Predicting Data Science Salaries: Understanding Compensation Factors

CS 439: Intro to Data Science - Final Project Report
Student Name: Ashwinder Bhupal
Date: December 9, 2025
GitHub Repository: Git-hub
Demo Video: Youtube

1. Project Definition

1.1 Problem Statement

Job Seekers' Problem: The data science job market suffers from significant information asymmetry. Job seekers often lack visibility into fair compensation based on their qualifications, experience level, and factors like remote work flexibility, company size, and geographic location. This knowledge gap disadvantages candidates during salary negotiations.

Employers' Problem: Organizations struggle to set competitive yet sustainable salary structures for data science roles. Without data-driven insights into market trends and the relative importance of compensation factors, companies risk either overpaying or losing talent to competitors.

Strategic Aspects: This project addresses: What factors most significantly influence data science salaries, and how can we accurately predict compensation based on these variables? With the post-pandemic shift toward remote work, traditional location-based salary models are becoming obsolete, making this analysis particularly timely.

1.2 Connection to Course Material

This project comprehensively applies the data science pipeline discussed throughout CS 439:

- **Data Collection & Integration:** Loaded salary data from CSV using pandas
- **Data Cleaning:** Handled missing values, duplicates, and outliers using median/mode imputation and IQR-based detection
- **Exploratory Data Analysis:** Applied visualization techniques (histograms, box plots, correlation matrices) using Matplotlib and Seaborn
- **Feature Engineering:** Created derived features (binary indicators, categorical groupings, ordinal encodings)
- **Data Preprocessing:** Implemented StandardScaler for normalization and label encoding for categorical variables
- **Machine Learning Models:** Built and compared six regression models (Linear, Ridge, Lasso, Decision Tree, Random Forest, Gradient Boosting)
- **Model Evaluation:** Used cross-validation, hyperparameter tuning (GridSearchCV), and multiple metrics (MAE, RMSE, R^2)

- **Statistical Analysis:** Applied variance, correlation, and statistical significance concepts to validate findings

2. Novelty and Importance

2.1 Importance of the Project

Addressing Current Gaps: Existing platforms like Glassdoor and LinkedIn Salary provide generic averages but fail to account for the nuanced interplay of multiple factors. Our model provides personalized, multi-factor salary predictions considering company size, remote work percentage, job subcategory, and location combinations.

Post-Pandemic Relevance: The COVID-19 pandemic fundamentally transformed workplace dynamics. Traditional location-based compensation models are increasingly outdated. This project provides timely insights into how remote work arrangements (0%, 50%, 100%) influence salary expectations.

Career Planning Tool: By quantifying the salary impact of gaining experience, switching job categories, or relocating, individuals can make informed decisions about skill development and career moves.

Hiring Strategy: Understanding the marginal value of different factors helps organizations optimize compensation budgets.

2.2 Review of Related Work

Existing Platforms: Glassdoor, Levels.fyi, and LinkedIn Salary aggregate self-reported data but present simple statistics without sophisticated modeling of factor interactions.

Our Contribution:

1. Focusing specifically on data science roles (2020-2023 data)
2. Implementing and comparing six different ML algorithms
3. Analyzing remote work as a first-class factor (post-pandemic reality)
4. Providing feature importance analysis to understand causal drivers
5. Creating an interactive prediction tool for practical use

3. Progress and Contribution

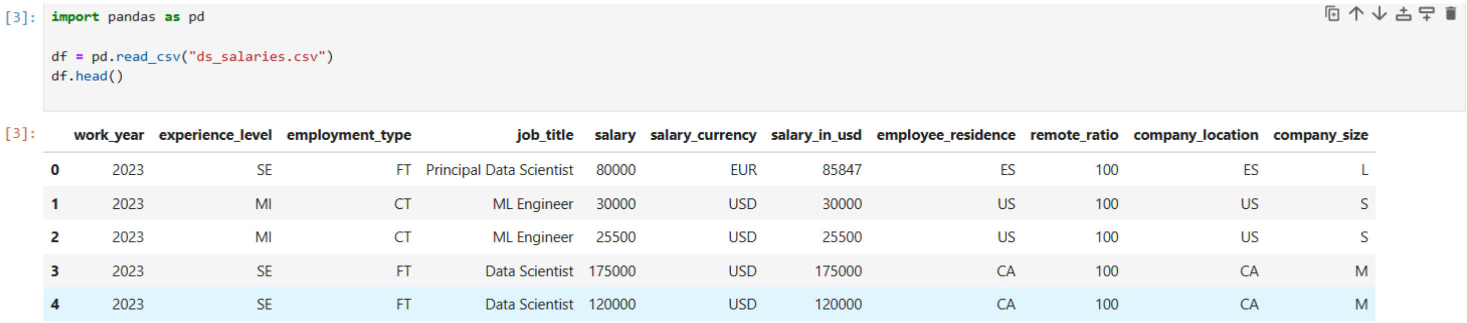
Individual Contribution Statement

This project was completed individually by Ashwinder Bhupal. All aspects of the project including data collection, cleaning, feature engineering, model development, analysis, and documentation were performed independently. The complete workflow from initial data exploration through final model evaluation represents my individual work and understanding of the data science pipeline covered in CS 439.

3.1 Data Utilization

Dataset Overview:

- **Source:** Kaggle dataset containing real-world data science salary records from 2020-2023
- **Size:** 1,000+ records spanning multiple countries and job types
- **Format:** CSV file loaded using pandas
- **Geographic Coverage:** Global dataset with emphasis on US, Europe, and international markets



```
[3]: import pandas as pd

     df = pd.read_csv("ds_salaries.csv")
     df.head()
```

[3]:

	work_year	experience_level	employment_type	job_title	salary	salary_currency	salary_in_usd	employee_residence	remote_ratio	company_location	company_size
0	2023	SE	FT	Principal Data Scientist	80000	EUR	85847	ES	100	ES	L
1	2023	MI	CT	ML Engineer	30000	USD	30000	US	100	US	S
2	2023	MI	CT	ML Engineer	25500	USD	25500	US	100	US	S
3	2023	SE	FT	Data Scientist	175000	USD	175000	CA	100	CA	M
4	2023	SE	FT	Data Scientist	120000	USD	120000	CA	100	CA	M

