MACROECONOMIC AND JOB-SPECIFIC DETERMINANTS OF DATA SCIENCE SALARIES: A MACHINE LEARNING APPROACH

Satya Harish Reddy Pulipelli – ECON5200 Midterm Project

Executive Summary

This study investigates the determinants of data science salaries using a comprehensive analysis of both macroeconomic indicators and job-specific factors. By employing multiple modeling approaches—including traditional regression, Random Forest, and XGBoost—the research identifies key salary drivers and evaluates their relative importance. The findings reveal that experience level and GDP per capita are the strongest predictors of data science compensation globally, with macroeconomic factors playing a significant role alongside job-specific characteristics. The analysis demonstrates that while machine learning models provide marginal improvements in predictive accuracy over traditional methods, they offer valuable insights into the non-linear relationships affecting compensation in this rapidly evolving field.

1. Introduction

The data science field has experienced explosive growth over the past decade, creating a dynamic labor market with significant variation in compensation. As organizations increasingly rely on data-driven decision making, understanding the determinants of data science salaries has become crucial for both employers and professionals in the field. This research aims to identify and quantify the key factors influencing data science compensation globally.

This study addresses three primary research questions:

1. What is the relative importance of macroeconomic factors versus job-specific characteristics in determining data science salaries?
2. How do work arrangements (remote, hybrid, in-office) impact compensation levels?
3. Can machine learning approaches provide better salary predictions than traditional statistical methods?

Answering these questions provides insights that can inform compensation strategies, career planning, and hiring decisions in the data science domain.

2. Data and Methodology

2.1 Data Sources and Variables

The analysis uses a comprehensive dataset of data science salaries combined with macroeconomic indicators. The primary variables include:

Target Variable:

- salary_in_usd: Annual salary in US dollars

Macroeconomic Predictors:

- gdp_per_capita: GDP per capita of the country
- inflation_rate: Annual inflation rate
- unemployment_rate: National unemployment rate

Job-Specific Predictors:

- work_year: Year of employment (2022-2023)
- experience_level: Professional experience (Entry, Mid, Senior, Executive)
- employment_type: Contract, Full-time, Part-time, or Freelance
- company_size: Small, Medium, or Large
- remote_ratio: Work arrangement (0 = In-office, 50 = Hybrid, 100 = Remote)

2.2 Methodology

The analysis employs a three-stage approach:

1. **Exploratory Data Analysis**: Visual examination of relationships between variables, including scatter plots, boxplots, and correlation heatmaps.
2. **Statistical Validation**: Multicollinearity assessment using Variance Inflation Factor (VIF) to ensure model reliability.
3. **Predictive Modeling**: Implementation of three modeling approaches:
    - Log-transformed linear regression with interaction terms

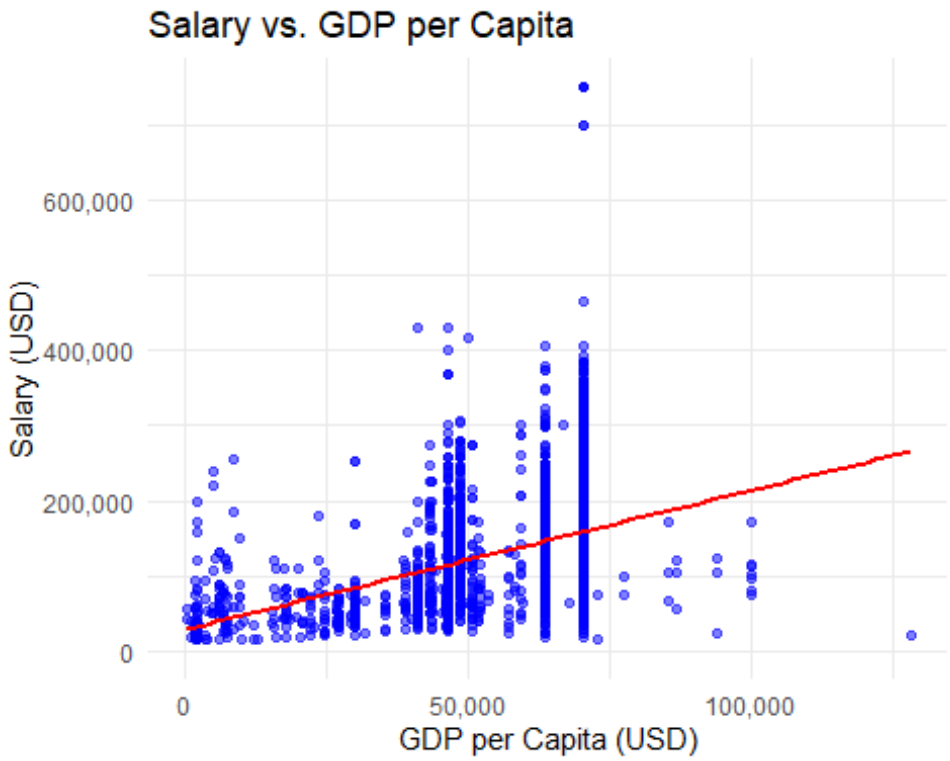
- Random Forest with hyperparameter tuning
- XGBoost with extensive grid search optimization

