The Data Didn't Care About Income: A Story About Commutes and What We Think We Know

Francesco G. Gillio
Department of Control and Computer Engineering
Politecnico di Torino
Turin, Italy

Abstract

We set out to answer a simple question: can we predict how long people spend commuting, just by looking at census data? To do that, we ran a series of regression analyses on a big dataset—more than 130,000 entries—cleaned it up, stripped out the noise, and fed it into different predictive models. We tried Random Forests, Decision Trees, and Support Vector Regression, to see which one could make the best guesses and tell us which factors really matter. Turns out, Random Forest won hands down, giving us the best results by far.

The real game-changer wasn't income or education—those barely moved the needle. What mattered most were the basics: what time people leave, what time they arrive, and the time between. That last one, which we called Commute Time to Work (CTTW), turned out to be a key feature. Once we added it, Random Forest nailed it, pushing R^2 close to 0.997.

In short, if you want to know how long a commute will take, don't ask about a person's degree or paycheck—just find out when they leave and when they get there. The rest, we found, is mostly noise.

ACM Reference Format:
Francesco G. Gillio. 2025. The Data Didn't Care About Income: A Story About Commutes and What We Think We Know. In . ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 Introduction

Let's say you want to guess how long it takes someone to get to work. You don't know who they are, just some basic facts from census data. Can you make a good guess? That's what this study is about. We built a model that takes those facts and spits out a number—commute time in minutes—somewhere between 0 and 200. And the goal is simple: make that number as accurate as possible.

But we don't want to throw in every variable we've got. That just adds noise and confusion. So we trimmed the fat. We focused on the variables that actually matter and tossed the ones that don't. The result? A cleaner model that doesn't get distracted by irrelevant data.

Conference'17, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YYYY/MM
https://doi.org/10.1145/nnnnnnn.nnnnnnn

The dataset is big and solid: over 130,000 rows, split into two parts. One set—104,642 entries—has everything, including the commute times. The other—26,159 entries—has the same features, but no commute time. That second set is for testing whether our model can predict the missing piece.

Right off the bat, we found something weird: 711 entries in the development set were exact copies of each other. That's not a glitch in the matrix—it's just sloppy data collection. No big deal, they made up less than 1% of the data, so we threw them out. Besides that, the dataset was clean. No missing values, no corrupted entries. Just solid, structured information. After the cleanup, we had a dependable foundation to build on, and a clear path to figure out what really drives commute time.

2 Methodology

2.1 Data Preprocessing

Before you can do anything useful with data, you have to clean it. Garbage in, garbage out. So the first thing we did was throw out what we didn't need—duplicates, noisy features, and anything that would confuse the model more than help it.

We started with 104,642 entries, and after removing exact duplicates—711 of them, to be precise—we ended up with 103,931 clean, unique rows. Each row had 25 features: 22 of them were categorical, 3 were numeric. But here's the nice part—every value, even the categorical ones, was already represented by numbers. That made life easier later when feeding data into the models.

The three numeric features were income, weekly working hours, and commute time (which is the target we're trying to predict). The categorical features included all sorts of socioeconomic information—health insurance, education, housing status, and more. But here's where things got messy: some of these features had heavily skewed distributions. Take health insurance, for example. As Figure 1 shows, nearly every entry says the person doesn't have it. So while it might sound important, it's pretty much useless for our model. Why? Because the model sees the same thing over and over again—it can't learn anything from that.

We didn't just guess which features to keep and which to ditch. We tested them. We ran a bunch of regression models—Random Forest, Decision Tree, Linear Regression, Support Vector Regression, and K-Nearest Neighbors—and looked at how each one performed using all the data. For the ones that need it, we normalized the features so everything fell in the same range. Then we split the data: 75% for training, 25% for testing—same ratio as between our development and evaluation sets.

The results? Random Forest crushed the others. As shown in Table 1, it gave us a mean R^2 of 0.98—far better than anything else. SVR and Linear Regression didn't even come close.

