Project Report: High-Accuracy Salary Prediction Model and Web Application

Author: Praveen (Beginner ML Student) **Date:** July 19, 2025 **Project:** End-to-End Machine Learning Model for Salary Prediction

1. Executive Summary

This report documents the end-to-end process of developing a high-accuracy machine learning model to predict employee salaries. The primary objective was to build a reliable predictive tool and deploy it as a user-friendly web application. Starting with a raw dataset, the project involved comprehensive data cleaning, exploratory data analysis, and **advanced feature engineering** to improve model intelligence. This crucial step involved creating new features to represent job seniority and the interaction between experience and job level, which significantly boosted the model's performance. Multiple regression models were trained and evaluated, with the **Random Forest Regressor** being selected as the final model for its superior performance, achieving an **R-squared of 0.92** and a Mean Absolute Error of approximately **$8,542**. The final, optimized model was then successfully deployed in an interactive web application built with Streamlit, demonstrating a complete and successful machine learning project lifecycle.

2. Introduction

2.1. Problem Statement and Motivation

In today's competitive job market, salary estimation is a common challenge for both companies and individuals. A reliable predictive tool can help in setting fair compensation, managing budgets, and providing career guidance. This project was motivated by the desire to apply machine learning fundamentals to a real-world problem, moving beyond theory to create a tangible and useful application. The goal was to build a model that could accurately predict salary based on key professional attributes like experience, education, and job title.

2.2. Project Objectives

- To perform a thorough analysis of the provided employee dataset to identify key factors influencing salary.
- To clean and preprocess the data to make it suitable for machine learning.
- To **engineer new, intelligent features** to improve the model's predictive power and overcome the limitations of a baseline model.
- To train and compare multiple regression models (Linear Regression, Random Forest, Gradient Boosting) to select the one with the highest accuracy.

- To deploy the final, trained model into an interactive web application for easy, real-world use.

2.3. Tools and Technologies

- **Language:** Python
- **Libraries:** Pandas, Seaborn & Matplotlib, Scikit-learn, Streamlit.
- **Dataset:** `salary_data_profession.csv`

3. Methodology: Project Workflow

The project was executed in a series of logical phases, moving from data understanding to final deployment.

Phase 1: Data Exploration and Preprocessing

The first step was to load the `salary_data_profession.csv` dataset. An initial inspection revealed 375 records with 6 columns.

- **Data Cleaning:** The `df.info()` command showed 2 missing values in each column. As this was a small fraction of the data, these rows were dropped using `df.dropna()`. Column names were also renamed for easier access (e.g., `Years of Experience` to `Experience`).

Phase 2: Exploratory Data Analysis (EDA)

To understand the relationships within the data, each attribute was visualized against `Salary`.

- **Numerical Features:** Scatter plots for `Experience` and `Age` showed a strong, positive linear relationship with `Salary`, confirming they were strong predictors.
- **Categorical Features:** Box plots revealed clear trends. Higher `Education` levels corresponded to higher salary ranges. Most importantly, `Job Title` showed a significant impact, but it also highlighted a key challenge: a simple model might not understand the inherent seniority in titles like 'Director' versus 'Junior Developer'.

Phase 3: Feature Engineering (The Key to High Accuracy)

A baseline model performed well but struggled to differentiate between job roles with similar experience levels. This was a critical issue that needed to be solved. To add more "intelligence" to the data, two new features were engineered:

1. **`Job_Level`:** A numerical feature was created by categorizing job titles into seniority levels (e.g., Director=5, Senior Manager=4, Engineer=2). This explicitly taught the model the job hierarchy that is obvious to humans but not to an algorithm.

2. **`Experience_x_Level`:** An interaction feature was created by multiplying `Experience` and `Job_Level`. This captured the crucial business logic that **experience is more valuable at a higher seniority level**.

Phase 4: Model Building and Evaluation

With the engineered features, the data was prepared for modeling by **one-hot encoding** the remaining categorical columns (`Gender`, `Education`). The dataset was then split into training (80%) and testing (20%) sets. Three models were trained and compared.

The **Random Forest Regressor** was selected as the final model due to its lowest Mean Absolute Error, indicating the most precise predictions. The feature engineering successfully improved the R-squared from a baseline of ~0.85 to 0.92.

Phase 5: Deployment

The final, trained Random Forest model and its required column structure were saved to files using `pickle`. An interactive web application was then built using **Streamlit**.

4. Challenges Faced and Solutions

- **Error:** `'streamlit' is not recognized...`
  - **Solution:** Used the command `python -m streamlit run app.py`.
- **Error:** `FileNotFoundError: 'salary_model.pkl'`
  - **Solution:** Corrected the file path in the `app.py` script to `'saved_models/salary_model.pkl'`.
- **Error:** `StreamlitAPIException: set_page_config() can only be called once...`
  - **Solution:** Reordered the script to ensure `st.set_page_config()` was the first Streamlit command.

5. Results and Conclusion

The project successfully achieved its objective. The final Random Forest model, enhanced with feature engineering, demonstrated high accuracy with an **R-squared of 0.92**. The final web application provides a stable, user-friendly, and highly accurate tool for salary estimation.

6. Key Learnings and Reflections

- **The Power of Feature Engineering:** The most significant takeaway was that the quality of the model is heavily dependent on the quality of the features.
- **Importance of EDA:** Visualizing the data was crucial for identifying the initial problem with the `Job Title` feature.

- **End-to-End Workflow:** Successfully moving from a raw CSV file to a deployed web application provides a complete understanding of the project lifecycle.

7. Future Scope

- Train on a larger, more recent, and location-specific dataset.
- Include `Location` as a feature.
- Perform **hyperparameter tuning** on the Random Forest model.