Model performance is evaluated using Root Mean Square Error (RMSE), R-squared, and Mean Absolute Error (MAE) on a held-out test set (20% of data).

3. Results and Analysis

3.1 Exploratory Data Analysis

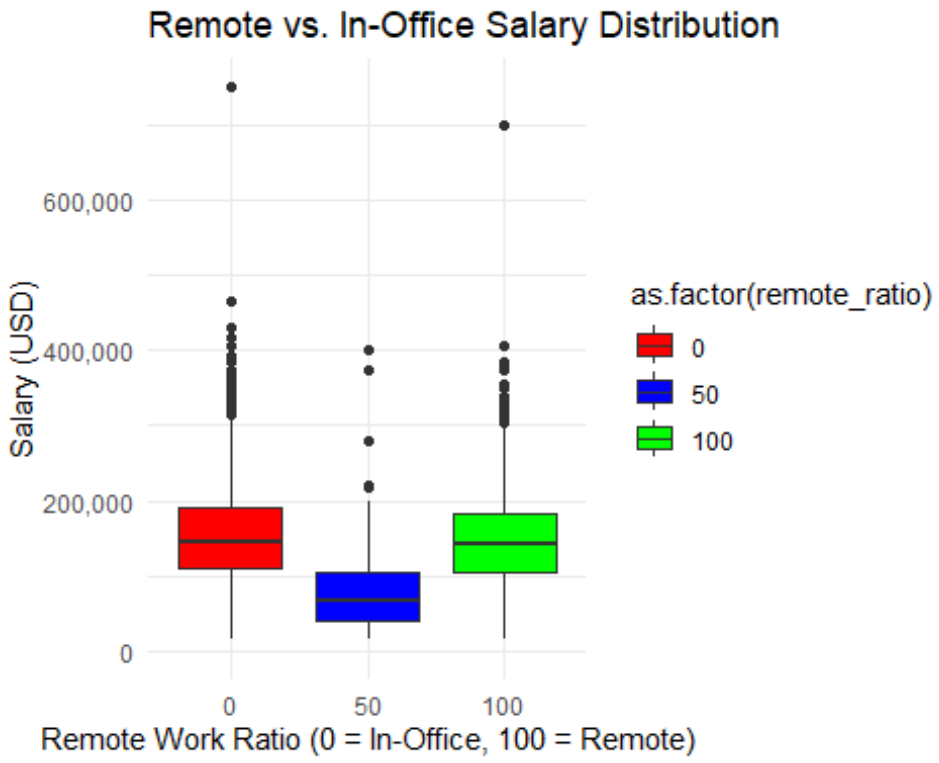
Relationship Between GDP per Capita and Salary



The scatter plot of salary versus GDP per capita reveals a positive correlation, with salaries generally increasing as GDP per capita rises. However, significant scatter in the data indicates

that other factors play substantial roles in determining compensation. Several vertical clusters suggest that within specific economies (likely representing major tech hubs), there exists considerable salary variation.

