Predicting Salary from Demographics and Professional Experience

Andrew, Taran, Ashley, and Krupa

Abstract

Salaries vary greatly across countries due to multiple factors, but the key factors that influence salaries are unclear. In this project, we aim to investigate predictors of salary, including education, experience, industry, gender, and race. Using over 28,000 survey responses from AskAManager.org and data cleaning, we focused on analyzing the US, UK, and Canada for our research question: What are the key factors that influence salaries within each country, and how do these factors differ across countries? We then applied a Random Forest Classifier model to look at salary factors and to evaluate which feature was the most important. Our results showed that industry, education level, and gender are the biggest predictors of salaries. Through our exploratory data analysis, we also saw that years of experience correlates with salary. Our project suggests that education and industry matter most and don't matter with countries when determining salary levels.

Introduction

Understanding key factors for high salaries is important for economic opportunities and not many studies focus on many factors; they often only focus on a single factor like gender or education. Research such as Blau and Kahn's Gender Wage Gap (2017) highlights this, but there is a gap in knowledge when it comes to multiple factors. Through this project, we aim to observe the importance of different factors and how they affect salaries. For example, is education or experience more important for a higher paying job? Salaries differ across countries, and we can explore these changes based on different factors.

We used survey data from AskAManager.org, which had over 28,000 responses globally. We narrowed our focus to the US, UK, and Canada, as they dominated the number of responses in the survey (around 26,000). This dataset gave us a good opportunity to look at different demographics and professions, and also to compare between countries. Additionally, it was

a great dataset to practice real-world data cleaning as it was quite messy. Our approach for data cleaning focused on normalizing currencies, experience binning, and handling missing or extra values. In our exploratory data analysis, we visualized salary distributions across key variables. Ultimately, we chose to use a Random Forest classifier because it handles mixed data well, can capture non-linear patterns and interactions between variables, which can give us feature importance, helping us identify which variables most impact salary across countries (Breiman 2001).

Our research provided us with new insights: being in the tech industry, having a master's degree and being a woman are the most important salary predictors according to our model. Surprisingly, the location doesn't seem to have a big impact on the predictions of salaries. Data analysis revealed different results: the most influential factors were having a PhD, 20-30 years of experience, being male, and being in the tech industry.

Methods

Our project's data comes from an online Google Forms survey that was open worldwide on AskAManager.org's website. The dataset has 17 variables; the quantitative variables were the base salary, bonus income, and years of experience. The categorical variables were country, industry, education level, gender, and race. Six of the columns are free-response entries that were an addition to the main required questions, which we removed during our data cleaning. The dataset contains more than 27,000 individual data points. Around 26,000 data points were focused on the US, UK, and Canada. The project workflow is highlighted in Figure 1.

Data Cleaning

Surveys are a great method of data collection however they often suffer from issues due to free response answers and incomplete answers. These unconstrained answers and null values complicate our ability to use this data to make predictions or identify patterns in the data and thus a level of data cleaning is required before performing any further analysis on this data.

Our first steps were to ensure consistency throughout the data where we performed several preprocessing steps aimed at ensuring standardization of values to prepare for data analysis. This started with consolidating compensation data into a single value where we combined bonuses and base pay all into one new column named total compensation. Additionally, as compensation was reported in each respondents corresponding national currency, we created a new column which was the respondents total compensation normalized into USD using current exchange rates.

Next, we decided to remove extraneous columns and data automatically provided by google surveys that didn't match the context of our analysis or provided hard to classify data. This

culminated in the removal of the timestamp column, job context and job title columns. The timestamp column just provided the time the form was submitted which provided little to no discernable data in regards to the person submitting. The job context and the job title columns did provide useful information, however, the responses in these columns were often incredibly verbose and provided far too much detail in regards to the respondents job description to the point it was far easier to group professions based on the industry column rather than attempting to bucket job titles or job context responses into a feasible number of classifications.

Because our dataset contained so many values, I decided it was best to just drop data points that were missing data in crucial categories such as gender, race, education level and industry. These categories were important for our analysis on demographic and occupational breakdowns. We considered using data augmentation techniques as an alternative to just dropping values however these attributes followed no structured pattern that would have allowed for common augmentation techniques. Given the size of the dataset, removing this sample of data still left with a large and robust sample that would still represent the population, allowing us to preserve data integrity.

In addition to incomplete entries, we also standardized key demographic columns like gender and race. Since the survey allowed for free-form text based responses, many respondents answered "man", "male", "woman" or "female" among many other responses. We decided to group these into 4 different categories just based on attempting to keep each category as balanced as possible, these categories were Man, Woman, Non-Binary and Other. Similarly, for race, we decided to group responses into broader racial groups such as Hispanic, White, Asian, Black or African American, or Other. This allowed us to retain a level of demographic specificity while ensuring that the category had a consistent set of values.

