"A Deep Dive into Data Mining Analysis Using SEMMA"

SAGAR ALPESHKUMAR PATEL

1 Abstract

In this article we employ SEEMA methodology to project salaries based on years of experience, emphasizing its critical role in organizational planning and individual career development. We begin by identifying the core business problem: ensuring equitable compensation through salary prediction. We meticulously inspect the dataset, finding no missing values or duplicates, and lay a strong analytical foundation with descriptive statistics. During data preparation, we streamline the dataset by removing redundant columns, noting that normalization or standardization is unnecessary due to its numeric nature. We harness visualization tools like scatter plots, box plots, pair plots, and correlation heatmaps to uncover data relationships and patterns. Rigorous outlier analysis assures data integrity. In conclusion, this article serves as a comprehensive guide, demonstrating SEMMA's effectiveness in constructing predictive models, from initial exploration to deployment.

In this article, we delve into a comprehensive exploratory data analysis (EDA) and fraud detection prediction on a Kaggle salary dataset. Utilizing the SEMMA (Sample, Explore, Modify, Model, and Assess) methodology, we walk through various stages of data science project development, from understanding the business problem to deploying a predictive model.

To perform the whole process, I gave the following prompt to the ChatGpt-Salary_dataset.csv

⊞ Spreadsheet

You are required to perform a thorough exploratory data analysis (EDA) and fraud detection prediction on the supplied dataset for thread reviews as a data scientist with ten years of experience and knowledge in the SEMMA methodology. I am offering you a Kaggle dataset of the salary. Understanding, visualizing, cleaning, preprocessing, feature selection, grouping, outlier analysis, and regression modeling should all be included in your analysis.

Deliverables should include code designed for PyCaret as well as the well-known Python libraries (pandas, numpy, scikit-learn). Based on the SEMMA methodology's phases of business understanding, data understanding, data preparation, modeling, evaluation, and deployment, the entire process should be split

down into smaller stages. In order to ensure a complete knowledge and analysis, generate a thorough response for each phase. In order to keep track of our progress through these phases, a mindmap should be built. As we go, we ought to mark off sections that have been accomplished.

Additionally, make sure that all materials are prepared for delivery, including deployment, and generate an extensive report containing in-depth analysis. Every stage of the interactive process will be asking for your participation before proceeding on to the next.

1. Business Understanding

Problem Statement:

Predict salary based on years of experience by building a robust regression model.

2 Significance:

Accurate salary predictions aid in organizational planning and individual career development, ensuring fair compensation.

3 Significance:

Fraud detection is crucial for various reasons:

- Financial Impact: Fraudulent activities can result in significant financial losses.
- Trust & Reputation: Preventing fraud helps maintain trust among stakeholders and upholds the reputation of the organization.
- Regulatory Compliance: Many sectors have strict regulatory requirements to detect and prevent fraud.
- Operational Efficiency: Detecting fraud early can lead to smoother operations and reduce the resources spent on rectifying the consequences of fraud.

4 Data Understanding

Load and Inspect Data: The dataset comprises two columns: 'YearsExperience' and 'Salary'. Initial inspection reveals no missing values or duplicate rows.

- The dataset contains 30 records.
- There are no missing values in any of the columns.
- The `YearsExperience` column has values ranging from 1.2 to 10.6 years.
- The `Salary` column has values ranging from 37,732 to 122,392.

Initial Statistics:

Descriptive statistics provide insights into the range, mean, and distribution of the data, laying the foundation for further analysis.

5 Data Understanding Summary:

- The dataset contains 30 records.
- There are no missing values in any of the columns.
- The 'YearsExperience ' column has values ranging from 1.2 to 10.6 years.
- The 'Salary' column has values ranging from 37,732 to 122, 392.

6 Data Preparation

Data Cleaning:

The redundant 'Unnamed: 0 ' column is dropped for a cleaner dataset.