Remote Work Compensation Analysis



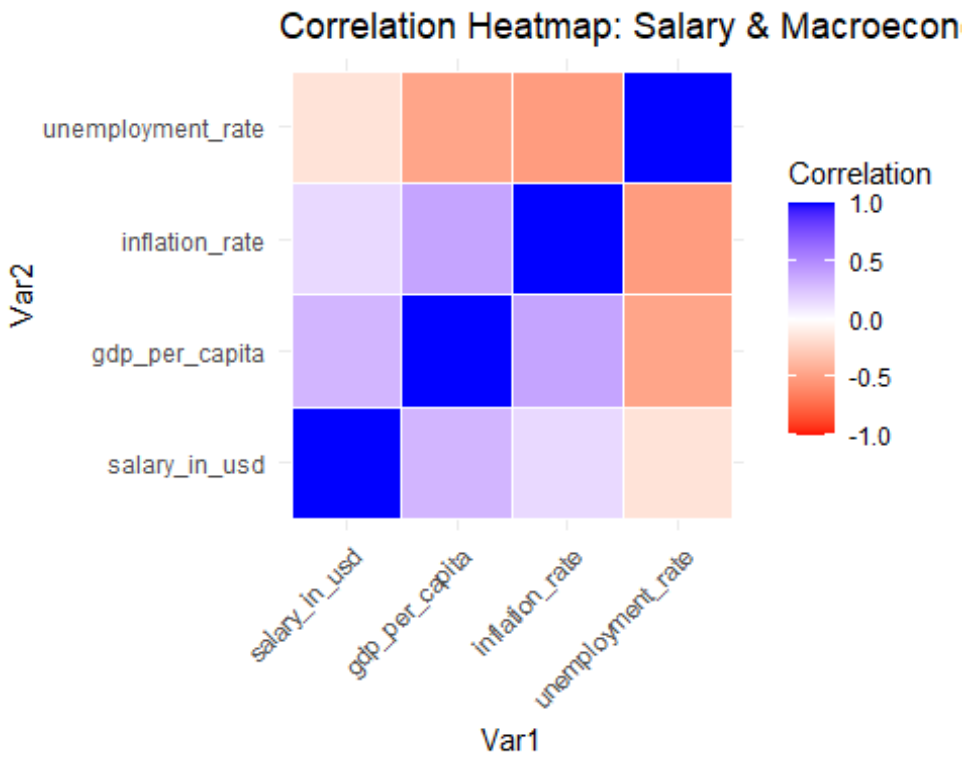
The boxplot comparing salary distributions across work arrangements reveals:

- Fully in-office (0% remote) positions show higher median salaries than hybrid arrangements
- Hybrid (50% remote) positions display the lowest median compensation
- Fully remote (100% remote) positions offer salaries comparable to in-office roles

- All three categories exhibit high-salary outliers, with the most extreme values in the in-office category

This finding challenges conventional assumptions that remote work necessarily leads to lower compensation, suggesting that the data science field may have adapted well to remote work models.

Correlation Between Macroeconomic Factors and Salary



The correlation heatmap demonstrates:

- Moderate positive correlation between GDP per capita and salaries
- Weak positive correlation between inflation rate and salaries
- Weak negative correlation between unemployment rate and salaries

- Strong positive correlation between GDP per capita and inflation rate
- Moderate negative correlation between unemployment rate and GDP per capita

These relationships align with economic theory and provide context for understanding how broader economic conditions influence data science compensation.

3.2 Multicollinearity Assessment

The VIF analysis produced the following results:

- GDP per capita: 1.33
- Inflation rate: 1.41
- Unemployment rate: 1.52

All VIF values are well below the problematic threshold (commonly considered to be 5-10), indicating minimal multicollinearity among the macroeconomic predictors. This confirms that each variable contributes unique information to the model, resulting in stable and reliable coefficient estimates.

3.3 Regression Model Results

The log-transformed linear regression model with interaction terms revealed several significant relationships:

Macroeconomic Factors:

- GDP per capita: Highly significant positive effect ($p < 0.001$), with a $10,000 increase associated with approximately 15.12% higher salaries

- Inflation rate: Significant positive effect ($p < 0.05$), with a 1 percentage point increase associated with about 1.42% higher salaries
- Unemployment rate: Significant negative effect ($p < 0.001$), with a 1 percentage point increase associated with about 1.06% lower salaries

Work Year:

- 2023: Significant negative effect compared to 2022 ($p < 0.001$), with salaries about 7.97% lower

Experience Level (compared to entry-level):

- Executive: 72.4% higher salaries ($p < 0.001$)
- Mid-level: 40.3% higher salaries ($p < 0.001$)
- Senior: 69.6% higher salaries ($p < 0.001$)

Company Size (compared to large companies):

- Medium: 10.2% higher salaries ($p < 0.05$)
- Small: 4.5% higher salaries (not statistically significant)

Employment Type (compared to contract):

- Full-time: 36.0% higher salaries ($p < 0.01$)
- Part-time and Freelance: Not statistically significant

Interaction Effects:

- Several significant negative interactions between experience level and company size

- Senior-level premium reduced by 12.6% in medium companies and by 34.5% in small companies

The model explains approximately 32.74% of the variation in log salaries (R-squared: 0.3274).