After ensuring that we dropped incomplete entries and ensuring that our key categories had consistent data, the next step was to address sections with poorly written or dirty data. Since the majority of respondents in our dataset came from the United States, United Kingdom and Canada, we decided to focus our research on these three countries for a more accurate and meaningful comparison. To support this comparison, we had to standardize country names as respondents often entered country names in inconsistent formats such as US, USA or America to indicate the United States. It was important to normalize the country data to consistent values to ensure accurate grouping and filtering later on in analysis. We created a mapping of common variations as well as common misspellings of the United States into "USA", United Kingdom into "UK" and Canada into "CANADA".

To complete the data cleaning process, we cleaned up the formatting of experience levels entries in the data. These entries had inconsistent formatting such as "2 - 4 years", "2-4years", and "1 year or less". We standardized the way experience levels into a numerical form where it was represented as two numbers separated by a hyphen such as "2-4" to represent "2-4 years" of experience. This binning of experience values made it far easier for grouping functions later on in analysis as well as having it in numerical format allowed for further feature engineering stages that may leverage these values. With this final transformation done to the data, we

performed a verification of the data, checking null values, confirming data types and doing a final check that our transformations occurred, ensuring that the data was ready for statistical analysis and modeling.

Data Analysis

Exploratory data analysis was used to visualize the distribution and impact of features. This included plots for the overall country data distribution, salary distributions within each country, and factors that influenced salary distributions within and across countries. A standard color encoding was used for the visualizations performed for each country: Green, Purple, and Orange for USA, CANADA, and UK respectively. Tools such as matplotlib, seaborn, and pandas were used for our visualizations.

Country-Level Distributions: Plotted the amount of data available for each country to understand bias, imbalance in data, and to inform our model training process later on. With a majority of data available only in the U.S., Canada, and UK, we proceeded with our analyses on these three countries. Boxplots of annual income were plotted to understand salary distributions in each country.

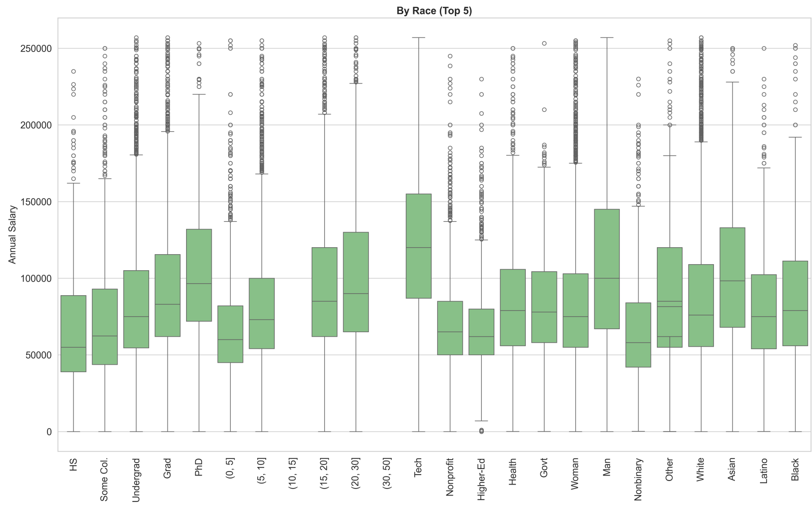
Within-Country Comparisons: We used boxplots to evaluate how salary is influenced by: Education Level, Years of Experience, Industry, Gender, and Race. For U.S. respondents, we further explored state-level salary differences, identifying the top and bottom five states by median salary. The top five and bottom five states based on median salaries were analyzed to examine the differences within the United States.

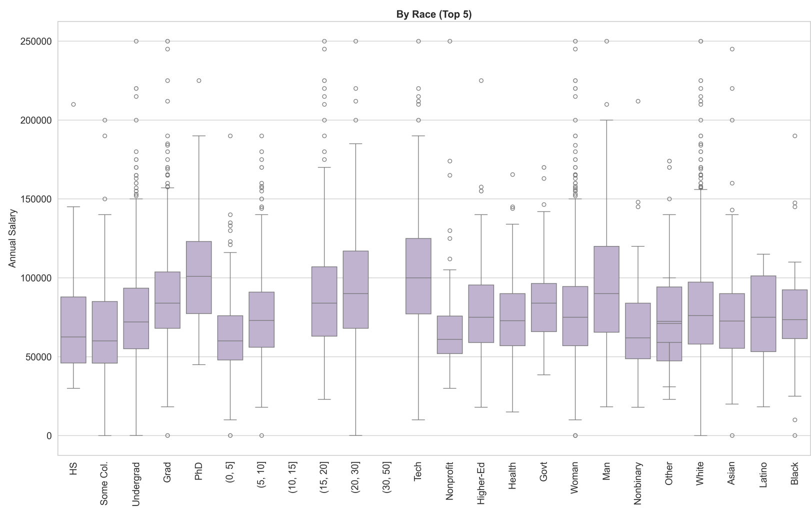
	ageRange	industry	annualSalary	addIncome	currency	\
0	25-34	Education (Higher Education)	55000	0.0	USD	
1	25-34	Computing or Tech	54600	4000.0	GBP	
2	25-34	Accounting, Banking & Finance	34000	0.0	USD	
3	25-34	Nonprofits	62000	3000.0	USD	
4	25-34	Accounting, Banking & Finance	60000	7000.0	USD	

