# **What Developers Want: Insights from the 2024 Stack Overflow Survey**

## **Motivation**

Every year, thousands of developers around the world take the Stack Overflow Survey. It's a treasure chest of data — opinions, salaries, habits, tools, you name it.

I wanted to dig into the 2024 results and answer a few big questions:

* What matters most when it comes to developer salaries?
* Can we spot any surprising trends hiding in the data?
* And if we train a machine learning model... how well can it actually predict what a developer might earn?

This project was built as part of Udacity’s Data Science Nanodegree. I followed the CRISP-DM process, because hey — why reinvent the wheel when a good framework already exists?

## **Libraries Used**

Nothing crazy here — just the essentials:

* pandas
* numpy
* matplotlib
* seaborn
* scikit-learn

All standard, all powerful. (And honestly, all you really need for a project like this.)

## **Files in this Repository**

* MatthewJayApril24.ipynb:  
   My Jupyter Notebook. It's where all the data wrangling, exploring, modeling, and final predictions happen. Think of it as the heart of the project.
* README.md:  
   You’re reading it! It's the roadmap to what this project is about, what's inside, and how it all fits together.

## **CRISP-DM Process (How I Tackled It)**

**Business Understanding** First, the big idea: figure out what factors drive developer salaries and see if we can predict them.

**Data Understanding** I started with simple questions. Where are people from? What do they know? How much experience do they have? Some histograms and correlation heatmaps helped a lot here.

**Data Preparation** There were missing values (because there always are). I cleaned things up, encoded categories, and got everything ready for machine learning.

**Modeling** I built a Random Forest Regressor and tuned it up using GridSearchCV. No fancy deep learning stuff — just something solid that could get the job done.

**Evaluation** The model pulled an R² score of about **0.049**. Not perfect, but it definitely found real patterns between skills, experience, and paychecks.

**Deployment (Scenario + Prediction)** Finally, I created a fictional developer profile and ran it through the model. It spit out a predicted salary, and honestly... the number made sense.

## **Summary of Results**

**So, what did we find?**

* Years of experience matters. A lot.
* Education level helps... but not as much as you might think.
* Country of employment? Huge factor. No surprise there.

**Creative Insight:** Remote developers — full-time remote, not hybrid — tend to pull higher salaries on average. Maybe it's the competition for top talent? Maybe it's the perks of flexibility? Either way, it's there in the numbers.

**Model Performance:** The Random Forest model wasn’t a mind reader, but it made respectable predictions. The R² score shows it picked up on real-world salary drivers, even if it couldn’t predict them perfectly every time.

**Prediction Scenario:** When I created a profile for an experienced remote developer with strong skills... the model predicted a top-tier salary. And honestly? It checks out.

## **Acknowledgments**

* Big thanks to Stack Overflow for making their survey data open and accessible.
* Thanks to Udacity for teaching me a structured way to tackle messy, real-world data problems.
* And a quick shoutout to all the random blog posts, StackOverflow answers, and tutorials that filled in the gaps along the way. You know who you are.

# **🔥 Final Thoughts**

This project wasn’t about building the most perfect model in the world. It was about understanding — really *seeing* — the hidden stories in the data.

And if I’m being honest? I had a lot of fun doing it.