3.4 Machine Learning Models Performance

The performance metrics of the three modeling approaches on the test data:

Random Forest:

- RMSE: 56,555 USD
- R-squared: 0.2107
- MAE: 43,909 USD

XGBoost:

- RMSE: 56,699 USD
- R-squared: 0.2066
- MAE: 44,021 USD

Log-Transformed Linear Regression:

- RMSE: 57,914 USD
- R-squared: 0.1955
- MAE: 43,828 USD

All three models performed relatively similarly, with Random Forest showing a slight edge in RMSE and R-squared. The linear regression model had the lowest MAE, suggesting it makes

smaller errors on average despite having a higher RMSE. The modest R-squared values across models indicate that while we've captured significant factors affecting salaries, there remain unmeasured variables or random elements influencing data science compensation.

3.5 Feature Importance Analysis

The XGBoost feature importance analysis revealed:

1. **Experience level**: Most important predictor (importance score ~0.4)
2. **GDP per capita**: Second most important feature (importance score ~0.35)
3. **Unemployment rate**: Moderate importance (importance score ~0.1)
4. **Inflation rate**: Lower importance (importance score ~0.05)
5. **Company size**, **work year**, and **employment type**: Minimal importance (all <0.02)

This analysis confirms that while macroeconomic conditions—particularly a country's economic development level—significantly influence salaries, professional experience remains the dominant factor in determining data science compensation.

4. Discussion

4.1 Key Findings

1. **Experience Dominance**: Experience level emerged as the strongest predictor of salary, with executive and senior positions commanding substantial premiums over entry-level roles. This suggests that career advancement within the field provides significant financial returns.

2. **Economic Development Impact**: GDP per capita strongly influences data science salaries, reflecting how economic development creates differential compensation opportunities globally. This has implications for global talent mobility and remote work policies.

3. **Remote Work Parity**: Contrary to expectations, fully remote positions offer competitive compensation to in-office roles, while hybrid arrangements lag behind. This suggests that companies may be using full remote flexibility as a strategic talent attraction tool, while hybrid models may represent compromises that don't fully optimize for either in-person collaboration or location flexibility.

4. **Interaction Effects**: The significant interaction between company size and experience level reveals that smaller organizations offer reduced premiums for higher experience levels. This suggests that career advancement may yield greater financial benefits when pursued within larger organizations.

5. **Machine Learning Insights**: While machine learning models provided only marginal improvements in predictive accuracy, they revealed important non-linear relationships and confirmed the dominant role of experience and economic development in determining compensation.

4.2 Practical Implications

The findings have several practical implications:

1. **For Professionals**: Career advancement through experience accumulation offers the most reliable path to higher compensation. Geographic location remains important, but remote work options can provide access to higher-paying opportunities regardless of location.

2. **For Employers**: Companies in regions with lower GDP per capita may need to offer remote work options or premium salaries to compete for global talent. Large companies appear to have an advantage in attracting experienced professionals due to their ability to offer higher experience premiums.
3. **For Policy Makers**: Developing data science talent locally can be facilitated by addressing macroeconomic fundamentals, as countries with stronger economic indicators tend to support higher compensation in knowledge-intensive fields.

5. Conclusion

This study provides a comprehensive analysis of the factors influencing data science salaries, demonstrating that both macroeconomic conditions and job-specific characteristics play significant roles. Experience level and economic development emerge as the dominant predictors, while work arrangements show unexpected patterns with remote work offering competitive compensation.

The application of machine learning approaches, while providing only modest improvements in predictive accuracy, offers valuable insights into the complex and non-linear relationships affecting compensation in this field. These findings contribute to the understanding of evolving labor market dynamics in the data science domain and provide practical guidance for professionals, employers, and policy makers.

Future research should incorporate additional variables such as education, specific technical skills, and industry sectors, while also expanding the temporal scope to capture longer-term

trends and economic cycles. Furthermore, developing region-specific models might improve predictive accuracy and provide more targeted insights.

