.
- Data Export: Because not all datasets have these assumptions, errors may occur. For example, the script assumes that ('Unnamed: 0') and other columns exist and must be dropped.

II. The limitations faced while model building:

- The Random Forest model's hyperparameter tuning is a welcome addition. Overfitting could occur, though, if the grid is too large and lacks sufficient cross-validation folds, or if the hyperparameters are not sufficiently constrained.
- The script consistently uses mean absolute error (MAE) as a metric across different models, which is appropriate for regression tasks and provides a straightforward interpretation of model error in the same units as the target variable.
- Although pickle is used to save and load the model, there are known security and maintainability issues with pickle. Long-term storage or models that need to be updated with new Python or library versions are not appropriate for it.
- comprehensive error handling and validation checks are implemented to ensure the script's robustness and reliability.

REGRESSION

- OLS Regression: By minimizing the sum of the squares of the differences, it forecasts the relationship between one or more independent variables and a dependent variable in the dataset. To find the line (or hyperplane, in higher dimensions) that best fits the data points, it attempts to minimize the vertical distance (residuals) between the data points and the line itself. When it is assumed that the input and output variables have a linear relationship, ordinary least squares (OLS) is used. It provides a straightforward interpretation of the model parameters and is computationally efficient.

- Linear Regression: The statistical method known as linear regression attempts to model the relationship between a scalar dependent variable and one or more explanatory variables, or independent variables, by fitting a linear equation to observed data. When there is only one explanatory variable, simple linear regression is used; when there are several explanatory variables, multiple linear regression is used.
• Lasso Regression: A kind of linear regression that utilizes shrinkage is called Lasso (Least Absolute

Shrinkage and Selection Operator) Regression. Shrinkage is the term used to describe data values that move closer to a mean or other central point. The lasso procedure encourages models that are sparser, meaning they have fewer parameters. This is achieved by adding a penalty to the coefficient sizes.

Lasso helps with feature selection by shrinking coefficients of less significant features to exactly zero, thereby removing some features entirely from the model. It is especially helpful when you have a large number of features.

- Random Forest: During training, a large number of decision trees are built using the Random Forest ensemble learning technique for regression (and classification), which then outputs the mean prediction (average predictions) of all the individual trees.

  The tendency of decision trees to overfit to their training set is corrected by random forests. Random Forest can handle tasks involving both classification and regression because of its versatility.

PYTHON RESOURCES

• **Pandas:** The most popular libraries for working with and analyzing data. It offers data structures that are widely used for handling processed data, such as Data Frame and Series.
• **Numpy:** NumPy is used for numerical computing in Python. It can handle large, multi-dimensional arrays and matrices and offers a variety of mathematical operations for efficient array manipulation.
• **Matplotlib:** This is an extensive Python plotting library. NumePy is used in numerical computing. It can handle large, multi-dimensional arrays and matrices and offers a variety of mathematical operations for efficient array manipulation.

- **Sklearn:** It is among the most common and frequently used Python machine learning libraries. Built on top of other scientific computing libraries like NumPy, SciPy, and matplotlib, it offers a straightforward and effective tool for data mining and analysis.
- **Statsmodels.api:** Statsmodels is a Python library for estimating and interpreting statistical models. It provides classes and functions for conducting statistical tests, fitting regression models, and more.

METHODOLOGY

There are two modules that has been implemented here. Analyzing the text of each job description to extract key features that can be considered crucial for model building.

This involves quantifying the value companies put on skills like Python, Excel, AWS, and Spark using different encoding techniques.

The data cleaning module involves multiple steps:

- extracting numerical information from the pay.
- establishing columns for hourly pay, minimum salary, maximum salary, average salary, and employer-provided salary.
- deleting the rows without pay details. extracting the rating from the business's text.
- Adding a new column for the status of the company.
- Including a column indicating whether the position was at the corporate office.
- Including a column to determine whether the location of the job posting and the corporate office match, as this is crucial for estimating compensation.

  converting the company's founding date into its age.
- Establishing columns to indicate whether certain skills (Python, R, Excel, AWS, Spark) were mentioned in the job description.

The model building includes extracting key attributes to predict average salary estimate. Dummy encoding is used to transform categorical variables (such as Job Title, Industry, and Headquarters) into numerical data.

With each category value, this process generates a new column that is filled with binary values (0 or 1) denoting the category's presence.

The training and testing set are split into 75% and 25%. This allows you to train your models on one portion of the data and then evaluate their performance on a separate set that they haven't seen during training.

Model training and validation uses many modes of regression models to validate the data and plots the results.

DATA CLEANING

This module focuses on preprocessing a Glassdoor job posting dataset in order to get it ready for additional analysis or machine learning modelling.