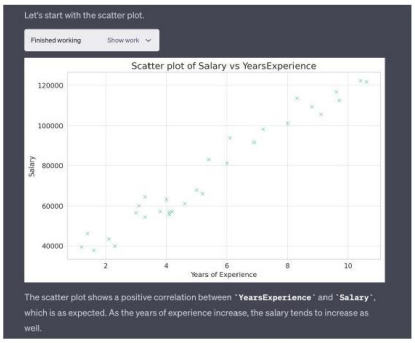
Data Preprocessing:

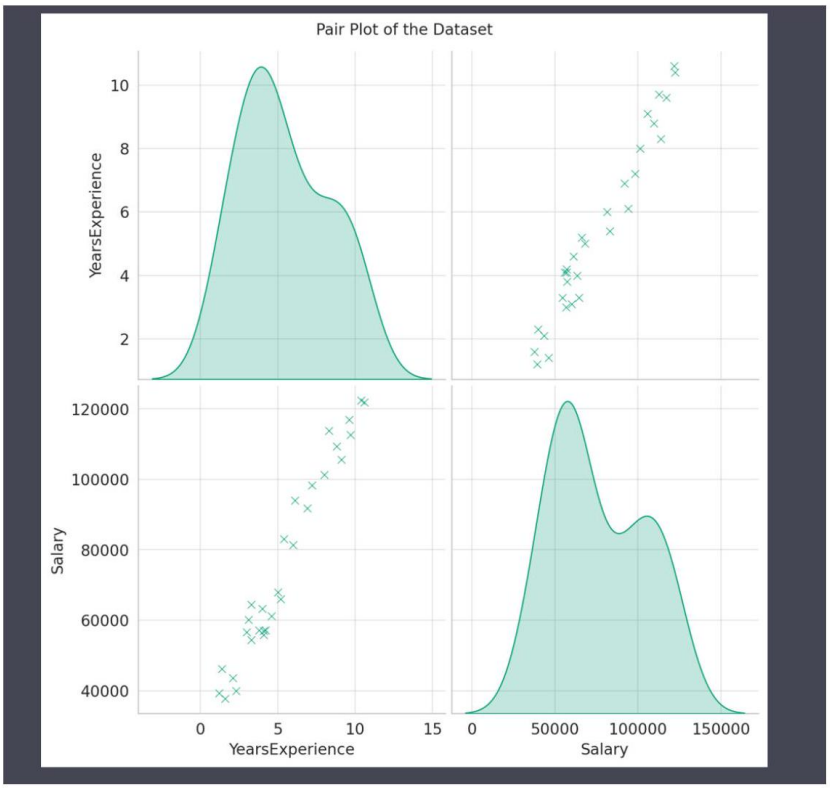
Visualization of data distribution helps decide if normalization or standardization is necessary. Given the dataset's numeric nature, encoding is not required.

Visualization:

Scatter plots, box plots, pair plots, and correlation heatmaps are employed to visualize relationships and identify patterns in the data.