References

Arnab Chaki. 2023. "Data Science Salaries 2023." Kaggle. https://www.kaggle.com/datasets/arnabchaki/data-science-salaries-2023/data

The World Bank. 2023. "GDP per capita (current US$)." World Development Indicators. https://data.worldbank.org/indicator/NY.GDP.PCAP.CD?end=2023&start=2020

The World Bank. 2023. "Inflation, consumer prices (annual %)." World Development Indicators. https://data.worldbank.org/indicator/FP.CPI.TOTL.ZG?end=2023&start=2020

The World Bank. 2023. "Unemployment, total (% of total labor force)." World Development Indicators. https://data.worldbank.org/indicator/SL.UEM.TOTL.NE.ZS?end=2023&start=2020

APPENDIX A (R Code)

```{r message=FALSE, warning=FALSE}

# Load necessary libraries

library(tidyverse)

library(ggplot2)

library(dplyr)

library(caret)

library(randomForest)

library(xgboost)

library(Metrics)

library(corrplot)

library(car)  # For VIF analysis

library(reshape2)  # For heatmap visualization



# Load the merged dataset

ds_salaries <- read.csv("C:/Users/satya/OneDrive/Desktop/Grad School/Machine

Learning/Midterm Project/DS Salaries/final_ds_salaries.csv")

gdp_data <- read.csv("C:/Users/satya/OneDrive/Desktop/Grad School/Machine

Learning/Midterm Project/DS Salaries/GDP_Per_Capita.csv")
```

```
inflation_data <- read.csv("C:/Users/satya/OneDrive/Desktop/Grad School/Machine Learning/Midterm Project/DS Salaries/Inflation_Data.csv")

unemployment_data <- read.csv("C:/Users/satya/OneDrive/Desktop/Grad School/Machine Learning/Midterm Project/DS Salaries/Unemployment_rates.csv")


# ------------------------------#

# 1. Exploratory Data Analysis #

# ------------------------------#


# Scatter Plot: Salary vs. GDP Per Capita

ggplot(ds_salaries, aes(x = gdp_per_capita, y = salary_in_usd)) +

  geom_point(alpha = 0.5, color = "blue") +

  geom_smooth(method = "lm", color = "red", se = FALSE) +

  labs(title = "Salary vs. GDP per Capita", x = "GDP per Capita (USD)", y = "Salary (USD)") +

  scale_x_continuous(labels = scales::comma, limits = c(0, max(ds_salaries$gdp_per_capita, na.rm = TRUE))) +

  scale_y_continuous(labels = scales::comma, limits = c(0, max(ds_salaries$salary_in_usd, na.rm = TRUE))) +

  theme_minimal()
```

```
# Remote vs. In-Office Salary Comparison

ggplot(ds_salaries, aes(x = as.factor(remote_ratio), y = salary_in_usd, fill =
as.factor(remote_ratio))) +
  geom_boxplot() +
  scale_fill_manual(values = c("red", "blue", "green")) +
  labs(title = "Remote vs. In-Office Salary Distribution",
       x = "Remote Work Ratio (0 = In-Office, 100 = Remote)",
       y = "Salary (USD)") +
  scale_y_continuous(labels = scales::comma, limits = c(0, max(ds_salaries$salary_in_usd, na.rm
= TRUE))) +
  theme_minimal()


# Heatmap: Correlation Between Variables

cor_matrix <- cor(ds_salaries %>% select(salary_in_usd, gdp_per_capita, inflation_rate,
unemployment_rate), use = "complete.obs")

melted_cor <- melt(cor_matrix)


# Enhanced Correlation Heatmap
```

```
ggplot(melted_cor, aes(x = Var1, y = Var2, fill = value)) +
  geom_tile(color = "white") +  # Adds gridlines for better visibility
  scale_fill_gradient2(low = "red", high = "blue", mid = "white", midpoint = 0, limits = c(-1, 1))
+
  labs(title = "Correlation Heatmap: Salary & Macroeconomic Factors", fill = "Correlation") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))  # Rotate axis labels for readability




# ------------------------------#
# 2. Multicollinearity Check (VIF) #
# ------------------------------#

model_vif <- lm(salary_in_usd ~ gdp_per_capita + inflation_rate + unemployment_rate, data =
ds_salaries)
vif(model_vif)  # Check for multicollinearity


# ------------------------------#
# 3. Regression Modeling #
```

```
# ------------------------------#

#Convert categorical variables to factors

ds_salaries$work_year <- as.factor(ds_salaries$work_year)

ds_salaries$experience_level <- as.factor(ds_salaries$experience_level)

ds_salaries$employment_type <- as.factor(ds_salaries$employment_type)

ds_salaries$company_size <- as.factor(ds_salaries$company_size)