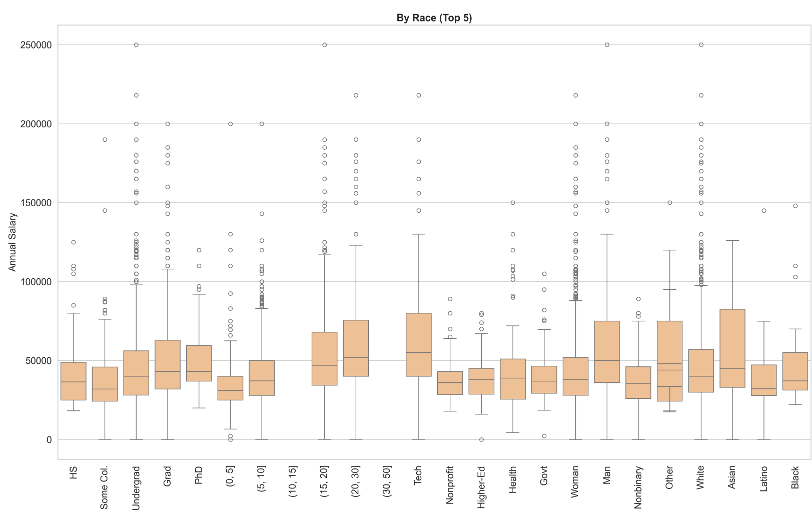
	workCountry	usState	usCity	overallProExp	fieldExp	\
0	USA	Massachusetts	Boston	5-7	5-7	
1	UK	NaN	Cambridge	8-10	5-7	
2	USA	Tennessee	Chattanooga	2-4	2-4	
3	USA	Wisconsin	Milwaukee	8-10	5-7	
4	USA	South Carolina	Greenville	8-10	5-7	

	eduLevel	gender	race	totalCompensation	compUSD	race_grouped
0	Master's degree	Woman	White	55000.0	55000.0	White
1	College degree	Non-binary	White	58600.0	78524.0	White

```
2   College degree       Woman  White           34000.0  34000.0       White
3   College degree       Woman  White           65000.0  65000.0       White
4   College degree       Woman  White           67000.0  67000.0       White
```


By Race (Top 5)
Annual Salary
250000
200000
150000
100000
50000
0
HS
Some Col.
Undergrad
Grad
PhD
(0, 5]
(5, 10]
(10, 15]
(15, 20]
(20, 30]
(30, 50]
Tech
Nonprofit
Higher-Ed
Health
Govt
Woman
Man
Nonbinary
Other
White
Asian
Latino
Black


By Race (Top 5)
Annual Salary
250000
200000
150000
100000
50000
0
HS
Some Col.
Undergrad
Grad
PhD
(0, 5]
(5, 10]
(10, 15]
(15, 20]
(20, 30]
(30, 50]
Tech
Nonprofit
Higher-Ed
Health
Govt
Woman
Man
Nonbinary
Other
White
Asian
Latino
Black