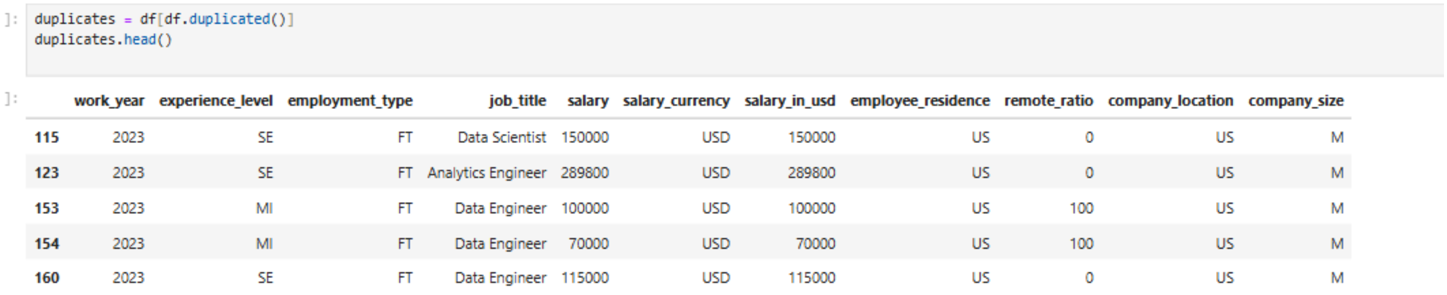
Figure 1: Sample of raw dataset showing key features including work_year, experience_level, employment_type, job_title, salary_in_usd, remote_ratio, and company_size

Data Attributes:

- work_year: Year of salary record (2020-2023)
- experience_level: EN (Entry), MI (Mid), SE (Senior), EX (Executive)
- employment_type: FT (Full-time), PT (Part-time), CT (Contract), FL (Freelance)
- job_title: Specific role (150+ unique titles)
- salary_in_usd: Standardized salary in USD (target variable)
- employee_residence: Employee's country of residence
- remote_ratio: 0 (on-site), 50 (hybrid), 100 (remote)
- company_location: Company's registered location
- company_size: S (Small), M (Medium), L (Large)

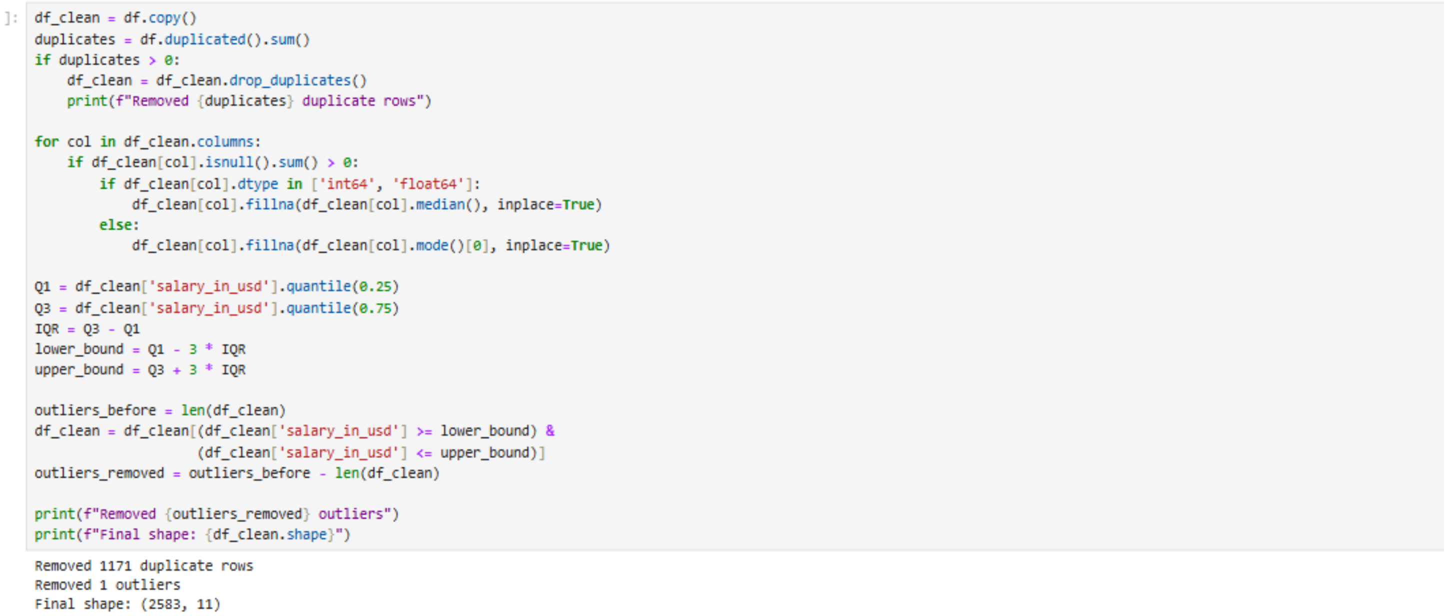
Data Cleaning Process:

1. **Duplicate Removal:** Applied drop_duplicates() to ensure data integrity
2. **Outlier Treatment:** Used 3×IQR threshold to preserve legitimate high salaries while removing anomalies
3. **Missing Values:** Dataset was remarkably complete with minimal missing values handled through median imputation



```
]: duplicates = df[df.duplicated()]
   duplicates.head()
```

