Salary Prediction and Insights Report

Preface:

Explain the dataset, key salary findings, and the predictive modeling workflow used to classify job salaries into ranges. This document is designed for reviewers who want a clear view of what was built, why it matters, and what I think can be improved next.

1. Problem Statement and Goal

The project investigates how compensation varies across the global data science job market and builds a predictive model that estimates a job's **salary range** from job attributes such as experience level, employment type, remote ratio, company size, company location, and job category.

Why this matters:

- Job seekers can compare salary expectations by role category, location, and remote work arrangement.
- Employers and hiring teams can sanity check compensation bands and identify market gaps.
- A working prediction pipeline shows an end-to-end workflow: preprocessing, exploratory analysis, modeling, evaluation and deployment.

2. Dataset Relevance and Scope

The dataset contains job records for data-related roles with salary information adjusted into USD. The modeling notebook works with **93,597** job records (rows) after the preparation steps used in the project. Features used for analysis and prediction (high level):

Field	Type	Meaning (plain language)
experience_level	categorical	Seniority level for the role
employment_type	categorical	Full-time, contract, etc.
job_category	categorical	Grouped job family (Data Science, Data Engineering, etc.)
employee_residence	categorical	Country where the worker is based
company_location	categorical	Country where the company is based
company_size	categorical	Small / Medium / Large company bucket
remote_ratio	categorical/ratio	On-site, hybrid, or fully remote bucket
adjusted_salary	numeric	Salary in USD (target variable for analysis)

3. Key Findings from Exploratory Analysis

This section is covered in depth in a separate report -> **EDA/Summary_Of_Findings.**

4. Predictive Modeling Approach

The predictive task was framed as a multi class classification problem. Instead of predicting an exact salary value, the pipeline predicts which salary band a job belongs to. This is often easier to communicate and aligns with how compensation is discussed in practice.

4.1 Target Construction (Salary Binning)

- Outliers were removed using the IQR rule on numeric columns to reduce distortion from extreme salaries.
- Adjusted salary values were converted into **7** ordered salary ranges using quantile based bin edges. This creates similarly sized groups from low to top.
- The resulting target column (**salary_range**) is what the models learn to predict.

4.2 Feature Encoding and Train/Test Split

- Categorical job attributes were label encoded to convert text categories into integer codes that models can read.
- Features (**X**) were built by dropping adjusted_salary and the salary_range target from the dataframe.
- The target (**y**) was set to salary_range.
- Data was split into training (80%) and test (20%) sets with a fixed random seed for reproducibility.

4.3 Models Tested and Model Selection

Three baseline models were trained and compared using test accuracy:

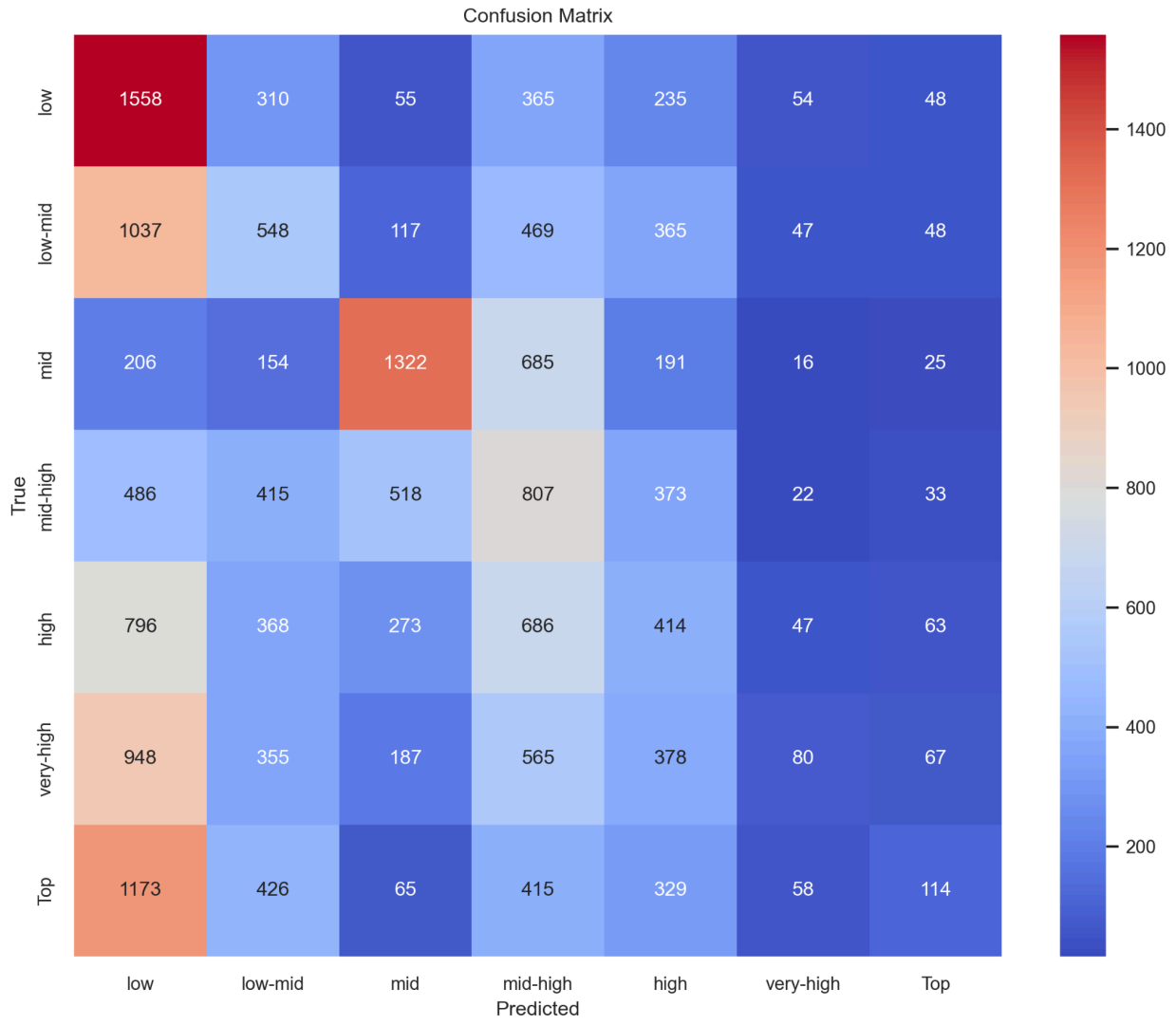
Model	What it is (simple view)	Observed accuracy
Logistic Regression	Linear classifier; strong baseline for multi-class problems	0.23
Random Forest Classifier	Ensemble of decision trees; captures non-linear patterns	0.26
Gradient Boosting Classifier	Boosted trees; often strong on tabular data	0.26

Selection rationale:

The best performing models by accuracy were Random Forest and Gradient Boosting (both 0.26). Random Forest was selected as the best model in the notebook because it matched the top score in that run and provides a practical baseline that can be interpreted with feature importance.

5. Evaluation Results

Evaluation used a confusion matrix and a classification report (precision, recall, and F1 score) for each salary band.



Confusion matrix for the selected model. Stronger predictions appear around the middle bands; extreme bands show more confusion.

```
Classification Report:
              precision    recall  f1-score   support

         low       0.25      0.59      0.35      2625
     low-mid       0.21      0.21      0.21      2631
         mid       0.52      0.51      0.51      2599
    mid-high       0.20      0.30      0.24      2654
        high       0.18      0.16      0.17      2647
   very-high       0.25      0.03      0.06      2580
         Top       0.29      0.04      0.08      2580

    accuracy                           0.26     18316
   macro avg       0.27      0.26      0.23     18316
weighted avg       0.27      0.26      0.23     18316
```

What the metrics indicate:

- Overall accuracy is **0.26**, which is modest for seven classes and indicates the current features do not fully explain salary bands.
- The **mid** band shows the strongest performance (precision about 0.52 and recall about 0.51), meaning the model is better at identifying typical salaries than extremes.
- The **very high** and **top** bands have very low recall (about 0.03–0.04). In practice, the model rarely predicts those classes correctly.
- Many mistakes occur between adjacent bands (for example mid to high predicted as high, or low to mid predicted as low). This is expected because boundaries between salary ranges are soft rather than sharp.

6. Reasons for Performance Limitations

- Salary is driven by variables not present here, such as specific job title, years of experience in years, industry, company revenue, and skills (cloud, ML, security, etc.)
- Label encoding introduces an artificial ordering for categories. For tree models this is usually okay, but for linear models it can be misleading. (One hot) encoding can help in some cases.
- Quantile binning forces equal sized classes but may group very different roles into the same band, especially in the right tail where salaries can jump sharply.