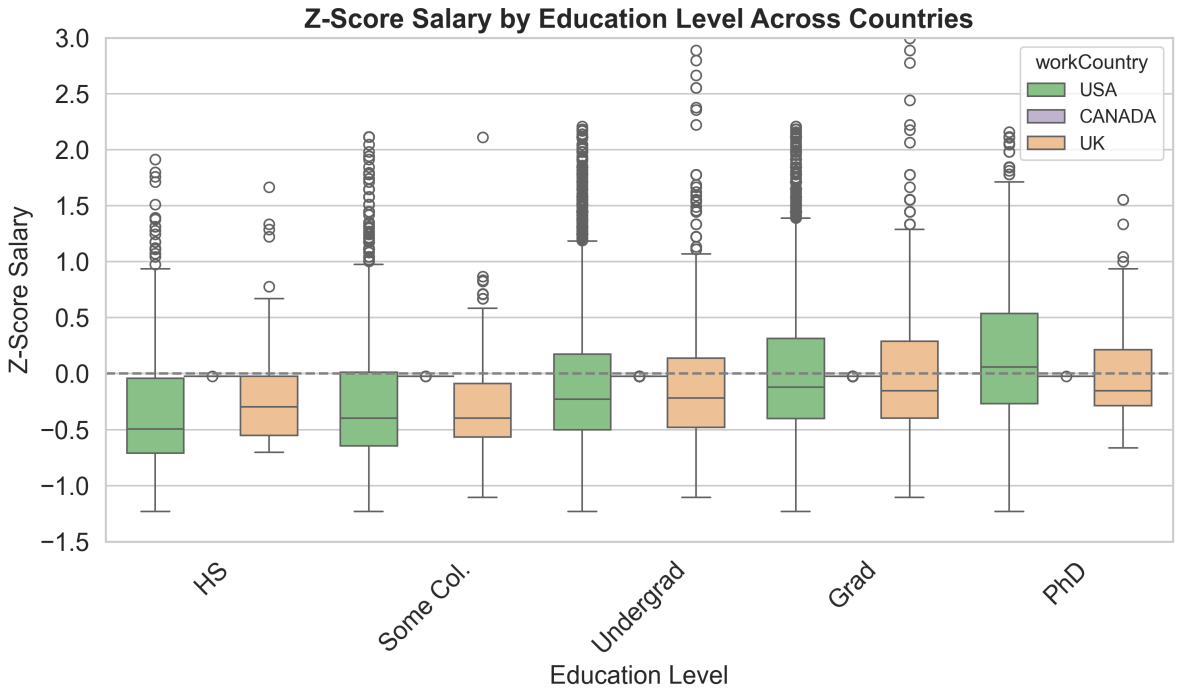

By Race (Top 5)
Annual Salary
250000
200000
150000
100000
50000
0
HS
Some Col.
Undergrad
Grad
PhD
(0, 5]
(5, 10]
(10, 15]
(15, 20]
(20, 30]
(30, 50]
Tech
Nonprofit
Higher-Ed
Health
Govt
Woman
Man
Nonbinary
Other
White
Asian
Latino
Black

Cross-country comparisons: Boxplots were also used for comparison of how different features influence salary distributions across countries. To make salaries comparable across different economic systems, we used z-score salary distributions for each key factor above. Plots were grouped by education, experiences, gender, race, and industry to highlight how each factor affects salary relatively in each country.

```
# set up color palette for consistent coloring
colors = plt.cm.Accent.colors
palette = {'USA': colors[0], 'CANADA': colors[1], 'UK': colors[2]}


# convert experience values into numeric and bin
df_clip['overallExpYears'] = df_clip['overallProExp'].map(exp_map)
df_clip['exp_bin'] = pd.cut(
    df_clip['overallExpYears'],
    bins=[0, 5, 10, 15, 20, 30, 50],
    labels=['0-5', '6-10', '11-15', '16-20', '21-30', '30+']
)


order = ['USA', 'CANADA', 'UK']  # used in hue_order

# plot z-score salary vs education
plt.figure(figsize=(10, 6))
sns.boxplot(data=df_clip, x='eduLevelShort', y='z_salary_country',
            hue='workCountry', hue_order=order, palette=palette, order=edu_order)
plt.axhline(0, linestyle='--', color='gray')
plt.title("Z-Score Salary by Education Level Across Countries", fontsize=TITLE_FONT, fontwei
plt.xlabel("Education Level", fontsize=LABEL_FONT)
plt.ylabel("Z-Score Salary", fontsize=LABEL_FONT)
plt.xticks(rotation=45, fontsize=TICK_FONT)
plt.yticks(fontsize=TICK_FONT)
plt.ylim(-1.5, 3)
plt.tight_layout()
plt.show()

# plot z-score salary vs experience
plt.figure(figsize=(10, 6))
sns.boxplot(data=df_clip, x='exp_bin', y='z_salary_country',
            hue='workCountry', hue_order=order, palette=palette)
plt.axhline(0, linestyle='--', color='gray')
plt.title("Z-Score Salary by Experience Bin Across Countries", fontsize=TITLE_FONT, fontweigh
plt.xlabel("Experience (Years)", fontsize=LABEL_FONT)
```

```
plt.ylabel("Z-Score Salary", fontsize=LABEL_FONT)
plt.xticks(fontsize=TICK_FONT)
plt.yticks(fontsize=TICK_FONT)
plt.ylim(-1.5, 3)
plt.tight_layout()
plt.show()

# plot z-score salary vs industry
df_ind = df_clip[df_clip['industryShort'].isin(top_industries)]
plt.figure(figsize=(10, 6))
sns.boxplot(data=df_ind, x='industryShort', y='z_salary_country',
            hue='workCountry', hue_order=order, palette=palette,
            order=industry_order)
plt.axhline(0, linestyle='--', color='gray')
plt.title("Z-Score Salary by Industry (Top 5) Across Countries", fontsize=TITLE_FONT, fontwe
plt.xlabel("Industry", fontsize=LABEL_FONT)
plt.ylabel("Z-Score Salary", fontsize=LABEL_FONT)
plt.xticks(rotation=45, fontsize=TICK_FONT)
plt.yticks(fontsize=TICK_FONT)
plt.ylim(-1.5, 3)
plt.tight_layout()
plt.show()

# plot z-score salary vs gender
plt.figure(figsize=(10, 6))
sns.boxplot(data=df_clip, x='genderShort', y='z_salary_country',
            hue='workCountry', hue_order=order, palette=palette, order=gender_order)
plt.axhline(0, linestyle='--', color='gray')
plt.title("Z-Score Salary by Gender Across Countries", fontsize=TITLE_FONT, fontweight='bold
plt.xlabel("Gender", fontsize=LABEL_FONT)
plt.ylabel("Z-Score Salary", fontsize=LABEL_FONT)
plt.xticks(fontsize=TICK_FONT)
plt.yticks(fontsize=TICK_FONT)
plt.ylim(-1.5, 3)
plt.tight_layout()
plt.show()

# plot z-score salary vs race
df_race = df_clip[df_clip['raceShort'].isin(top_races)]
plt.figure(figsize=(10, 6))
sns.boxplot(data=df_race, x='raceShort', y='z_salary_country',
            hue='workCountry', hue_order=order, palette=palette, order=race_order)
```