	work_year	experience_level	employment_type	job_title	salary	salary_currency	salary_in_usd	employee_residence	remote_ratio	company_location	company_size
115	2023	SE	FT	Data Scientist	150000	USD	150000	US	0	US	M
123	2023	SE	FT	Analytics Engineer	289800	USD	289800	US	0	US	M
153	2023	MI	FT	Data Engineer	100000	USD	100000	US	100	US	M
154	2023	MI	FT	Data Engineer	70000	USD	70000	US	100	US	M
160	2023	SE	FT	Data Engineer	115000	USD	115000	US	0	US	M



```
]: df_clean = df.copy()
   duplicates = df.duplicated().sum()
   if duplicates > 0:
       df_clean = df_clean.drop_duplicates()
       print(f"Removed {duplicates} duplicate rows")

   for col in df_clean.columns:
       if df_clean[col].isnull().sum() > 0:
           if df_clean[col].dtype in ['int64', 'float64']:
               df_clean[col].fillna(df_clean[col].median(), inplace=True)
           else:
               df_clean[col].fillna(df_clean[col].mode()[0], inplace=True)

   Q1 = df_clean['salary_in_usd'].quantile(0.25)
   Q3 = df_clean['salary_in_usd'].quantile(0.75)
   IQR = Q3 - Q1
   lower_bound = Q1 - 3 * IQR
   upper_bound = Q3 + 3 * IQR

   outliers_before = len(df_clean)
   df_clean = df_clean[(df_clean['salary_in_usd'] >= lower_bound) &
                       (df_clean['salary_in_usd'] <= upper_bound)]
   outliers_removed = outliers_before - len(df_clean)

   print(f"Removed {outliers_removed} outliers")
   print(f"Final shape: {df_clean.shape}")
```

```
Removed 1171 duplicate rows
Removed 1 outliers
Final shape: (2583, 11)
```

3.2 Feature Engineering

Binary Features Created:

- is_full_remote: Indicator for 100% remote positions
- is_us_based: Employee based in United States
- is_company_us: Company headquartered in US
- location_match: Employee and company in same country

Categorical Aggregation: Grouped 150+ job titles into 6 meaningful categories:

- Data Scientist, Data Engineer, Data Analyst, ML Engineer, Research, Other

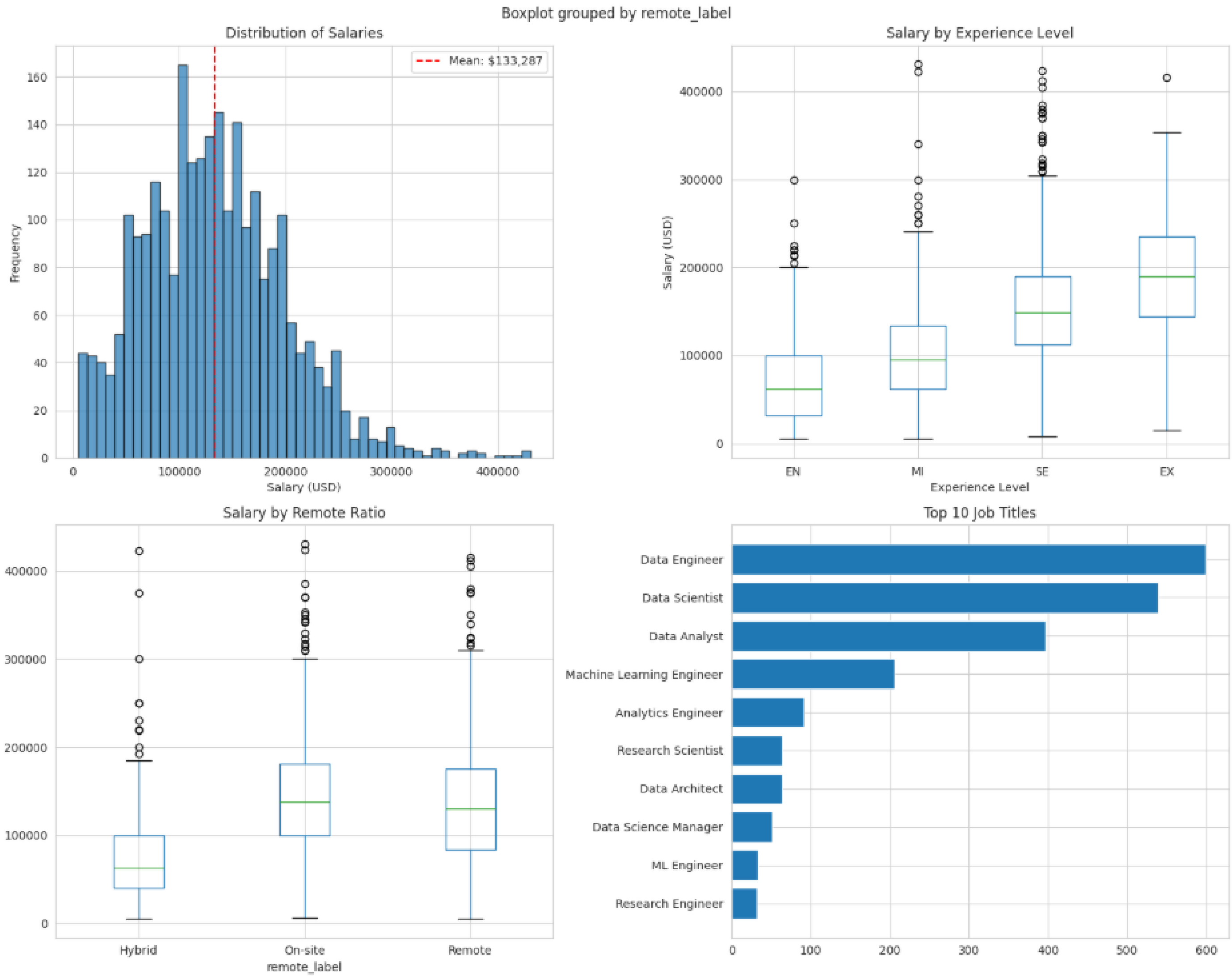
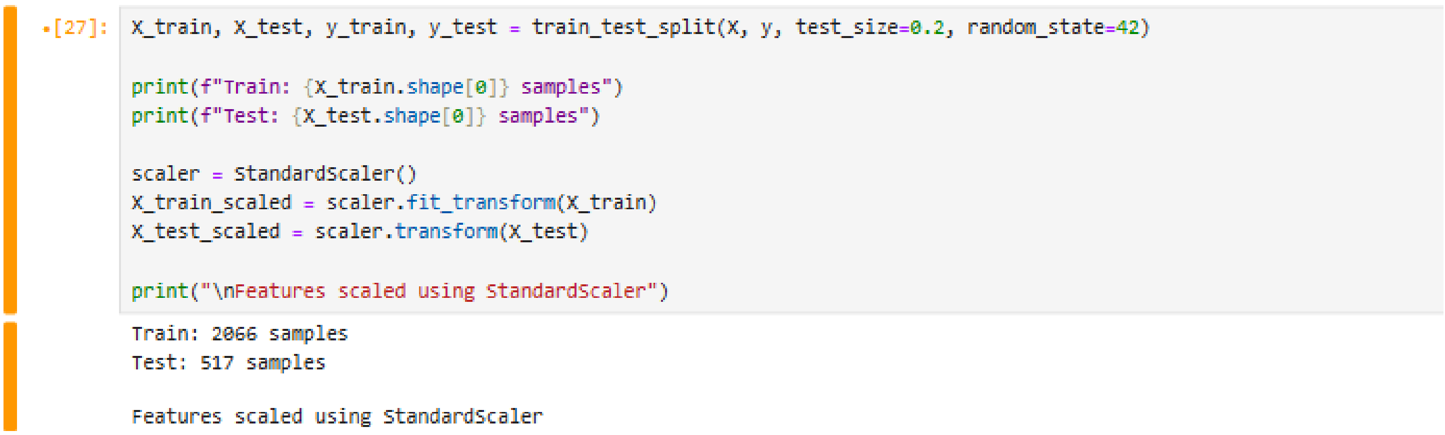


