Medium Search Write Z

All your favorite parts of Medium are now in one sidebar for easy access.

Okay, got it

Profile

Stories

Stats

Following

Find writers and publications to follow.

See suggestions

Predicting Developer Salaries with Machine Learning

Pratiti Soumya Follow 7 min read · Jun 12, 2025

A Step-by-Step Journey Through the Stack Overflow 2024 Developer Survey

As a data science learner, I wanted to dive deep into a *real-world problem. So, I chose to analyze the 2024 Stack Overflow Developer Survey and see if I could predict how much developers earn based on their background, education, and job type.*

In this post, I'll walk you through everything I did : from cleaning messy survey data to building models and interpreting results using SHAP. Along the way, I learned just how complex salary prediction really is.

Project Goals

My main goal was to build a predictive model that estimates annual compensation and explain how key variables affect predictions.

> *- Build a predictive regression model*
>
> *- Tune and evaluate performance*
>
> *- Interpret the results using SHAP to explain which features influenced salary most.*

The goal was to follow a complete data science process: cleaning and preprocessing real-world data, performing exploratory data analysis (EDA), building regression models, tuning them, and interpreting their outputs using SHAP.

Dataset Overview

The dataset had **65,000+ survey responses** across 114 columns.

But many of the columns were:

- free text (hard to use in modeling)
- Multiple-choice questions with messy formatting
- Mostly empty or not useful for predicting salary

So instead of using all 114 columns, I selected a subset of structured, relevant variables that were:

- *Easy to work with*
- *Likely to affect salary*
- *Mostly filled in by respondents*

I focused on the following columns:

- `Country` – where the developer lives
- `EdLevel` – their highest education level
- `YearsCode` – how many years they've been coding
- `Employment` – whether they're full-time, student, etc.

- `DevType` – what kind of developer they are

My target variable was:

- `ConvertedCompYearly` : self-reported salary converted to USD.

After filtering for non-null salary values and cleaning text-based entries, the working dataset was reduced to **~23,000** valid observations.

Exploratory Data Analysis

Before modeling, I want to explore the following questions using visual analysis :

1. What were the most important factors that influenced a developer's salary?
2. How does salary vary across different education levels, countries, and job types?
3. Can I build a machine learning model to predict a developer's salary based on their background and work profile?

The next step was to perform some visual EDA (exploratory data analysis) to look for trends and patterns in the data.

Salary Distribution

The distribution was highly right-skewed, most salaries fell below $200,000, with a small number of very high earners pulling the tail to the right. This skewness is typical in real-world income data, where a few individuals earn significantly more than the average.

This plot confirmed the need for using models that can handle non-linear relationships or skewed distributions.

Education Level and Salary

As expected, those with advanced degrees such as Master's or Doctoral degrees generally reported higher average salaries. Interestingly, some groups like professional degree holders and even those with "some college" experience showed higher earnings. This may reflect other factors like job type, skills, or years of experience.

Overall, this confirmed that `EdLevel` is an important feature for our salary prediction model.

Years of Experience vs. Salary

I expected to see a strong link between experience and salary, but the plot showed a lot of variation. Some developers with just a few years of experience earned very high salaries, while others with decades of experience earned much less.

This tells me that experience alone doesn't explain salary well. That's why I chose to use models like decision trees and random forests. These models can handle complex patterns.

Modeling Approaches

Linear Regression

To start, I used a simple linear regression model. This gave me a baseline to compare other models against. Linear regression assumes a straight-line relationship between features and the target.