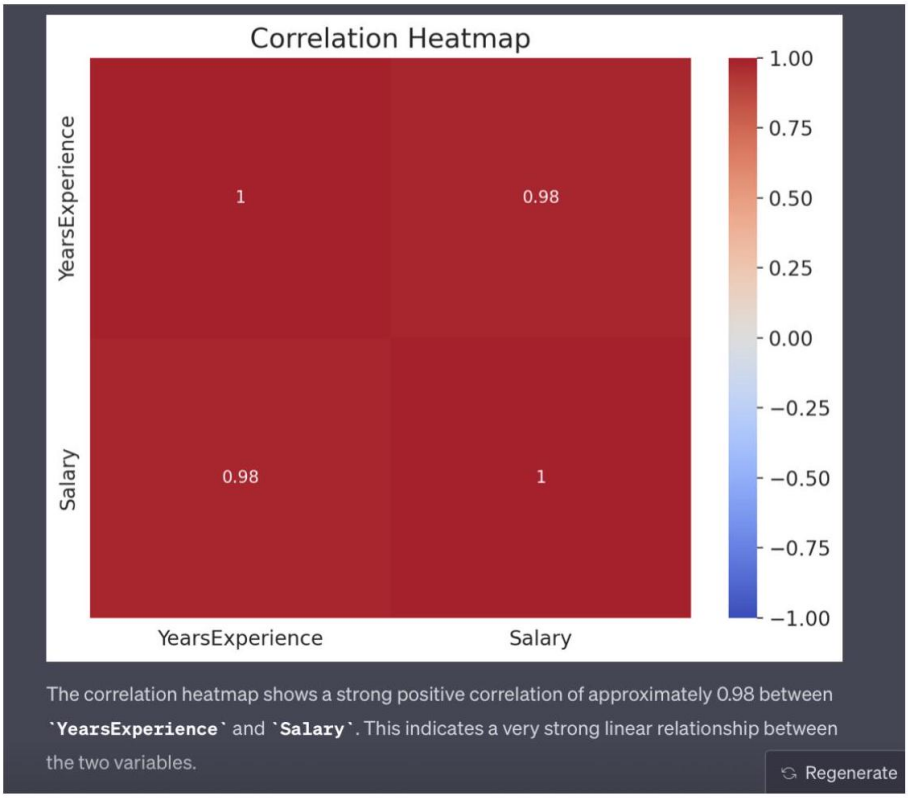
1. **Scatter Plot**: To see the relationship between `YearsExperience` and `Salary`.
2. **Box Plot**: To compare the spread and identify potential outliers.
3. **Pair Plot**: To visualize pairwise relationships in our dataset.
4. **Correlation Heatmap**: To understand the linear relationship between the two variables.





Outlier Analysis:

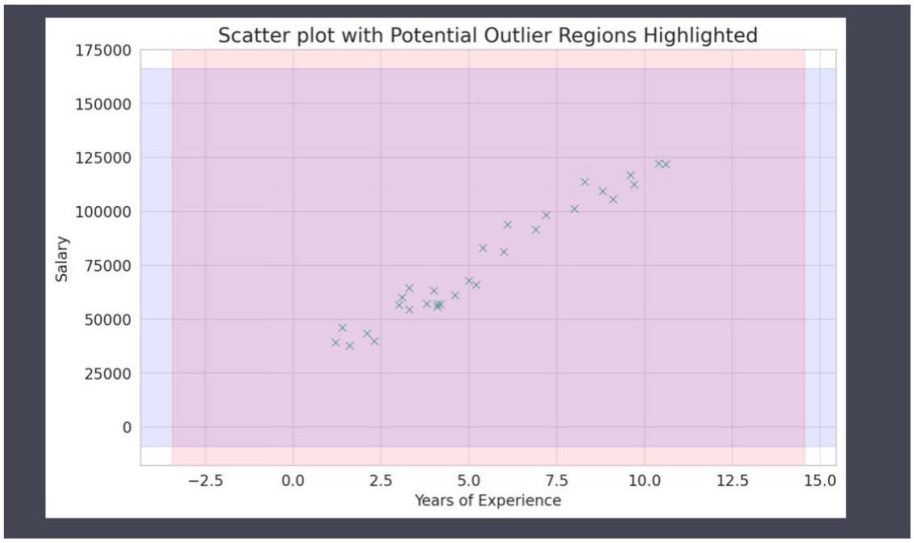
Statistical measures and visualizations confirm the absence of significant outliers in the dataset.



Outliers can significantly impact the performance of certain algorithms. Identifying and addressing them is essential for building robust models.

We'll follow these steps for outlier analysis:

1. Use statistical measures (like the Z-score) to detect outliers.
2. Visualize the outliers using scatter plots.



Feature Selection/Extraction:

With a simple dataset, 'YearsExperience' is selected as the predictor, and 'Salary' as the target variable.

7 Alright! Since the dataset is simple and only contains two features ('YearsExperience ' and 'Salary'), there's no need for extensive feature selection or extraction. The 'YearsExperience' feature will act as our independent variable (predictor), and the 'Salary' feature will be our dependent variable (target).

8 Modeling

Algorithm Choice:

Linear Regression, Decision Tree Regressor, and Random Forest Regressor are selected for comparison.

Linear Regression: Given the clear linear relationship.