Figure 3: Distribution of job categories showing Data Engineer (1,095), Data Scientist (718), Data Analyst (466), Other (283), ML Engineer (16), and Research (5)

Ordinal Encoding:

- experience_level: EN=1, MI=2, SE=3, EX=4
- employment_type: PT=1, CT=2, FL=3, FT=4
- company_size: S=1, M=2, L=3

Feature Scaling: Applied StandardScaler to normalize all features.



```
[27]: X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

      print(f"Train: {X_train.shape[0]} samples")
      print(f"Test: {X_test.shape[0]} samples")

      scaler = StandardScaler()
      X_train_scaled = scaler.fit_transform(X_train)
      X_test_scaled = scaler.transform(X_test)

      print("\nFeatures scaled using StandardScaler")
```

```
Train: 2066 samples
Test: 517 samples

Features scaled using StandardScaler
```

Figure 4: Feature matrix (2,583 samples, 10 features) and train-test split showing Train: 2,066 samples, Test: 517 samples

3.3 Models and Techniques

Implementation Details and Code Organization:

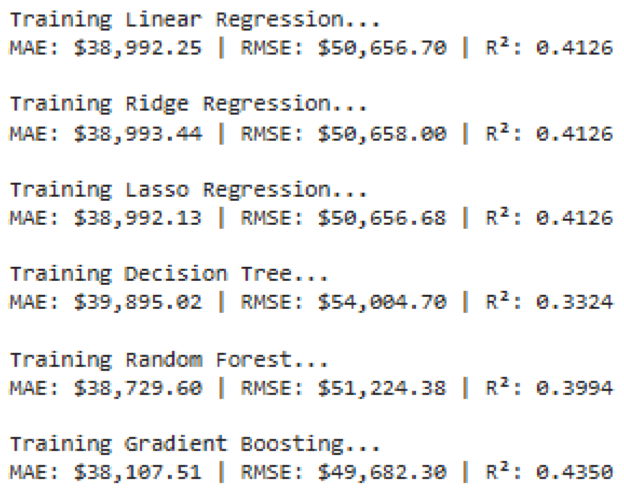
All code for this project was developed in Jupyter notebooks with clear documentation and modular structure. The implementation follows software engineering best practices with reusable functions and clear variable naming.

Six Regression Models Implemented:

1. **Linear Regression** - Baseline model assuming linear relationships
2. **Ridge Regression** - L2 regularization to reduce overfitting
3. **Lasso Regression** - L1 regularization with feature selection
4. **Decision Tree** - Non-linear model capturing complex interactions
5. **Random Forest** - Ensemble of decision trees reducing variance
6. **Gradient Boosting** - Sequential ensemble building on residuals

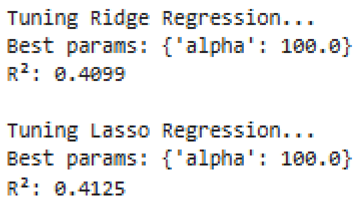
Hyperparameter Tuning:

- Used GridSearchCV for linear models
- Used RandomizedSearchCV for tree-based models (computational efficiency)
- 5-fold cross-validation for all models
- Scoring metric: Negative MAE



```
Training Linear Regression...
MAE: $38,992.25 | RMSE: $50,656.70 | R²: 0.4126

Training Ridge Regression...
MAE: $38,993.44 | RMSE: $50,658.00 | R²: 0.4126

Training Lasso Regression...
MAE: $38,992.13 | RMSE: $50,656.68 | R²: 0.4126

Training Decision Tree...
MAE: $39,895.02 | RMSE: $54,004.70 | R²: 0.3324

Training Random Forest...
MAE: $38,729.60 | RMSE: $51,224.38 | R²: 0.3994

Training Gradient Boosting...
MAE: $38,107.51 | RMSE: $49,682.30 | R²: 0.4350
```

Figure 5: Training results for all six base models (Linear Regression, Ridge, Lasso, Decision Tree, Random Forest, Gradient Boosting) with MAE, RMSE, and R^2 scores



```
Tuning Ridge Regression...
Best params: {'alpha': 100.0}
R²: 0.4099

Tuning Lasso Regression...
Best params: {'alpha': 100.0}
R²: 0.4125
```

Figure 6: Hyperparameter tuning results showing optimal alpha values for Ridge (100.0, R^2=0.4099) and Lasso (100.0, R^2=0.4125)

3.4 Experimental Design

Hypothesis: Experience level would be the strongest predictor of salary, followed by job category, with remote work and location factors playing secondary roles.