After training and testing, I planned to compare the model's error and R^2 score to see how well it performs.

```
# Step 1: Split Target and Features Matrix
y= df_encoded['ConvertedCompYearly'] #set target va
X= df_encoded.drop('ConvertedCompYearly', axis=1)

# Step 2: Train Test Split
X_train, X_test, y_train, y_test= train_test_split(

# Step 3: Train Linear Regression Model
lr_model= LinearRegression()
lr_model.fit(X_train,y_train)

# Step 4: Predict Linear Regression Model
train_lr_pred = lr_model.predict(X_train)
train_lr_r2= lr_model.score(X_train, y_train)
print(f'Linear Model training r^2: {train_lr_r2:.3f
train_lr_rmse= root_mean_squared_error(y_train, tra
print(f'Linear Model training RMSE: {train_rmse:.2f
test_lr_pred= lr_model.predict(X_test)
test_lr_r2= lr_model.score(X_test, y_test)
print(f'Linear Model test r^2:{test_lr_r2:.3f}')
```

```
test_lr_rmse= root_mean_squared_error(y_test, test_
print(f'Linear Model test RMSE: {test_lr_rmse:.2f}'
```

I trained a linear regression model to predict salary using the features I selected. The model's performance was quite weak:

- R^2 Score (Train): 0.107
- R^2 Score (Test): 0.127
- RMSE (Train): ~$189,000
- RMSE (Test): ~$108,000

This means the model was only explaining about 10–13% of the variation in salaries — and the prediction error(RMSE) was very high.

Linear regression failed to capture complexity or non-linear interactions.

Next Step: Trying Tree-Based Models

To improve performance, I tried tree based models to capture non-linear relationships and noisy data like Salary.

Decision Tree Regressor

- **Train R^2**: 0.955
- **Test R^2**: -1.093

- **RMSE (Test)** : $167,672

The test results were poor:

- The R^2 score was **negative**, meaning the model did worse than simply predicting the average salary for everyone.
- The **test RMSE was over $167,000**, indicating very large errors on new data.

This confirmed that the model was **overfitting** ; it performs well on the training set but failed to generalize to unseen data.

Random Forest Regressor (Untuned)

- **Train R^2**: 0.810
- **Test R^2**: -0.236
- **RMSE (Test)** : $128,836

Random Forest reduced overfitting but still generalized poorly on the test set.

Feature Selection and Model Tuning

Using `feature_importances_` from the Random Forest model, I selected the top 10 most impactful features.

I retrained a Random Forest model using only these top 10 features and ran **GridSearchCV** to tune hyperparameters.

Final Model (Tuned Random Forest on Top 10 Features)

```
from sklearn.ensemble import RandomForestRegressor
from sklearn.model_selection import GridSearchCV

# Define hyperparameter grid
param_grid = {
    'max_depth': [15,20, 25],
    'min_samples_split': [2, 5, 10],
    'min_samples_leaf': [1, 2, 4]
}
#Set Up the grid search
grid_search_model=GridSearchCV(estimator=RandomFores
                         param_grid=param_grid,
                         scoring='neg_root_mean_squa
grid_search_model.fit(X_train_top, y_train) # Fit on

#View Best Parameters
print("Best Parameters:", grid_search_model.best_par
best_model= grid_search_model.best_estimator_

# Predict Best Model
train_best_pred= best_model.predict(X_train_top)
train_best_r2= best_model.score(X_train_top, y_train
train_best_rmse= root_mean_squared_error(y_train, tr

test_best_pred= best_model.predict(X_test_top)
```

```
    test_best_r2= best_model.score(X_test_top, y_test)
    test_best_rmse= root_mean_squared_error(y_test, test

    print(f"Train R^2: {train_best_r2:.3f}")
    print(f"Train RMSE: {train_best_rmse:.3f}")
    print(f"Test R^2: {test_best_r2:.3f}")
    print(f"Test RMSE: {test_best_rmse:.3f}")
```

- **Train R²:** 0.188
- **Test R²:** 0.045
- **Test RMSE:** ~113,000 USD

Despite tuning, the model showed limited ability to generalize. The predicted salaries remained far from actual values across most test samples.

Why Performance Was Low

Even after tuning, my best model explained less than 20% of the salary variation. Why?

- **Salary data is noisy and self-reported**
- The dataset lacks details like **company size**, **bonus structure**, and **job titles**
- Some factors like negotiation skill or market conditions aren't captured in surveys

So I shifted focus from accuracy to **interpretability**.