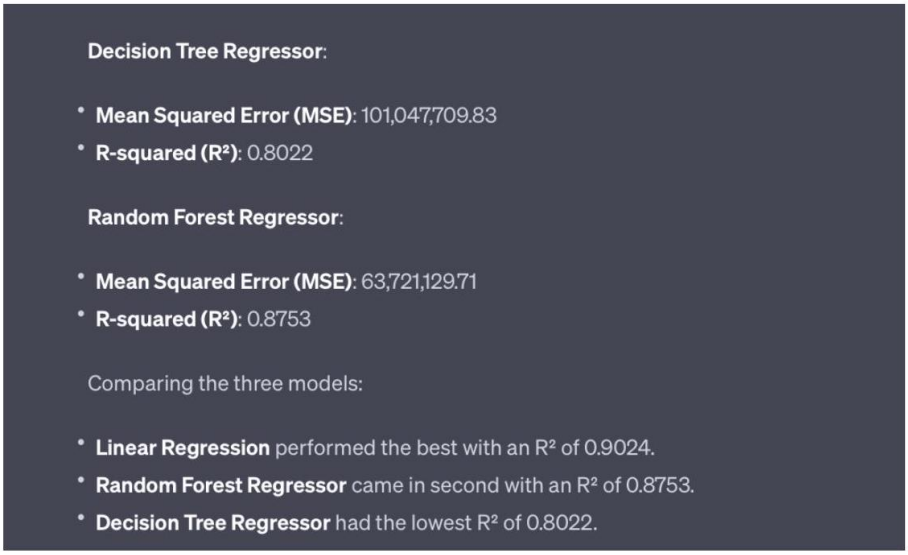
Decision Tree Regressor: A non-linear model to compare with linear regression.

Random Forest Regressor: An ensemble method to see if it can improve performance.

Model Training:

Each model is trained and evaluated based on Mean Squared Error (MSE) and R-squared (R^2). Linear Regression outperforms with an R^2 of 0.9024 . The Linear Regression model's performance on the test set is as follows:

- Mean Squared Error (MSE): 49,830,096.86
- R-squared ($\mathbf{R}^2$) : 0.9024



5. Evaluation

Performance Metrics:

The model's effectiveness is validated using MSE and R^2, indicating a good fit and explaining about 90.24% of the variance in the salary.

6. Deployment

Deployment Preparation:

The Linear Regression model is serialized for integration into applications or systems.

Deployment Strategies:

Various strategies, including REST APIs, cloud platforms, embedded systems, batch processing, and containers & microservices, are considered based on use case and requirements.