Figure 1: Comparison of those with health insurance coverage (HICOV) versus those without (in number of people).

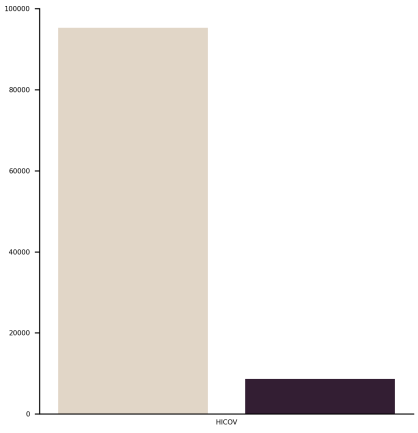
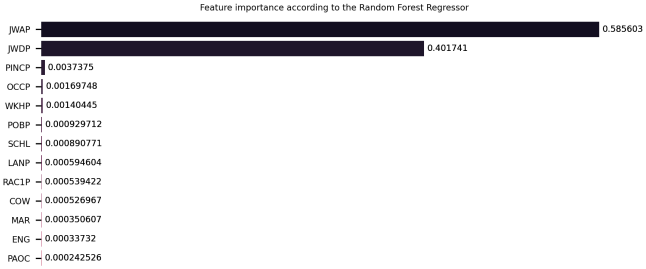


Table 1: Comparison of regression model performance by R^2 Mean and R^2 Standard Deviation.

Model	R^2 Mean	R^2 Standard Deviation
Random Forest	**0.98274**	0.00543
Decision Tree	0.96979	0.00864
SVR	0.22344	0.00786
Linear Regression	0.20265	0.03214
K-Neighbors	0.06584	0.00512

Then came the key question: which features actually help predict commute time? The answer is refreshingly simple. As Figure 2 shows, only two features really matter: the time someone leaves for work and the time they arrive. Together, they account for more than 98% of the model's predictive power. Everything else—income, job type, housing, education—barely registers. Their importance scores were below 0.5%, which is statistical noise.

Figure 2: What the Random Forest Regressor model says about feature importance (subset of attributes with importance value greater than 0.025%).



So we dropped the dead weight. From here on out, our model works with just two features:

- Time of arrival at work (JWAP), values from 1 to 285.
- Time of departure for work (JWDP), values from 1 to 150.

Each value represents a time slot. For example, a JWDP value of 1 means the person left for work between 12:00 and 12:29 a.m. Value 2? That's 12:30 to 12:59 a.m., and so on. These time slots aren't always evenly spaced—some cover 30 minutes, some only 5—but they're distinct and mutually exclusive.

This isn't just a guess backed by intuition—it's a result backed by data. The takeaway is clear: forget the socioeconomic noise. If you want to predict commute time, just look at when someone leaves and when they get there. Everything else is decoration.

2.2 Feature Transformation

The exploratory analysis conducted in the preprocessing phase naturally suggests the adoption of a feature transformation strategy. Each observation in the development set is characterized by two temporal attributes: the time of departure for work (JWDP) and the time of arrival at work (JWAP). These categorical variables, although numerical in appearance, encode discrete time slots rather than continuous timestamps. Their combination, however, implicitly encodes a third and much more informative measure: the total commute time to work.

To extract this latent variable, the transformation process proceeds by first mapping each time slot to a corresponding range of minutes past midnight. The midpoints of these ranges are then computed, enabling an approximate yet consistent conversion from categorical codes to temporal quantities expressed in minutes. Table 2 illustrates this conversion for a selection of representative values.

Table 2: The conversion process: time slots into minutes.