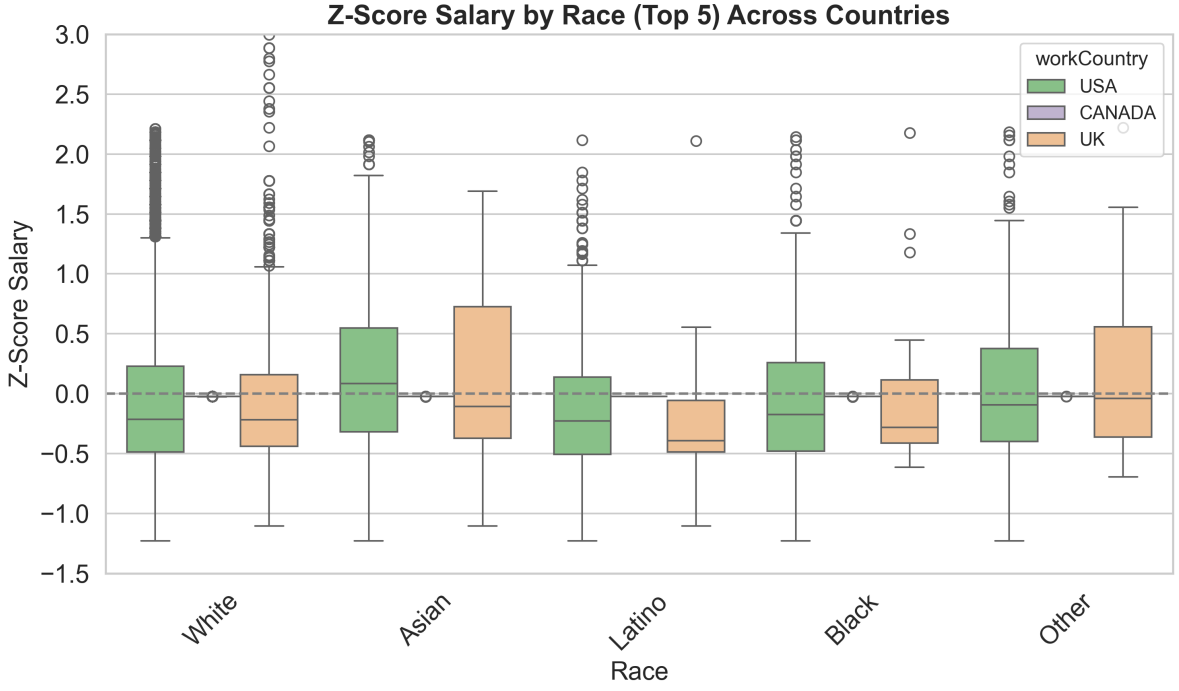
```
plt.axhline(0, linestyle='--', color='gray')
plt.title("Z-Score Salary by Race (Top 5) Across Countries", fontsize=TITLE_FONT, fontweight=
plt.xlabel("Race", fontsize=LABEL_FONT)
plt.ylabel("Z-Score Salary", fontsize=LABEL_FONT)
plt.xticks(rotation=45, fontsize=TICK_FONT)
plt.yticks(fontsize=TICK_FONT)
plt.ylim(-1.5, 3)
plt.tight_layout()
plt.show()
```




Z-Score Salary by Experience Bin Across Countries
workCountry
USA
CANADA
UK
Z-Score Salary
3.0
2.5
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
0–5
6–10
11–15
16–20
21–30
30+
Experience (Years)


Z-Score Salary by Industry (Top 5) Across Countries
workCountry
USA
CANADA
UK
Z-Score Salary
3.0
2.5
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
Tech
Nonprofit
Higher-Ed
Health
Govt
Industry


Z-Score Salary by Gender Across Countries
workCountry
USA
CANADA
UK
Z-Score Salary
3.0
2.5
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
Woman
Man
Nonbinary
Other
Gender


Z-Score Salary by Race (Top 5) Across Countries
workCountry
USA
CANADA
UK
Z-Score Salary
3.0
2.5
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
White
Asian
Latino
Black
Other
Race

Model Description

We employed a Random Forest Classifier to make predictions on the estimated salary brackets and bucket the results. We chose this model because it accommodates heterogeneous features– continuous variables, like total compensation (salary and bonuses combined) and years of experience, working alongside one-hot encoded features, like country, industry, and race– without requiring strict distributional assumptions or extensive feature scaling. Its robustness to outliers, inclination to capture nonlinear interactions and built-in estimation for variable importance align with our project statement and help us produce accurate predictions for salaries and rank the relative influence of socioeconomic and demographic variables.