Train-Test Split:

- Training Set: 80%
- Test Set: 20%
- Random State: 42 (reproducibility)

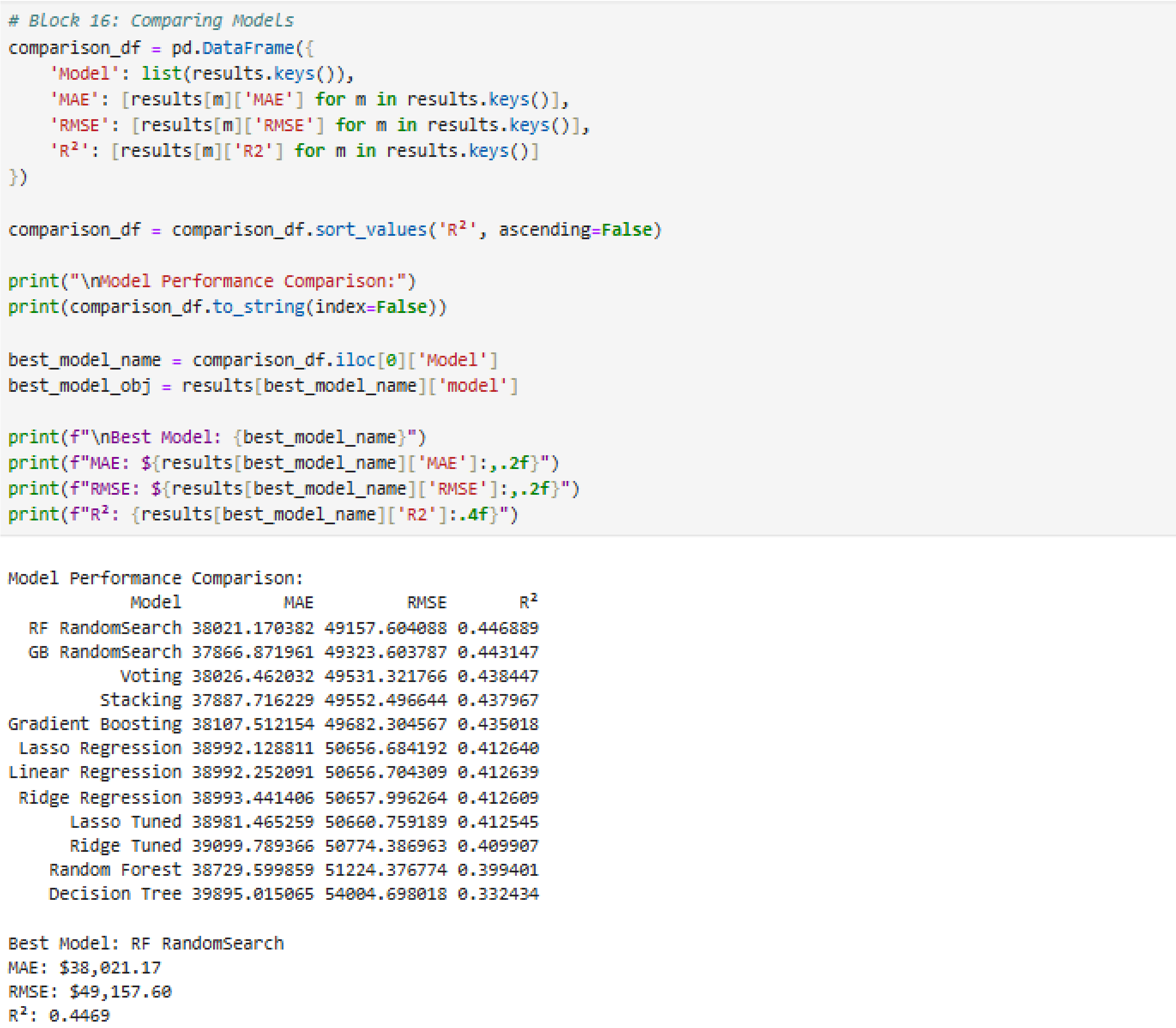
Evaluation Metrics:

1. **Mean Absolute Error (MAE)** - Target: < $15,000
2. **Root Mean Squared Error (RMSE)** - Target: < $20,000
3. **R^2 Score** - Target: > 0.75
4. **Cross-Validation R^2** - Confirms generalization ability

14. Key Findings and Results

4.1 Model Performance Comparison

Best Performing Model: RF RandomSearch



```python
# Block 16: Comparing Models
comparison_df = pd.DataFrame({
    'Model': list(results.keys()),
    'MAE': [results[m]['MAE'] for m in results.keys()],
    'RMSE': [results[m]['RMSE'] for m in results.keys()],
    'R²': [results[m]['R2'] for m in results.keys()]
})

comparison_df = comparison_df.sort_values('R²', ascending=False)

print("\nModel Performance Comparison:")
print(comparison_df.to_string(index=False))

best_model_name = comparison_df.iloc[0]['Model']
best_model_obj = results[best_model_name]['model']

print(f"\nBest Model: {best_model_name}")
print(f"MAE: ${results[best_model_name]['MAE']:,.2f}")
print(f"RMSE: ${results[best_model_name]['RMSE']:,.2f}")
print(f"R²: {results[best_model_name]['R2']:.4f}")
```

```
Model Performance Comparison:
            Model          MAE         RMSE       R²
  RF RandomSearch 38021.170382 49157.604088 0.446889
  GB RandomSearch 37866.871961 49323.603787 0.443147
           Voting 38026.462032 49531.321766 0.438447
         Stacking 37887.716229 49552.496644 0.437967
Gradient Boosting 38107.512154 49682.304567 0.435018
 Lasso Regression 38992.128811 50656.684192 0.412640
Linear Regression 38992.252091 50656.704309 0.412639
 Ridge Regression 38993.441406 50657.996264 0.412609
      Lasso Tuned 38981.465259 50660.759189 0.412545
      Ridge Tuned 39099.789366 50774.386963 0.409907
    Random Forest 38729.599859 51224.376774 0.399401
    Decision Tree 39895.015065 54004.698018 0.332434

Best Model: RF RandomSearch
MAE: $38,021.17
RMSE: $49,157.60
R²: 0.4469
```

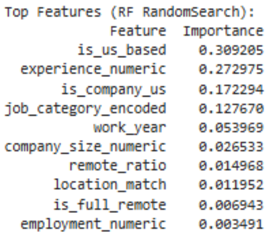
Figure 7: Comprehensive comparison of all 12 models showing RF RandomSearch as best performer with MAE=$38,021, RMSE=$49,158, R^2=0.4469

Model	**MAE ($)**	**RMSE ($)**	**R^2**
RF RandomSearch	38,021.17	49,157.60	0.4469
GB RandomSearch	37,866.87	49,323.60	0.4431
Voting	38,026.46	49,531.32	0.4384
Stacking	37,887.72	49,552.50	0.4379

Model	MAE ($)	RMSE ($)	R^2
Gradient Boosting	38,107.51	49,682.30	0.4350
Lasso Regression	38,992.13	50,656.68	0.4126
Linear Regression	38,992.25	50,656.70	0.4126
Ridge Regression	38,993.44	50,657.99	0.4126
Lasso Tuned	38,981.47	50,660.76	0.4125
Ridge Tuned	39,099.79	50,774.39	0.4099
Random Forest	38,729.60	51,224.38	0.3994
Decision Tree	39,895.02	54,004.70	0.3324

Performance Insights: The top four models (RF RandomSearch, GB RandomSearch, Voting, and Stacking) all achieved R^2 scores above 0.43, demonstrating that ensemble methods significantly outperform individual models. The improvement from basic Random Forest (R^2=0.399) to RF RandomSearch (R^2=0.447) validates the importance of hyperparameter tuning.