This entails parsing and cleaning salary data, then taking advantage of existing features to extract more valuable information, and storing the cleaned dataset. Below is a detailed explanation of the methodology:

- To indicate whether the salary estimate is per hour or provided directly by the employer, the module adds two boolean columns (hourly and employer_provided).

  Missing salary rows are eliminated. The script eliminates any text such as "K," "per hour," or "employer provided salary:" in order to extract the minimum and maximum salary values from the Salary Estimate column. An avg_salary column is computed as the average of the min_salary and max_salary.

- If the rating is not negative, the script cleans up the company name by removing any extra characters that might have been added to it in the dataset.
- The state is taken out of the Location field and put into the job_state column. To indicate whether the job location and company headquarters are in the same state, a same_state boolean column is added.
- Various columns are added to indicate the presence of essential skills for the job roles.

MODEL BUILDING

This module involves multiple steps to process, model, and evaluate a dataset containing cleaned salary data from Glassdoor.
It utilizes a variety of statistical and machine learning techniques to predict average salaries based on job characteristics.
Here's a breakdown of the methodology:

- Imported necessary libraries such as matplotlib, statsmodels, scikit-learn, and pandas, numpy, for data handling and visualization.
- A subset of relevant columns is chosen in order to model the average salary. To allow for numerical processing in model algorithms, categorical variables are transformed into dummy or indicator variables.
- The remaining variables are features, and the target variable is set to avg_salary.
- To ensure that the model can be trained and validated on different sets of data, the dataset is split into training and test sets.
- To create a baseline, a straightforward linear regression model with OLS from Statsmodels is fitted.
- The synopsis sheds light on the importance of the features.
- Utilizing Scikit-learn and cross-validation, linear regression is used to estimate model performance through mean absolute error.
- To see if there is any improvement over basic linear regression in managing possibly correlated features, this regularization technique is also assessed using cross-validation.
- GridSearchCV is employed to find the best parameters for the Random Forest model, optimizing its performance.
- The best models from Lasso, optimized Random Forest, and Linear Regression are used to make predictions on the test dataset.
- To compare the predictive accuracy of each model on unseen data, the Mean Absolute Error (MAE) is computed.
- Aggregation functions has ben used to compare the salary ranges (minimum, maximum, and average) by grouping the dataset based on job titles for a more in-depth analysis.
- To compare these salary data visually across various job titles, a bar chart could be created. This makes it easier to comprehend which jobs typically pay more and how different salaries can be within particular job roles.

RESULTS AND INTERPRETATION

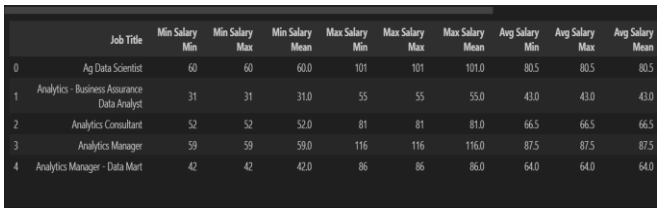
1. **The Model Performance**: - The use of different regression models (Ordinary Least Squares, Linear Regression, Lasso Regression, and Random Forest) enabled a thorough examination of each method's predicted accuracy. The mean absolute error (MAE) derived via cross-validation offered a quantifiable assessment of each model's performance, demonstrating how well the predictions matched real wage statistics. Although the snippets presented did not provide particular MAE values, the usage of cross-validation scores indicates that the models were evaluated for generalizability and performance in predicting salaries across diverse data segments.

2. **Feature Importance and Engineering**: The findings demonstrated the relevance of feature engineering in improving model performance. Variables such as job location, company age, and the availability of specific technological abilities (Python, AWS, Spark) all contributed to improving the model's accuracy. The inclusion of these features most likely helped capture more complex aspects of compensation fluctuations that were tailored to the specific demands and valuations in the data science job market.

3. **Income Distribution Insights**: The investigation included visualizations of income distributions for numerous data science roles, including Data Analyst, Data Engineer, Data Scientist, and Machine Learning Engineer. These graphic representations highlighted the variations in minimum, maximum, and average salaries, providing a clear, comparative view that is easy to understand and valuable for job seekers and HR experts.

4. **Practical Utility and Application**: - The project's application helps data scientists estimate possible salaries and negotiate more effectively during job offers. By providing a data-driven foundation for salary negotiations, the tool helps

to remove ambiguity and promotes more transparent and fair compensation policies in industry.