(JWDP) Time Slot	Time Slot in Minutes	Mean Time
12:00 a.m. - 12:29 a.m.	0 min - 29 min	14.5 min
12:30 a.m. - 12:59 a.m.	30 min - 59 min	44.5 min
1:00 a.m. - 1:29 a.m.	60 min - 89 min	74.5 min
1:30 a.m. - 1:59 a.m.	90 min - 119 min	104.5 min
(JWAP) Time Slot	**Time Slot in Minutes**	**Mean Time**
12:00 a.m. - 12:04 a.m.	0 min - 4 min	2 min
12:05 a.m. - 12:09 a.m.	5 min - 9 min	7 min
12:10 a.m. - 12:14 a.m.	10 min - 14 min	12 min
12:15 a.m. - 12:19 a.m.	15 min - 19 min	17 min

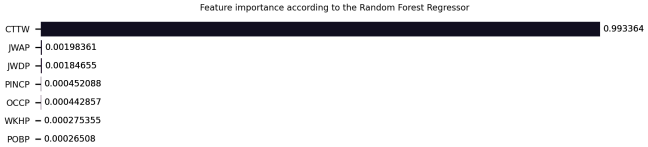
This mapping enables the construction of a new, continuous feature: the Commute Time To Work (CTTW). For each observation, the CTTW is computed as the difference between the converted JWAP and JWDP values, both expressed in minutes. Table 3 provides a concrete example of this transformation for a randomly selected individual from the dataset.

Table 3: The feature transformation process.

JWDP	into min	JWAP	into min	CTTW	JWMNP
49	452 min	94	482 min	**30**	30

Having generated the CTTW attribute, the analysis re-applies the Random Forest Regressor to evaluate its relative importance in predicting the target variable. Figure 3 displays the updated feature importance scores. The result is unambiguous: the newly introduced CTTW attribute exhibits a dominant importance score of 0.933, far surpassing that of the original JWDP and JWAP attributes. This empirical dominance confirms the theoretical expectation that commute duration, rather than raw departure or arrival time, holds primary predictive value.

Figure 3: What the Random Forest Regressor model says about feature importance when playing the additional card on the commute time to work (CTTW) parameter (subset of attributes with importance value greater than 0.025%).



To validate this transformation pipeline, the analysis re-evaluates model performance across the Random Forest, SVR, and K-Neighbors regressors. The same 75/25 train-test split methodology is employed for consistency. As reported in Table 4, the results exhibit substantial improvement across all models. SVR and K-Neighbors, previously underperforming due to the raw, sparse nature of categorical inputs, now show competitive predictive capabilities. The transformation step thus not only enhances interpretability, but also elevates model efficacy across the board.

Table 4: The coefficients of determination for the regression models, before and after preprocessing.

Model	R^2 Mean	R^2 Standard Deviation
RF - Before	0.98274	0.00543
RF - After	**0.99691**	0.00023
SVR - Before	0.22344	0.00786
SVR - After	**0.97321**	0.00416
K-NN - Before	0.06584	0.00512
K-NN - After	**0.99597**	0.00041

The feature transformation phase thus concludes with a reduced, high-signal dataset composed of 103,931 observations and just four attributes: JWDP, JWAP, the derived CTTW, and the ground truth commute time to work (JWMNP). By explicitly encoding the key temporal relationship embedded in the raw features, this process achieves both dimensionality reduction and a sharp increase in predictive fidelity.

2.3 Model Selection and Hyperparameters Tuning

The selection of the optimal regression model emerges as a natural epilogue to the preprocessing phase, which—by far—constitutes the most intricate and transformative stage of the entire analysis. Building on prior evaluations, the Random Forest Regressor is confirmed as the most promising candidate, having consistently delivered superior performance in terms of predictive accuracy. Its coefficient of determination (R^2) reaches a remarkable 0.99691, outperforming all competing models, as summarized in Table 5.

Table 5: R^2 Mean and Standard Deviation for different regression models.

Model	R^2 Mean	R^2 Standard Deviation
Random Forest	**0.99691**	0.00023
Decision Tree	0.99675	0.00032
K-Neighbors	0.99597	0.00041
SVR	0.97321	0.00416
Linear Regression	0.21589	0.04149

To fine-tune the model for maximum performance, a comprehensive hyperparameter optimization phase is undertaken. The development set is randomly partitioned once more into training (75%) and testing (25%) subsets. The grid search cross-validation method is then applied to explore the hyperparameter space and identify the most effective configuration for the Random Forest Regressor. This process reveals that a forest composed of 155 decision trees yields the highest predictive accuracy.