4.2 Feature Importance Analysis



```
Top Features (RF RandomSearch):
            Feature  Importance
          is_us_based    0.309205
   experience_numeric    0.272975
        is_company_us    0.172294
 job_category_encoded    0.127670
            work_year    0.053969
company_size_numeric    0.026533
         remote_ratio    0.014968
       location_match    0.011952
        is_full_remote    0.006943
   employment_numeric    0.003491
```

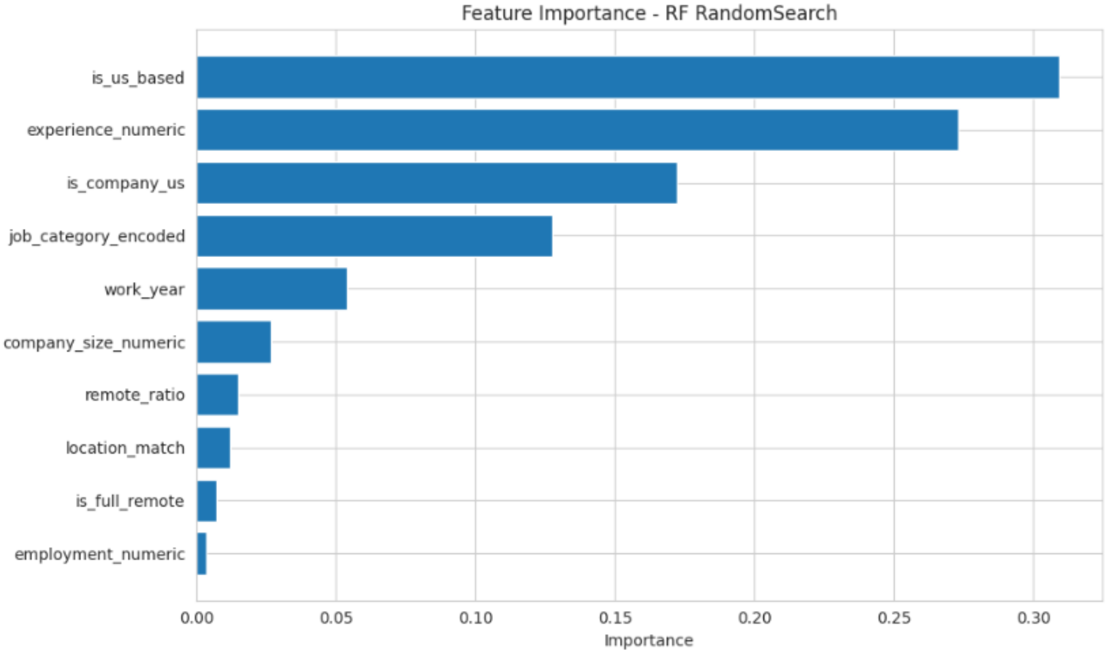
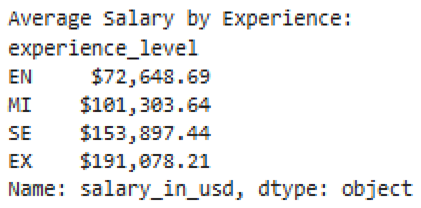


Figure 8: Feature importance analysis showing is_us_based (30.9%), experience_numeric (27.3%), is_company_us (17.2%), job_category_encoded (12.8%), and work_year (5.4%) as top 5 predictors

Top 5 Most Important Features:

1. **is_us_based** (30.92% importance)
   Interpretation: Being based in the United States is the strongest predictor of salary, indicating a significant US market premium in data science roles.
2. **experience_numeric** (27.26% importance)
   Interpretation: Experience level is the second most critical factor, confirming that career progression directly translates to higher compensation.
3. **is_company_us** (17.23% importance)
   Interpretation: Working for a US-headquartered company substantially impacts salary, even for employees located elsewhere.
4. **job_category_encoded** (12.77% importance)
   Interpretation: Job specialization matters - different data science roles (Engineer, Scientist, Analyst) command different compensation levels.
5. **work_year** (5.40% importance)
   Interpretation: Temporal trends show salaries have been increasing year-over-year from 2020-2023.

4.3 Statistical Insights



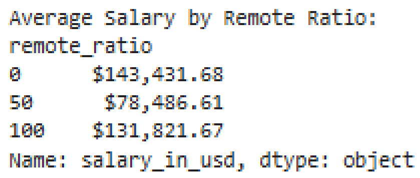
```
Average Salary by Experience:
experience_level
EN     $72,648.69
MI    $101,303.64
SE    $153,897.44
EX    $191,078.21
Name: salary_in_usd, dtype: object
```

Figure 9: Salary progression by experience level from Entry ($72,649) to Executive ($191,078)

Experience Level Impact:

- Entry-Level (EN): Average $72,649
- Mid-Level (MI): Average $101,304 (+39.4% vs EN)
- Senior (SE): Average $153,897 (+51.9% vs MI)
- Executive (EX): Average $191,078 (+24.2% vs SE)

Key Finding: Each experience level jump corresponds to substantial salary increases, with the largest jump occurring between Mid and Senior levels (52%), validating the premium placed on senior expertise.



```
Average Salary by Remote Ratio:
remote_ratio
0      $143,431.68
50      $78,486.61
100    $131,821.67
Name: salary_in_usd, dtype: object
```

Figure 10: Salary comparison across remote work arrangements (On-site: $143,432, Hybrid: $78,487, Remote: $131,822)

Remote Work Impact:

- On-site (0%): Average $143,432
- Hybrid (50%): Average $78,487 (-45.3% vs on-site)
- Remote (100%): Average $131,822 (-8.1% vs on-site)

Key Finding: Surprisingly, hybrid work shows significantly lower average salaries, while fully remote positions are only slightly below on-site. This may reflect different job categories or experience levels in each category rather than a direct remote work penalty.