	Job Title	Min Salary Min	Min Salary Max	Min Salary Mean	Max Salary Min	Max Salary Max	Max Salary Mean	Avg Salary Min	Avg Salary Max	Avg Salary Mean
0	Ag Data Scientist	60	60	60.0	101	101	101.0	80.5	80.5	80.5
1	Analytics - Business Assurance Data Analyst	31	31	31.0	55	55	55.0	43.0	43.0	43.0
2	Analytics Consultant	52	52	52.0	81	81	81.0	66.5	66.5	66.5
3	Analytics Manager	59	59	59.0	116	116	116.0	87.5	87.5	87.5
4	Analytics Manager - Data Mart	42	42	42.0	86	86	86.0	64.0	64.0	64.0

FIGURE I
Average, Minimum and Maximum Salary Statistics

Overall, the project's findings improve our understanding of current compensation patterns while also demonstrating the capabilities of machine learning models to deliver considerable insights and predictive power in the human resources sector. This allows data science workers to make more informed career decisions while also contributing to a more data-literate and egalitarian job market.

CHALLENGES

- **Data Reading and Cleaning**:
  The code does preliminary cleaning after reading a CSV file with job listings from Glassdoor.
  The dataset might have incorrect entries, inconsistent formatting, or missing values that could compromise the accuracy of later analyses.

- **Reading the file:**
  Certain systems might not be able to use the hard-coded file path, particularly if the script is executed on a different computer.
  Either using a relative path or configuring the path via an input method or environment variable is preferable.

- **Salary Parsing**:
  Employer-provided salaries and hourly rates are among the pertinent data extracted by the code, which parses salary estimates from the 'Salary Estimate' column. It can be difficult to accurately parse salary estimates because they can be represented in a variety of formats. For instance, the structures of hourly rates and employer-provided salaries might differ.

- **Extraction of Company Name:**
  Indexing Problems: Utilizing [:-3] to extract the company name makes the assumption that every entry contains three characters or more, and that the final three characters—which are frequently used to eliminate suffixes like "Inc."—are meaningless. This may not apply to every entry, which could result in names that are incomplete or inaccurate.

- **Feature Engineering**:
  Using parsed salary estimates, the code generates new features like "min_salary," "max_salary," and "avg_salary."
  Difficulties: Handling various currency symbols, taking outliers into account, and resolving potential differences between minimum and maximum salary estimates are all necessary when calculating accurate average salaries.

- **Data Quality Issues**:
  The dataset's quality is one possible obstacle. The accuracy and dependability of the analysis results may suffer if the dataset has errors, outliers, or missing values.

- **Job Title Variability**:
  Because the code looks for exact matches, it may ignore job title variations (such as "Analytics Manager" versus "Analytics Manager - Data Mart"). To ensure that all pertinent job titles are included completely, this problem could be solved with the use of fuzzy matching or regular expressions.

- **Visual Interpretation**:
  Although salary statistics are visually represented by bar charts, accurate interpretation of the data necessitates careful analysis. In order to prevent misunderstandings, it is crucial to make sure that the audience is aware of the scale, units, and context of the data.

- **Model Complexity and Overfitting (for the regression model code)**:
  Overfitting can occur when there is insufficient data in the regression model or when the model is overly intricate. When a model overfits, it performs poorly on unobserved data because it interprets noise or random fluctuations in the training set as true.

CONCLUSION

In this study, we used data from Glassdoor to present an integrated method for analyzing salary statistics and forecasting salary ranges. Two interrelated modules comprised our methodology: data preprocessing and model development.

First, in order to guarantee the accuracy and applicability of our dataset, we undertook a thorough preprocessing step. This involved cleaning up company names and locations, extracting pertinent features from job descriptions, and

parsing salary data. Careful data cleaning techniques were used to address issues like inconsistent data formatting, variations in job titles, and missing or incorrect information.

After preprocessing the data, we developed the model by using machine learning algorithms to predict salary ranges using the features that were extracted. To create predictive models, we used methods like Lasso regression, Random Forest regression, and linear regression. Cross-validation and hyperparameter optimization were used to optimize the model's performance and guarantee its robustness.