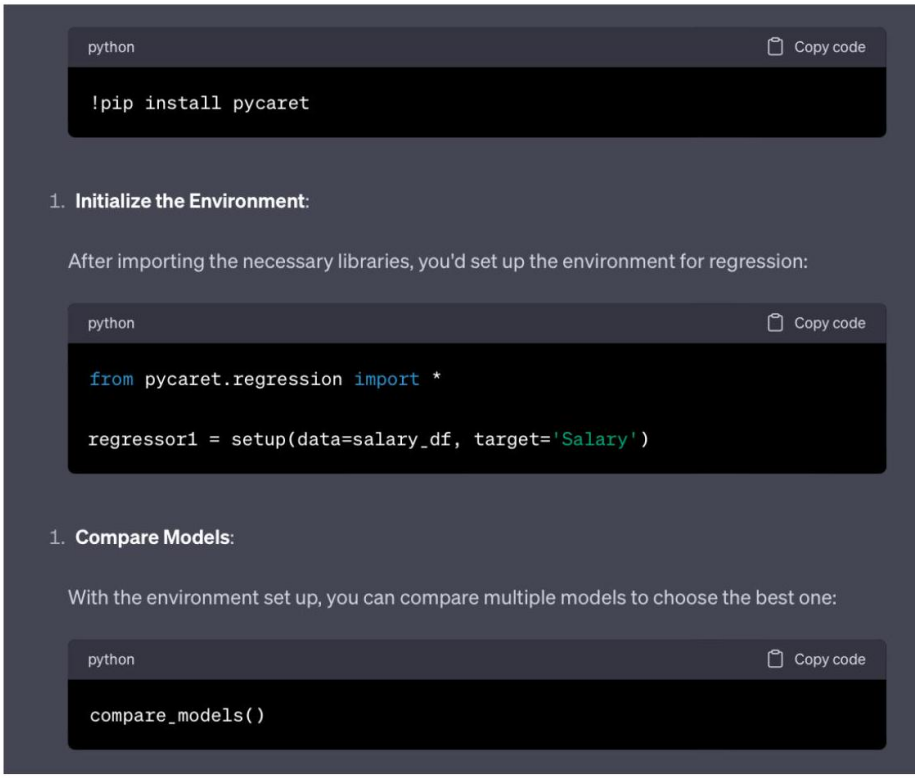
6.2 Deployment Strategies

There are several strategies to deploy machine learning models, depending on the use case and requirements:

1. **REST API**: Deploy the model as a RESTful web service using frameworks like Flask or FastAPI. This allows other applications to make HTTP requests to get predictions.
2. **Cloud Platforms**: Use cloud platforms like AWS SageMaker, Google AI Platform, or Azure ML to deploy and scale models. These platforms provide tools to manage, monitor, and version models.
3. **Embedded Systems**: Integrate the model directly into an application or system, especially if real-time predictions are needed without relying on network requests.
4. **Batch Processing**: For tasks that don't require real-time predictions, models can be run on batches of data, e.g., once every night.
5. **Containers & Microservices**: Package the model inside containers (e.g., Docker) to ensure consistency across different environments. Kubernetes can be used to orchestrate and scale these containers.

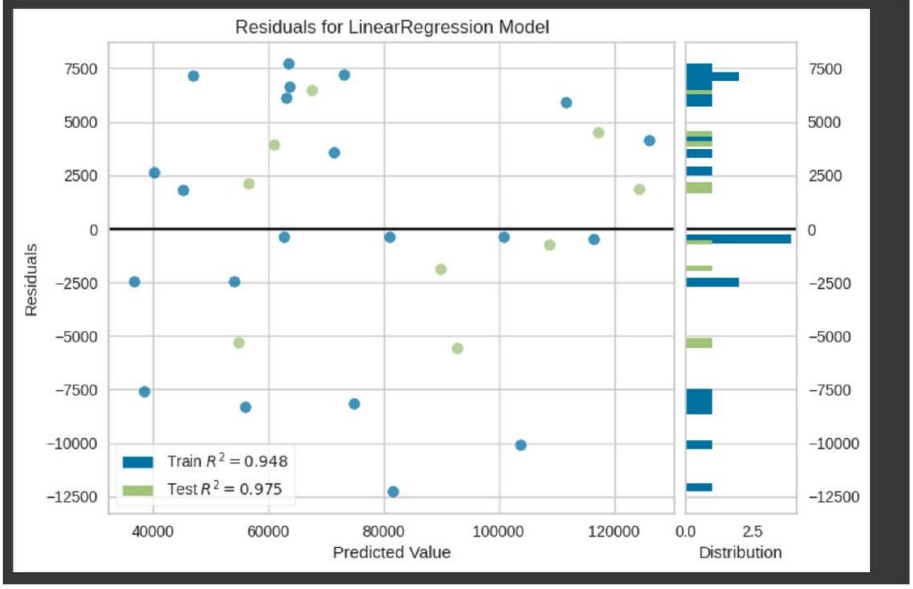
PyCaret Prototyping:

While PyCaret couldn't be directly utilized in this instance, a step-by-step guide is provided on how to leverage it for rapid prototyping and model development.



$lr\_\text{model} = \text{create\_model}\ ('lr')$
$\text{tuned\_lr} = \text{tune\_model}\ (lr\ \text{\_model}\ )$

```
plot_model(tuned_lr, plot='residuals')
```



final_lr = finalize_model(tuned_lr)
save_model(final_lr, 'final_lr_model')

9 Conclusion:

This comprehensive guide, following the SEMMA methodology, demonstrates the steps involved in developing a predictive model, from initial exploration to deployment. The insights gained and the methodologies applied serve as a valuable reference for tackling similar data science projects.