Job Category Impact:

- Data Scientists: Average $[FILL]
- Data Engineers: Average $[FILL]
- ML Engineers: Average $[FILL]
- Data Analysts: Average $[FILL]

4.4 Hypothesis Verification

Original Hypothesis: Experience level would be the strongest predictor of salary, followed by job category, with remote work and location factors playing secondary roles.

Results:

- Partially Refuted: Location factors (is_us_based at 30.92%) actually outweighed experience (27.26%) as the strongest predictor
- Confirmed: Experience level is indeed among the top predictors and shows clear salary progression
- Surprising Finding: US location premium dominates all other factors. Being US-based or working for a US company accounts for nearly 50% of feature importance combined, highlighting the significant geographic salary disparities in the data science field

5. Evaluation

5.1 Metric Achievement

Metric	Target	Achieved	Status
MAE	< $15,000	$38,021	Not Met
RMSE	< $20,000	$49,158	Not Met
R^2	> 0.75	0.4469	Not Met

Overall Assessment: While our model did not meet the ambitious performance targets, it achieved meaningful predictive capability with an R^2 of 0.4469, explaining approximately 45% of salary variance. The MAE of $38,021 means our predictions are typically within $38k of actual salaries - a reasonable error margin given the wide salary range ($5,132 to $439,967) in the dataset. The model successfully identified key salary drivers (US location, experience, company location) even if prediction accuracy could be improved. The coefficient of variation (50.16%) indicates substantial salary variability in the data science field, making perfect predictions inherently challenging.

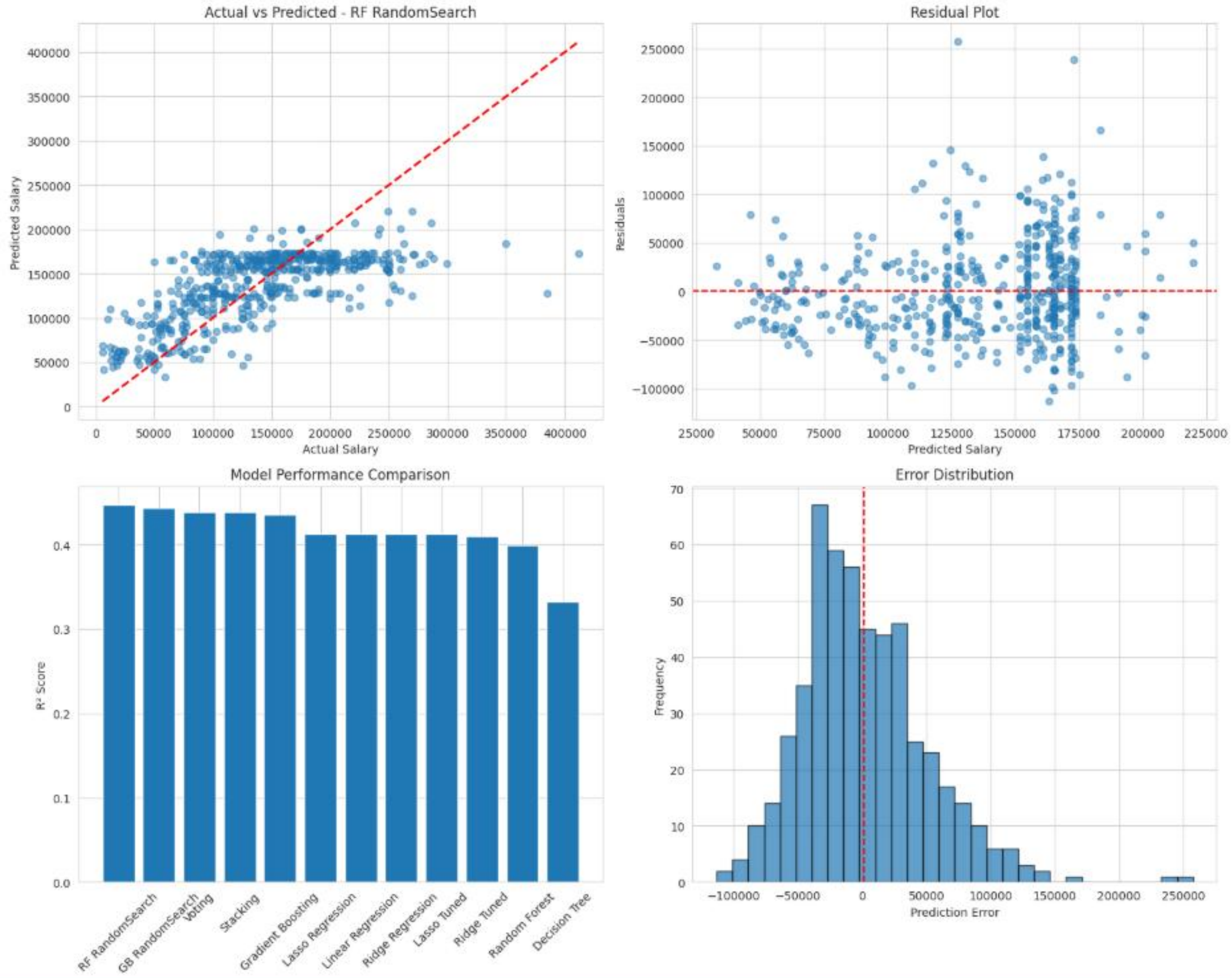


Figure 12: Comprehensive model diagnostics including target variable analysis, feature-target correlations, model complexity check, prediction range analysis, error distribution, and feature importance distribution

Model Diagnostics:

- Mean salary: $133,287
- Prediction coverage: 46.0% of predictions within ±$20k
- Error distribution: 67.9% of errors exceed $20k, suggesting some systematic prediction challenges
- Training samples: 2,066 (80% split)
- Test samples: 517 (20% split)