SHAP for Interpretability

I used SHAP to interpret the trained model and understand which features had the greatest impact on salary predictions:

```
import shap

# Explain model predictions
explainer = shap.Explainer(best_model, X_train_top)
shap_values = explainer(X_train_top) #compute SHAP

# Visualize global feature importance
shap.plots.bar(shap_values) # plot overall importanc
```

Key observations:

- Developers in the United States had the strongest positive influence on salary predictions.
- More years of experience correlated with higher predictions.

- Executive and managerial developer types increased predicted compensation.

SHAP confirmed that the model prioritized location, experience, and role type.

Conclusions

1. Does education level significantly affect a developer's salary?

Insight: Yes , developers with higher degrees (Master's, PhD, Professional) generally earn more. However, there are notable exceptions where developers with lower formal education (e.g. self-taught) still report high salaries.

2. Is there a strong correlation between years of experience and salary?

Insight: A weak positive trend exists, but it's not linear. Some developers with very little experience report high salaries, and vice versa. This suggests that experience alone is not a strong predictor.

3. Which countries are associated with higher average salaries?

Insight: The United States consistently ranks at the top, followed by certain countries in Europe. Country was the most influential feature in the SHAP analysis.

4. What developer roles (DevType) are linked to

higher compensation?

Insight: Executive-level roles (e.g. C-Suite, VP, Director) and specialized roles (e.g. Project Manager) had higher predicted salaries compared to general developer titles.

5. Which features have the highest predictive power for salary?

Insight: Based on SHAP values and feature importance from Random Forest, the top predictors were:

- `Country`
- `YearsCode`
- `DevType`
- `EdLevel`

Final Thoughts

This project was a full walkthrough of the **end-to-end data science process**:

- *Data cleaning and preprocessing*
- *Exploratory visualizations*
- *Model building and tuning*
- *Model evaluation*
- *Post-hoc interpretibility using SHAP*

Salary prediction in this context proved highly challenging due to:

- High variance in self-reported salary data
- Lack of features like company size, job title, industry, or bonus structures
- Non-normalized comparisons across countries with different economic standards. I chose not to normalize salary values across countries or apply cost-of-living adjustments because the dataset lacked reliable reference baselines (e.g. average country salaries, regional indices).

Explore the Full Project on GitHub
Source code, notebook, and full SHAP visuals are available here:
🔗 **View Repository**

Data Science Data Scientist Predictive Analytics

Written by Pratiti Soumya

Follow

64 followers · 353 following

Data Scientist

No responses yet

Z Zeynep yilmaz

What are your thoughts?

More from Pratiti Soumya

Pratiti Soumya

The Story Of Amazon.com - Jeff...

It was July of 2005. At school, there was to be a class test...

Jul 24, 2017 55

Pratiti Soumya

The Story Of NIKE — Sports, Enthusiasm,...

This beaten road leads to the city stadium. It often remind...

Aug 22, 2017 64

Pratiti Soumya

How To Overcome Price Sensitivity of...

In today's world of intense competition, where several...

Mar 28, 2017 7

Pratiti Soumya

A/B TESTING FOR A MARKETING...

A Full Funnel Analysis of a Simulated Email Campaign...

Jun 26

See all from Pratiti Soumya

Recommended from Medium

In Level Up Co... by Fareed K...

Building a Self-Improving Agentic...

Specialist agents, multi-dimensional eval, Pareto...

✦ 6d ago 1.1K 5

In Interview Pre... by Pragya ...

Target Sr Data Scientist—Advanced...

Recently, a friend of mine interviewed at Target for a...

✦ Jul 22 39

Rohan Dutt

10 Forecasting Models Used for Revenue,...

From ARIMA to LSTM, see how industries predict the...

Nov 9 16

In Data Science... by Andres ...

What's Trending in Data Science and ML...

Key Learnings from the "Nordic Data Science and...

Nov 3 283 2

Sowmiya V

How to Pick Portfolio Projects That Impres...

Photo by Anastassia Anufrieva on Unsplash

Nov 14 19 2

Maximilian Oliver

End-to-End Data Science in Productio...

From raw data to real-time predictions—a complete...

Aug 4 13

See more recommendations