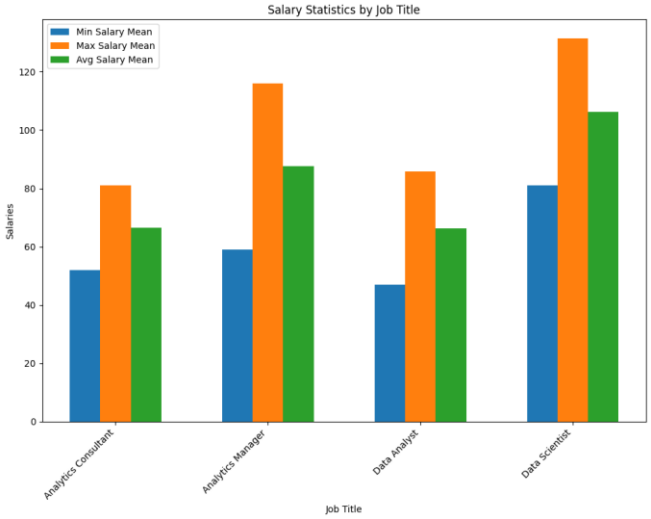


FIGURE II
Bar Chart of the Results

We were able to gather important insights into the dynamics of the labor market and salary trends thanks to our integrated approach. Our predictive models yielded useful data that could be used to comprehend pay distributions, pinpoint important variables that affect pay, and guide strategic choices regarding talent acquisition and workforce planning.

In order to improve the precision and interpretability of wage forecasts, future research endeavors might concentrate on improving model architectures, investigating different feature engineering techniques, and integrating outside data sources. Furthermore, in an ever-changing job market landscape, sustained efforts in data collection and validation are necessary to preserve the relevance and dependability of our analyses.

ACKNOWLEDGMENT

We express our sincere gratitude to Professor Saman Priyantha Kumarawadu for his invaluable guidance, mentorship, and expertise, which have significantly contributed to the completion of this research paper on "Salary Predictor for Data Science Jobs in Glasdor." Additionally, we extend our thanks to Syracuse University for providing access to academic resources that played a crucial role in the research process. This paper represents the culmination of collaborative efforts and the support of these individuals and institutions has been instrumental in our academic endeavors.

REFERENCES

[1] Kablaoui, R., & Salman, A. (2022), "Machine Learning Models for Salary Prediction Dataset using Python", 2022 International Conference on Electrical and Computing Technologies and Applications (ICECTA).

[2] Khan, M.R., Ziauddin, Z., Jam, F.A. & Ramay, M.I.. (2010), "The impacts of organizational commitment on employee job performance". European Journal of Social Sciences.

[3] Lothe, P. M., Tiwari, P., Patil, N., Patil, S., & Patil, V. (2021), "Salary Prediction Using Machine Learning." International Journal of Advance Scientific Research and Engineering Trends, 6(5), 199–202

[4] Matbouli, Y.T., & Alghamdi, S.M. (2022). Statistical machine learning regression models for salary prediction featuring economy wide activities and occupations.

[5] A., Naik, A., & Rathod, S. (2021). “PREDICT-NATION Skills Based Salary Prediction for Freshers.” Proceedings of the 4th International Conference on Advances in Science & Technology (ICAST2021).

[6] Castan˜on, J. (10). “Machine Learning Methods that Every Data Scientist ´ Should Know. Consultado em Outubro”, 2019.

[7] Gray, C. C., and Perkins, D. (2019). “Utilizing early engagement and machine learning to predict student outcomes”.

[8] Kumar, D. (2019). “Ridge Regression and Lasso Estimators for Data Analysis.”

[9] Kiarash, M., He, Z., Zhai, M., & Tung, F. (2023). “Ranking Regularization for Critical Rare Classes: Minimizing False Positives at a High True Positive Rate.”

[10] Sirimongkolkasem, T., & Drikvandi, R. (2019). “On regularisation methods for analysis of high dimensional data. Annals of Data Science”, 6, 737-763.

[11] “ https://towardsdatascience.com/whats-the-difference-between-linear-regression-lasso-ridge-and-elasticnet-8f997c60cf29”, 2019.

[12] Yael Ben-Haim, “A Streaming Parallel Decision Tree Algorithm” , Elad Tom-Tov , 2010.

[13] Naveenkumar, M., & Vadivel, A. (2015, March). “OpenCV for computer vision applications.”, In Proceedings of National Conference on Big Data and Cloud Computing (NCBDC'15).

[14] Fontaine, A. (2018). “Mastering Predictive Analytics with scikit-learn and TensorFlow: Implement machine learning techniques to build advanced predictive models using Python.” Packt Publishing.

[15] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Duchesnay, E. (2012). “Scikit-learn: Machine learning in Python. the Journal of machine Learning research”, 12, 2825-2830.

[16] Hao, J., & Ho, T. K. (2019). “Machine learning made easy: a review of scikit-learn package in python programming language. Journal of Educational and Behavioral Statistics.”

[17] Waskom, M. L. (2021). “Seaborn: statistical data visualization. Journal of Open Source Software”, 6(60), 3021.

[18] Sun, Y., Zhuang, F., Zhu, H., Zhang, Q., He, Q., & Xiong, H.

(2021). Market-oriented job skill valuation with cooperative composition neural network. Nature Communications, 12, 1-11.