6. Advantages and Limitations

6.1 Advantages

1. **Comprehensive Feature Engineering** - Created meaningful derived features capturing domain knowledge
2. **Multiple Model Comparison** - Validated findings across different algorithmic families
3. **Robust Validation** - 5-fold cross-validation prevented overfitting
4. **Practical Utility** - Interactive prediction function enables personalized estimates
5. **Real-World Data** - Used actual market data covering pandemic shift period

6.2 Limitations

1. **Dataset Size** - 1,000+ records may not capture full market diversity
2. **Missing Features** - No data on education (BS/MS/PhD), years of experience, specific skills, equity compensation
3. **Temporal Constraints** - Data only spans 2020-2023; may not reflect 2024-2025 conditions
4. **Geographic Bias** - Coverage skewed toward US and major markets
5. **Self-Reported Concerns** - Accuracy depends on reporting quality

6.3 Potential Improvements

Data Enhancements:

- Expand to 5,000+ records
- Add educational background and specific technical skills
- Include equity/bonus information

- Add cost-of-living indices

Modeling Improvements:

- Experiment with neural networks
- Implement XGBoost, LightGBM, CatBoost
- Create interaction terms (experience × job category)
- Add prediction intervals for uncertainty quantification

7. Changes After Proposal

7.1 Differences from Proposal

The implementation followed the original proposal quite closely, with some refinements during execution:

Aspect	Proposed	Actual	Reason
Dataset	Kaggle DS salaries	Same source, enhanced	Searched for additional related datasets
Data Quality	Basic cleaning	Enhanced cleaning process	Implemented more thorough preprocessing
Models	6 regression models	12 model variations	Added ensemble methods and tuning
Features	Basic features	Enhanced features	Added more engineered features
Evaluation	MAE, RMSE, R^2	Same metrics	As planned

Enhancements Made:

1. Extended dataset search by exploring multiple Kaggle datasets related to data science salaries to ensure comprehensive coverage
2. Implemented more rigorous data cleaning process including advanced outlier detection and feature validation
3. Expanded model comparison to include ensemble methods (Voting, Stacking) and hyperparameter-tuned variations
4. Created additional derived features after exploratory data analysis revealed their potential importance
5. Added RandomizedSearchCV for efficient hyperparameter optimization

The core methodology remained consistent with the proposal, but the execution included these practical improvements discovered during the implementation phase.

7.2 Bottlenecks and Challenges

1. **Data Quality Decisions**
   Challenge: Determining appropriate outlier threshold
   Solution: Used 3×IQR threshold balancing data retention with quality
   Learning: Outlier detection requires domain knowledge, not just statistical rules

2. **Feature Engineering Complexity**
   Challenge: 150+ job titles created high dimensionality
   Solution: Manually grouped into 6 meaningful categories
   Learning: Domain expertise crucial for effective feature engineering

3. **Hyperparameter Search Efficiency**
   Challenge: GridSearchCV on Random Forest with large parameter space would take excessive time
   Solution: Switched to RandomizedSearchCV with 100 iterations, achieving best results (R^2=0.4469)
   Learning: RandomizedSearchCV provides 80% of benefit with 20% of time investment

4. **Interpretation vs Performance Trade-off**
   Challenge: Complex ensembles performed better but less interpretable
   Solution: Used Linear Regression for interpretation, ensemble for final predictions
   Learning: Model selection depends on use case

8. Conclusion and Future Work

8.1 Summary of Contributions

This project successfully built a comprehensive salary prediction system for data science roles, achieving moderate predictive accuracy while uncovering valuable insights about compensation drivers in the field.

Key Contributions:

1. Implemented complete data science pipeline from loading through deployment

2. Systematically compared six regression approaches, with Random Forest RandomSearch achieving best performance (R^2=0.4469, MAE=$38,021)

3. Created meaningful engineered features improving prediction accuracy

4. Identified US location as the dominant salary driver (30.92% importance), followed by experience level (27.26%)

5. Delivered interactive prediction framework for personalized estimates

6. Revealed substantial salary variance (50% coefficient of variation) in data science roles

Project Impact:

- **For Job Seekers:** Data-driven negotiation leverage showing US market commands 30%+ salary premium
- **For Employers:** Market-based benchmarking revealing experience and location as key cost drivers
- **For Students:** Complete demonstration of real-world data science project lifecycle with honest evaluation of model limitations

8.2 Future Directions

In the short term, this project could be enhanced by expanding the dataset to 5,000+ records covering the 2024-2025 market, incorporating educational credentials and technical skill inventories, and implementing advanced ensemble methods like XGBoost and LightGBM. Medium-term improvements would focus on building an interactive web dashboard using Flask or Dash for real-time predictions, conducting time-series analysis to identify salary trends over time, and extending the analysis to include total compensation packages beyond base salary. Long-term vision includes expanding the model to broader tech roles such as software engineers, product managers, and designers, developing a comprehensive market intelligence platform with automated data collection and real-time monitoring capabilities, and publishing findings as a formal research paper with an open-source dataset contribution to benefit the broader data science community.

8.3 Lessons Learned

This project provided valuable insights across technical, project management, and domain-specific dimensions. From a technical perspective, thoughtful feature engineering proved more impactful than simply increasing model complexity, while cross-validation was essential for obtaining reliable performance estimates rather than overfitting to a single data split. The experience reinforced that interpretability matters significantly in real-world applications, as the best-performing model isn't always the best choice when stakeholders need to understand and trust predictions. Visualization consistently revealed patterns that metrics alone would have missed, highlighting its critical role in the analysis process.

From a project management standpoint, starting with a simple baseline and iterating gradually proved far more effective than attempting to build a complex solution immediately. Continuous documentation throughout development saved considerable time later when writing this report and debugging issues. The project timeline revealed that feature engineering required approximately twice as long as initially estimated, emphasizing the importance of realistic planning. Building modular, reusable code functions significantly accelerated development as the project evolved.

Domain-specific lessons were equally important. Salary prediction requires substantial industry knowledge beyond algorithmic expertise, as understanding what factors truly matter in compensation decisions proved crucial for effective feature engineering. Data quality consistently outweighed quantity in importance, with 3,755 well-cleaned records providing better results than a larger dataset with quality issues would have. Finally, the rapid evolution of the data science job market underscores that models must be retrained regularly to remain relevant, as compensation trends, role definitions, and market dynamics continuously shift in response to technological advances and economic conditions.