To quantify the relative influence of different demographic and professional features on compensation, we used a multi-class salary bracket classification. totalComp was bucketed, following the US federal tax bands by applying fixed monetary cutpoints. After categorization, we were left with approximately 25900 records spanning the three major countries. All predictors, except totalComp, were retained. Numerical features were unchanged but categorical features like country, industry, etc were converted to binary indicators using one-hot encoding, with the first level dropped to avoid collinearity.

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_squared_error, r2_score
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import classification_report

df = pd.read_csv('cleandata.csv')

# Define bins and labels with ranges attached
bins = [0, 11000, 44725, 95375, 182100, 231250, float('inf')]
labels = [
    'Low (0-11k)',
    'Lower Middle (11k-44.7k)',
    'Middle (44.7k-95.3k)',
    'Upper Middle (95.3k-182.1k)',
    'High (182.1k-231.2k)',
    'Very High (231.2k+)'
]

# Apply the cut
df['SalaryCategory'] = pd.cut(df['compUSD'], bins=bins, labels=labels)

# View counts per category
```

```
print(df['SalaryCategory'].value_counts())
print(df[['compUSD', 'SalaryCategory']].head())

print(df.dtypes)
print(df["overallProExp"].unique())

# group other response
print(df["gender"].unique())
print("NaNs in SalaryCat: ", df["SalaryCategory"].isna().sum())


# Drop rows where salary category couldn't be assigned (NaN)
df = df.dropna(subset=['SalaryCategory'])

print("NaNs in SalaryCat: ", df["SalaryCategory"].isna().sum())
```

```
SalaryCategory
Middle (44.7k-95.3k)           13316
Upper Middle (95.3k-182.1k)     7090
Lower Middle (11k-44.7k)        3401
High (182.1k-231.2k)            1023
Very High (231.2k+)             1006
Low (0-11k)                       82
Name: count, dtype: int64
   compUSD            SalaryCategory
0  55000.0      Middle (44.7k-95.3k)
1  78524.0      Middle (44.7k-95.3k)
2  34000.0  Lower Middle (11k-44.7k)
3  65000.0      Middle (44.7k-95.3k)
4  67000.0      Middle (44.7k-95.3k)
ageRange               object
industry               object
annualSalary            int64
addIncome             float64
currency               object
workCountry            object
usState                object
usCity                 object
overallProExp          object
fieldExp               object
eduLevel               object
gender                 object
```

```
race                   object
totalCompensation     float64
compUSD               float64
race_grouped           object
SalaryCategory       category
dtype: object
['5-7' '8-10' '2-4' '21-30' '11-20' '0-1' '41+' '31-40']
['Woman' 'Non-binary' 'Man' 'Other']
NaNs in SalaryCat:  14
NaNs in SalaryCat:  0
```

The data was partitioned into a 80/20 training-testing split. We fitted the RandomForestClassifier from the scikit-learn library with n_estimators=16 and random_state= 42(default depth and split criteria). The forest, despite having a relatively small size, was able to converge while keeping inference latency to a minimum. No additional hyperparameter tuning was applied because the grid search did not improve validation accuracy, but that is an avenue that could be explored in the future. Model assessment used accuracy and the macro-averaged precision, recall and F-score as supplied by the classification_report utility.

Results

Key Insights from EDA From our EDA, several clear patterns emerged. The technology industry, 20–30 years of experience, and holding a PhD were consistently associated with higher median salaries across all three countries. However, the strength of these effects varied by country. The impact of working in tech and having a PhD was more pronounced in the US, while in the UK, years of experience (particularly 21–30 years) had a stronger influence on salary. Significant gender disparities were also observed. In all three countries, women and nonbinary individuals had lower median salaries compared to men, with the gap being most prominent in the US (as shown in Figure 2).

Race also showed a notable effect: identifying as Asian was associated with higher median earnings in both the US and UK, with a stronger impact in the US. Therefore, higher salaries were more likely among Asian individuals, especially in the US context. These insights informed our modeling approach by highlighting the most predictive features for salary and emphasizing where country-specific adjustments might be necessary.

Random Forest Results The forest achieved an overall accuracy of 0.88 on the test set and a weighted F-score of 0.86. Performance was heterogeneous across all salary bands: majority data was collected from the middle class, which led to the model classifying that part of the data almost perfectly, but the minority low class was rarely recovered, leading to negligible precision and recall. Despite this problem, the model's macro average F-score is acceptable

for descriptive purposes proving that our chosen feature set contains more than enough signal to categorize the broad compensation distribution.

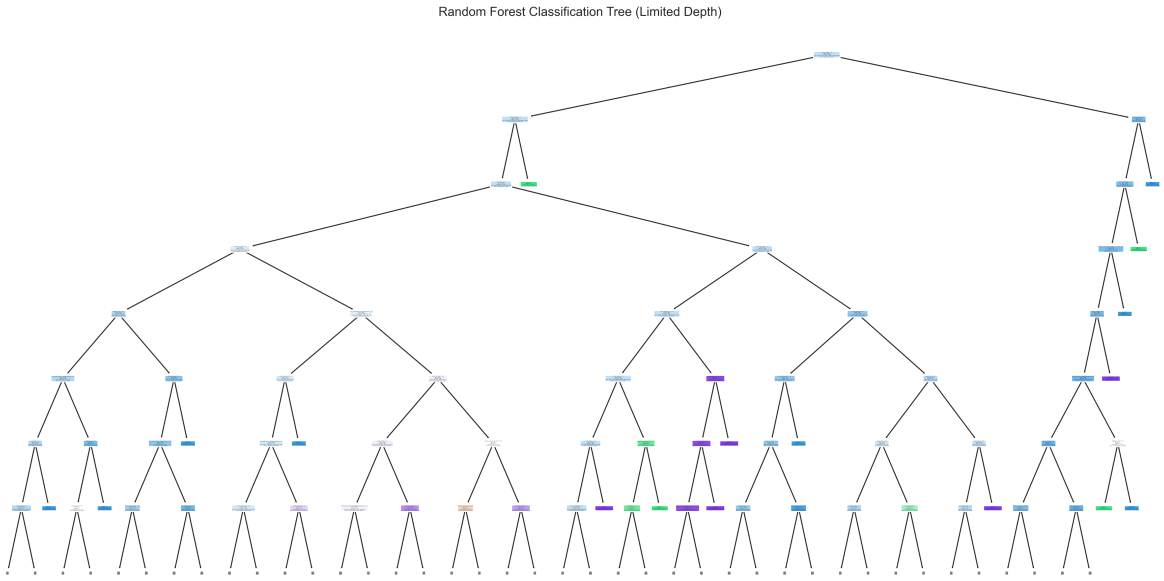
```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_squared_error, r2_score
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import classification_report
import matplotlib.pyplot as plt
from sklearn.tree import plot_tree