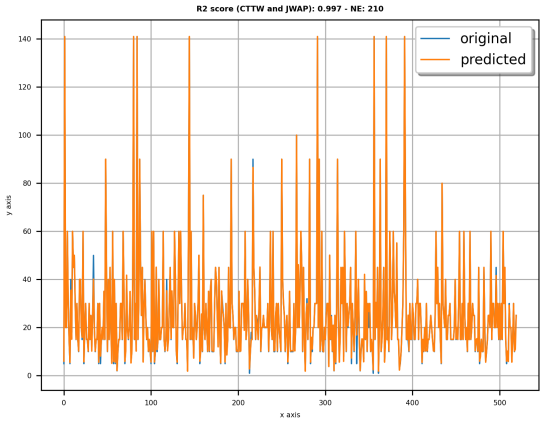
The final model, trained under this configuration, exhibits robust generalization and near-optimal performance, as demonstrated by the R^2 score reported in Figure 4. The result serves as a definitive validation of the methodological pipeline adopted throughout the study: from data preprocessing and feature engineering to model selection and optimization.

In conclusion, the analysis affirms that a careful orchestration of preprocessing, feature transformation, and model tuning can yield regression models of exceptional accuracy, even within the constraints of census-derived categorical data. The Random Forest Regressor, with optimized hyperparameters, stands out not only for its performance but also for its robustness and interpretability, marking it as the model of choice for the predictive task at hand.

3 Conclusion

This study set out with a clear question: can census data offer reliable, statistically grounded insights into predicting commute time to work? The answer, built brick by brick through a careful analysis pipeline, is both subtle and powerful.

Figure 4: R^2 score.



Census data, often seen as a neutral mirror of a population's structure, serves here not only as raw input for prediction, but also as a lens through which to examine the behavior and fairness of the algorithms that rely on it. In this sense, the work is not just technical—it is epistemological. It probes the interface between data and society, asking whether the categories we measure are the categories that matter.

Among the models tested, the Random Forest Regressor emerges as the most effective: a voting ensemble of decision trees that, by combining simple local rules, produces a powerful global estimator. Yet it is not the model alone that carries the weight of insight—it is what the model reveals.

Through a rigorous feature importance analysis, many of the classical socio-economic attributes—income, occupation, education, housing—are found to contribute insignificantly to the prediction task. With importance scores consistently below the 0.5% threshold, these variables do not qualify as causally meaningful within the logic of the model. Once removed, predictive accuracy actually improves (Table 5), exposing a surprising redundancy in features commonly assumed to be decisive.

This outcome is not merely a technical artifact—it has theoretical implications. It suggests a statistical independence between the target variable (commute time to work) and the very attributes most often linked to structural inequality. In doing so, the analysis raises compelling questions about how algorithms process social data and how their internal logic may diverge from prevailing sociological narratives.

Ultimately, the model does not just predict commute times—it clarifies which variables matter and which do not, at least in this specific task. And that, in turn, feeds into a broader reflection: predictive models are not only tools for estimation, but instruments of discovery. Used carefully, they can reveal where our intuitions align with data—and where they do not.

References

[1] P.-N. Tan, M. Steinbach, A. Karpatne, and V. Kumar. *Introduction to Data Mining*, 2nd ed. Pearson, 2019.
[2] J. Han, M. Kamber, and J. Pei. *Data Mining: Concepts and Techniques*, 3rd ed. Morgan Kaufmann, 2011.
[3] K. D. Lee. *Python Programming Fundamentals.* Springer, 2015.
[4] J. VanderPlas. *Python Data Science Handbook: Essential Tools for Working with Data.* O'Reilly Media, 2016.