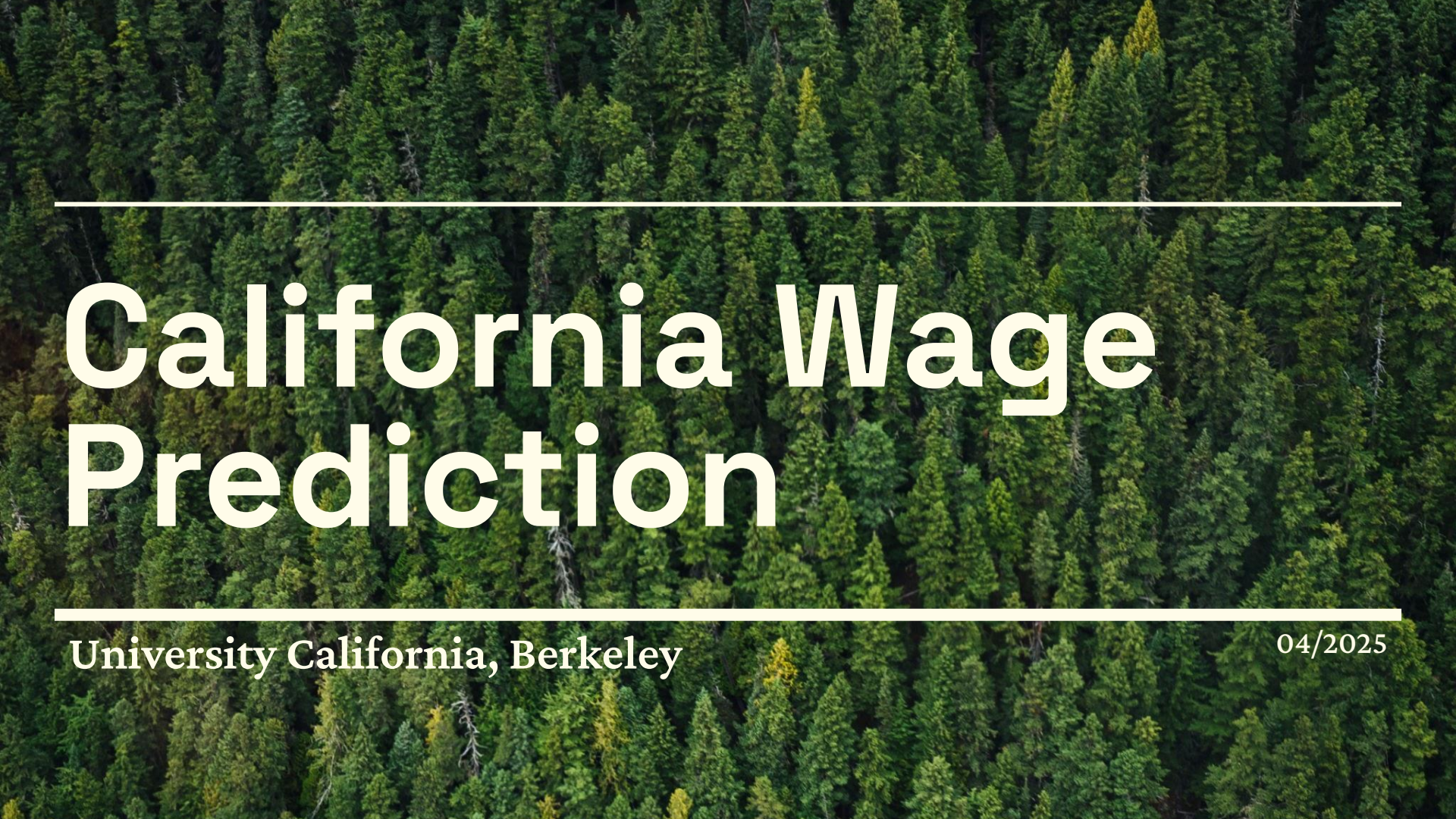


California Wage Prediction

University California, Berkeley

04/2025

The Team

01

Team

Venkat Ramdass

Palo Alto, CA
CTO, Oncept Inc
MIDS expected graduation in Spring 2026

Rohan Krishnamurthi

24, Boston, MA
Data engineer for HarbourVest partners
MIDS expected graduation in Spring 2026

Safiya Alavi

23, San Jose, CA
Research and Data Science Associate at Propel Bio Partners
MIDS expected graduation in Spring 2026

Victor Ndayambaje

Austin, TX
Cloud Data Engineer Consultant at Accenture
MIDS expected graduation in Spring 2026

Motivation

02

Motivation & Research Question

Why This Matters

- Income inequality is a persistent issue in the U.S., with major policy and social implications.
- The high-dimensional, categorical nature of ACS data creates a rich modeling challenge.