df = pd.read_csv('cleandata.csv')

# Define bins and labels with ranges attached
bins = [0, 11000, 44725, 95375, 182100, 231250, float('inf')]
labels = [
    'Low (0-11k)',
    'Lower Middle (11k-44.7k)',
    'Middle (44.7k-95.3k)',
    'Upper Middle (95.3k-182.1k)',
    'High (182.1k-231.2k)',
    'Very High (231.2k+)'
]

# Apply the cut
df['SalaryCategory'] = pd.cut(df['compUSD'], bins=bins, labels=labels)


# Drop rows where salary category couldn't be assigned (NaN)
df = df.dropna(subset=['SalaryCategory'])


# drop all columns associated with salary and compensation
df.drop(columns=['annualSalary', 'addIncome'], inplace=True)

# Features and target
X = df.drop(columns=['compUSD', 'SalaryCategory'])
y = df['SalaryCategory']

# Convert categorical features to numeric
X = pd.get_dummies(X, drop_first=True)
```

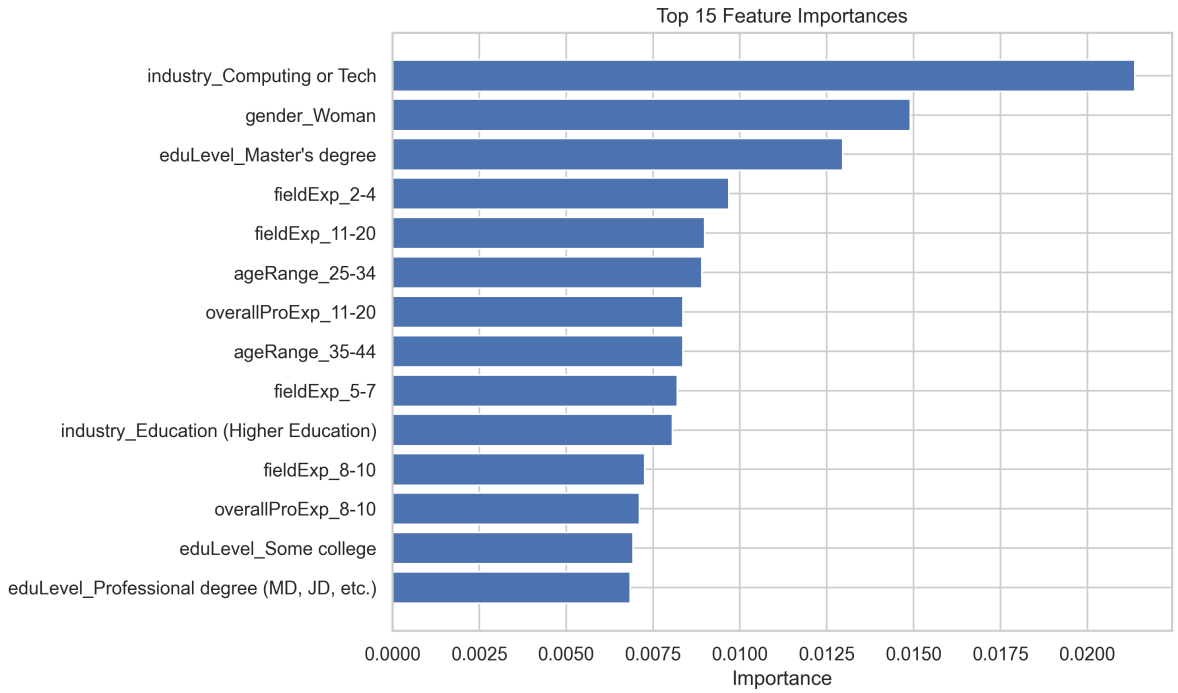
```
# Split into training and test sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42, sti



clf = RandomForestClassifier(
    n_estimators=120,
    random_state=42,
    max_depth=100,
)
clf.fit(X_train, y_train)