Project Report: High-Accuracy Salary Prediction Model and Web Application

Author: Praveen (Beginner ML Student) **Date:** July 19, 2025 **Project:** End-to-End Machine Learning Model for Salary Prediction

1. Executive Summary

This report documents the end-to-end process of developing a high-accuracy machine learning model to predict employee salaries. The primary objective was to build a reliable predictive tool and deploy it as a user-friendly web application. Starting with a raw dataset, the project involved comprehensive data cleaning, exploratory data analysis, and **advanced feature engineering** to improve model intelligence. This crucial step involved creating new features to represent job seniority and the interaction between experience and job level, which significantly boosted the model's performance. Multiple regression models were trained and evaluated, with the **Random Forest Regressor** being selected as the final model for its superior performance, achieving an **R-squared of 0.92** and a Mean Absolute Error of approximately **$8,542**. The final, optimized model was then successfully deployed in an interactive web application built with Streamlit, demonstrating a complete and successful machine learning project lifecycle.

2. Introduction

2.1. Problem Statement and Motivation

In today's competitive job market, salary estimation is a common challenge for both companies and individuals. A reliable predictive tool can help in setting fair compensation, managing budgets, and providing career guidance. This project was motivated by the desire to apply machine learning fundamentals to a real-world problem, moving beyond theory to create a tangible and useful application. The goal was to build a model that could accurately predict salary based on key professional attributes like experience, education, and job title.

2.2. Project Objectives

- To perform a thorough analysis of the provided employee dataset to identify key factors influencing salary.
- To clean and preprocess the data to make it suitable for machine learning.
- To **engineer new, intelligent features** to improve the model's predictive power and overcome the limitations of a baseline model.
- To train and compare multiple regression models (Linear Regression, Random Forest, Gradient Boosting) to select the one with the highest accuracy.
- To deploy the final, trained model into an interactive web application for easy, real-world use.

2.3. Tools and Technologies

- **Language:** Python
- **Libraries:** Pandas, Seaborn & Matplotlib, Scikit-learn, Streamlit.
- **Dataset:** `salary_data_profession.csv`

3. Methodology: Project Workflow

The project was executed in a series of logical phases, moving from data understanding to final deployment.

Phase 1: Data Exploration and Preprocessing

The first step was to load the `salary_data_profession.csv` dataset. An initial inspection revealed 375 records with 6 columns.

- **Data Cleaning:** The `df.info()` command showed 2 missing values in each column. As this was a small fraction of the data, these rows were dropped using `df.dropna()`. Column names were also renamed for easier access (e.g., `Years of Experience` to `Experience`).

Phase 2: Exploratory Data Analysis (EDA)

To understand the relationships within the data, each attribute was visualized against `Salary`.

- **Numerical Features:** Scatter plots for `Experience` and `Age` showed a strong, positive linear relationship with `Salary`, confirming they were strong predictors.
- **Categorical Features:** Box plots revealed clear trends. Higher `Education` levels corresponded to higher salary ranges. Most importantly, `Job Title` showed a significant impact, but it also highlighted a key challenge: a simple model might not understand the inherent seniority in titles like 'Director' versus 'Junior Developer'.

Phase 3: Feature Engineering (The Key to High Accuracy)

A baseline model performed well but struggled to differentiate between job roles with similar experience levels. This was a critical issue that needed to be solved. To add more "intelligence" to the data, two new features were engineered:

1. **`Job_Level`:** A numerical feature was created by categorizing job titles into seniority levels (e.g., Director=5, Senior Manager=4, Engineer=2). This explicitly taught the model the job hierarchy that is obvious to humans but not to an algorithm.
2. **`Experience_x_Level`:** An interaction feature was created by multiplying `Experience` and `Job_Level`. This captured the crucial business logic that **experience is more valuable at a higher seniority level**.

Phase 4: Model Building and Evaluation

With the engineered features, the data was prepared for modeling by **one-hot encoding** the remaining categorical columns (`Gender`, `Education`). The dataset was then split into training (80%) and testing (20%) sets. Three models were trained and compared.

The **Random Forest Regressor** was selected as the final model due to its lowest Mean Absolute Error, indicating the most precise predictions. The feature engineering successfully improved the R-squared from a baseline of ~0.85 to 0.92.

Phase 5: Deployment

The final, trained Random Forest model and its required column structure were saved to files using `pickle`. An interactive web application was then built using **Streamlit**.

4. Challenges Faced and Solutions

- **Error:** `'streamlit' is not recognized...`
  - **Solution:** Used the command `python -m streamlit run app.py`.
- **Error:** `FileNotFoundError: 'salary_model.pkl'`
  - **Solution:** Corrected the file path in the `app.py` script to `'saved_models/salary_model.pkl'`.
- **Error:** `StreamlitAPIException: set_page_config() can only be called once...`
  - **Solution:** Reordered the script to ensure `st.set_page_config()` was the first Streamlit command.

5. Results and Conclusion

The project successfully achieved its objective. The final Random Forest model, enhanced with feature engineering, demonstrated high accuracy with an **R-squared of 0.92**. The final web application provides a stable, user-friendly, and highly accurate tool for salary estimation.

6. Key Learnings and Reflections

- **The Power of Feature Engineering:** The most significant takeaway was that the quality of the model is heavily dependent on the quality of the features.
- **Importance of EDA:** Visualizing the data was crucial for identifying the initial problem with the `Job Title` feature.
- **End-to-End Workflow:** Successfully moving from a raw CSV file to a deployed web application provides a complete understanding of the project lifecycle.

7. Future Scope

- Train on a larger, more recent, and location-specific dataset.
- Include `Location` as a feature.
- Perform **hyperparameter tuning** on the Random Forest model.