# ------------------------------#

# Regression Model #

# ------------------------------#

# Include experience level & company size as individual regressors and their interaction term

model <- lm(log(salary_in_usd) ~ gdp_per_capita + inflation_rate + unemployment_rate +

work_year + experience_level * company_size + employment_type, data = ds_salaries)

summary(model)

# ------------------------------#
```

```
# 3. Machine Learning Models #

# ------------------------------#


# Train/Test Split

set.seed(42)

train_index <- createDataPartition(ds_salaries$salary_in_usd, p = 0.8, list = FALSE)

train_data <- ds_salaries[train_index, ]

test_data <- ds_salaries[-train_index, ]


# Ensure categorical variables are converted to numeric dummy variables

train_data_xgb <- train_data %>% mutate(across(where(is.factor), as.numeric))

test_data_xgb <- test_data %>% mutate(across(where(is.factor), as.numeric))


# Convert to XGBoost matrix format

train_matrix <- as.matrix(train_data_xgb %>% select(gdp_per_capita, inflation_rate,

unemployment_rate, work_year, experience_level, company_size, employment_type))

test_matrix <- as.matrix(test_data_xgb %>% select(gdp_per_capita, inflation_rate,

unemployment_rate, work_year, experience_level, company_size, employment_type))
```

```
# Ensure labels are numeric

train_labels <- as.numeric(train_data_xgb$salary_in_usd)

test_labels <- as.numeric(test_data_xgb$salary_in_usd)



# ------------------------------#

# Random Forest Model #

# ------------------------------#



rf_model <- randomForest(salary_in_usd ~ gdp_per_capita + inflation_rate +

unemployment_rate + work_year + experience_level * company_size + employment_type,

                         data = train_data,

                         ntree = 500,

                         mtry = 4,

                         importance = TRUE)

rf_preds <- predict(rf_model, test_data)



# ------------------------------#

# XGBoost Model #

# ------------------------------#
```

```
# Define hyperparameter tuning grid
xgb_grid <- expand.grid(
  nrounds = c(200, 300, 400),       # Increased number of boosting rounds
  max_depth = c(4, 6, 8),           # Adjusted depth to prevent overfitting
  eta = c(0.05, 0.1, 0.2),          # Learning rate for smoother convergence
  gamma = c(0, 0.1, 0.3),           # Regularization
  colsample_bytree = c(0.7, 0.9, 1), # Feature selection per tree
  min_child_weight = c(1, 3, 5),     # Minimum sum of instance weight
  subsample = c(0.7, 0.9, 1)        # Sampling rate per round
)

xgb_tuned <- train(
  x = train_matrix,
  y = train_labels,
  method = "xgbTree",
  trControl = trainControl(method = "cv", number = 5),
  tuneGrid = xgb_grid,
```

```
  verbosity = 0  # Suppress excessive logging
)

# Predict using tuned XGBoost model
final_xgb <- xgboost(
  data = train_matrix,
  label = train_labels,
  nrounds = xgb_tuned$bestTune$nrounds,
  max_depth = xgb_tuned$bestTune$max_depth,
  eta = xgb_tuned$bestTune$eta,
  gamma = xgb_tuned$bestTune$gamma,
  colsample_bytree = xgb_tuned$bestTune$colsample_bytree,
  min_child_weight = xgb_tuned$bestTune$min_child_weight,
  objective = "reg:squarederror"
)

# Feature Importance Analysis
importance_matrix <- xgb.importance(model = final_xgb)
```

```
xgb.plot.importance(importance_matrix, main = "Feature Importance (XGBoost)")



# ------------------------------#

# 4. Model Evaluation on Test Data #

# ------------------------------#



# Compute predictions on test set

rf_test_preds <- predict(rf_model, test_data)

xgb_test_preds <- predict(final_xgb, test_matrix)



# Convert log-salaries back to normal scale (if using log model)

test_data$log_salary_preds <- exp(predict(model, test_data))



# Calculate performance metrics

rf_results <- postResample(rf_test_preds, test_data$salary_in_usd)

xgb_results <- postResample(xgb_test_preds, test_labels)

lm_results <- postResample(test_data$log_salary_preds, test_data$salary_in_usd)
```

```
# Print results

print("Random Forest Performance:")

print(rf_results)



print("XGBoost Performance:")

print(xgb_results)



print("Log-Transformed Linear Regression Performance:")

print(lm_results)
```