# Predict and evaluate
y_pred = clf.predict(X_test)
print(classification_report(y_test, y_pred))



plt.figure(figsize=(20, 10))
plot_tree(clf.estimators_[0], feature_names=X.columns, class_names=clf.classes_, filled=True
plt.title("Random Forest Classification Tree (Limited Depth)")
plt.show()

# Get feature importances
importances = clf.feature_importances_

# Get feature names
feature_names = X_train.columns

# Create a DataFrame for better visualization
feature_importances = pd.DataFrame({'feature': feature_names, 'importance': importances})

# Sort the DataFrame by importance
feature_importances = feature_importances.sort_values(by='importance', ascending=False)

top_n = 15  # Change to however many top features you want to display
top_features = feature_importances.iloc[1:top_n]

plt.figure(figsize=(10, 6))
plt.barh(top_features['feature'][::-1], top_features['importance'][::-1])  # Reverse for high
plt.xlabel('Importance')
plt.title(f'Top {top_n} Feature Importances')
plt.tight_layout()
```

```
plt.show()
```

```
/Users/andrewxue/Desktop/csds133SalarySurvey/.venv/lib/python3.12/site-packages/sklearn/metri
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
/Users/andrewxue/Desktop/csds133SalarySurvey/.venv/lib/python3.12/site-packages/sklearn/metri
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
/Users/andrewxue/Desktop/csds133SalarySurvey/.venv/lib/python3.12/site-packages/sklearn/metri
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
```

	precision	recall	f1-score	support
High (182.1k-231.2k)	0.75	0.22	0.34	211
Low (0-11k)	0.00	0.00	0.00	12
Lower Middle (11k-44.7k)	0.99	0.69	0.82	684
Middle (44.7k-95.3k)	0.90	1.00	0.95	2638
Upper Middle (95.3k-182.1k)	0.83	0.95	0.89	1429
Very High (231.2k+)	0.84	0.34	0.49	210
accuracy			0.88	5184
macro avg	0.72	0.53	0.58	5184
weighted avg	0.88	0.88	0.87	5184





The presentation finding that industry membership (particularly in the technology sector), postgraduate education, and self-identified gender exert the largest marginal effects on the predicted bracket, while the country indicator contributes relatively little once all salaries are expressed in U.S. dollars, was replicated by qualitative inspection of impurity-based feature importances. These patterns are mostly consistent with the patterns observed in the exploratory data analysis section.

In summary, the random forest classifier model was able to capture the dominant trends in the cleaned dataset and its feature importance scores clearly demonstrates the important variables, preserving interpretability and giving us reliable predictions.

Discussion

Our model results and data analysis reinforce the idea that industry and education level are the most important influences on higher salaries. Gender differences in salary also were significant, which exhibits the everlasting gender wage gap. Interestingly, the impact of location was minimal in our model once industry and education were accounted for. This suggests that global shifts in remote work and skill-based pay are changing salary structures!

However, there are certain limitations to our research. Because the data survey was self-reported, there may be bias in salary information or demographic accuracy. Additionally, the overwhelming number of responses from the US limits how much we can generalize our findings

to other parts of the world. Nevertheless, the consistency of our evaluation across the three countries we analyzed shows that it may be okay to generalize these results because they are similar across the US, UK, and Canada.

Nevertheless, the consistency of model rankings across the three countries we analysed suggests our central message is robust. Industry and education dominate, gender differentials persist, and geography matters lesser and lesser once pay is converted to a common currency. Future work could (i) supplement self-reports with verified compensation databases, (ii) oversample regions outside North America and the UK, (iii) incorporate cost-of-living or purchasing-power adjustments, and (iv) apply explainable-AI tools such as SHAP values to probe whether feature effects vary within remote-only subsamples. Such extensions would sharpen our understanding of how the evolving mix of remote work, skills-based hiring, and demographic factors is reshaping the global salary landscape.

Conclusion

Our project highlights that industry, education, and gender are the major factors influencing salary. Contrary to our expectations and our initial breakdown of our work, the country of employment had a small effect. Increasing experience does boost salary, but its effect is not as large after the threshold of 20 years is hit. For the future, we can expand our analysis to more countries and industries to improve global representation; additionally, we can also adjust for cost-of-living and job markets for our analysis and model.

Roles

Andrew was responsible for data cleaning and performed all of the data cleaning tasks. Krupa performed some feature engineering and split up the exploratory data analysis with Ashley and looked into the salary distributions in detail. Taran and Andrew worked on the model together. All members of the team worked together for the final presentation and final report.

References