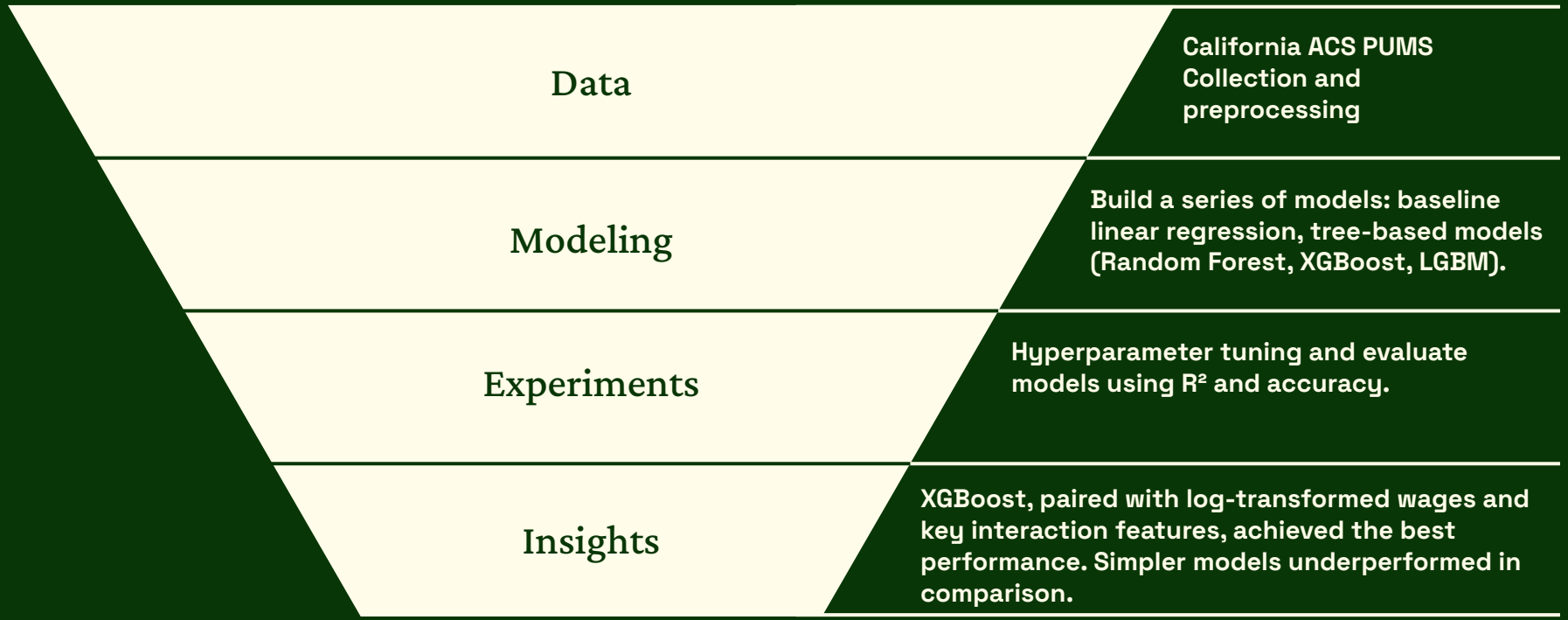
Project Goal & Context

- We use California ACS PUMS data to predict individual wages using demographic and economic attributes.
- Prior research used basic models with limited tuning; we explore advanced ensemble methods.

Key Research Questions

- What features most strongly predict wage levels in this population?
- Can tuned ensemble models (XGBoost, LGBM), combined with thoughtful feature engineering, outperform simpler approaches like linear regression?

Project Plan



Data

03

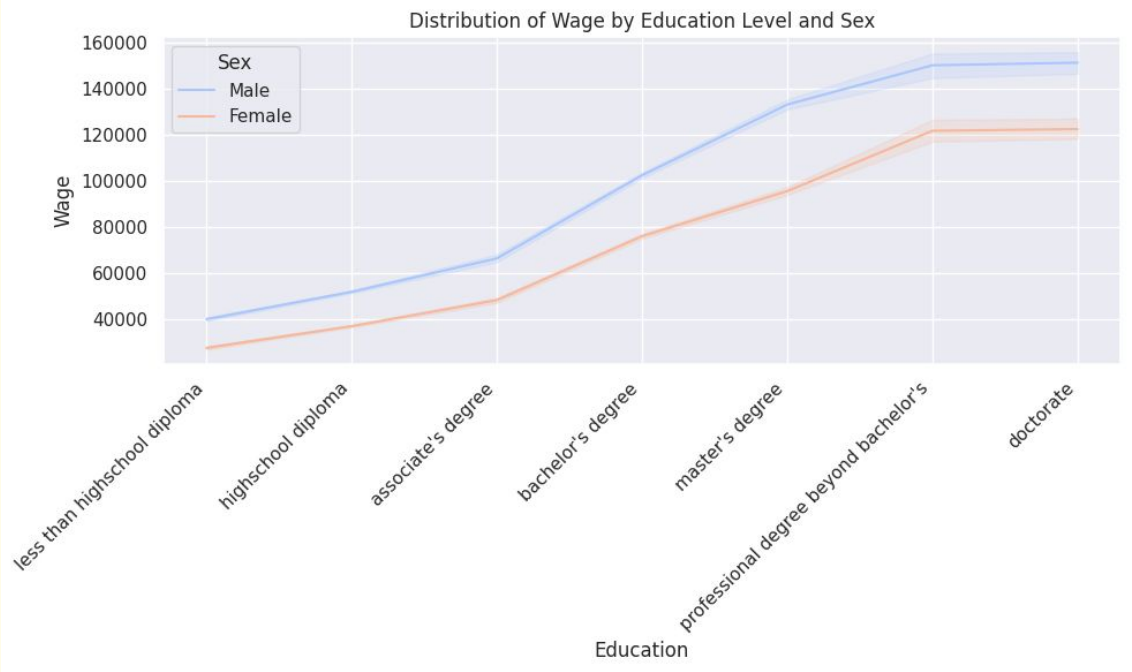
SAFIYA

Data preprocessing

- Data comes from American Community Survey (ACS) conducted by the US Census Bureau on 3.5 million across the country (randomly selected). Subset of California was 390,000 records.
- The data consists of 287 fields. We extracted 17 fields to a separate file that became our source.
- Data was cleaned during field extraction, handling blank value records.
- Some fields were optimized. For example, education level (SCHL) was reduced to contain fewer categories.
- Multiple Race related fields (RAC1P, HISP) were combined to create a new field (RACE).
- Any further processing of data was left for EDA and Feature Engineering.

SAFIYA

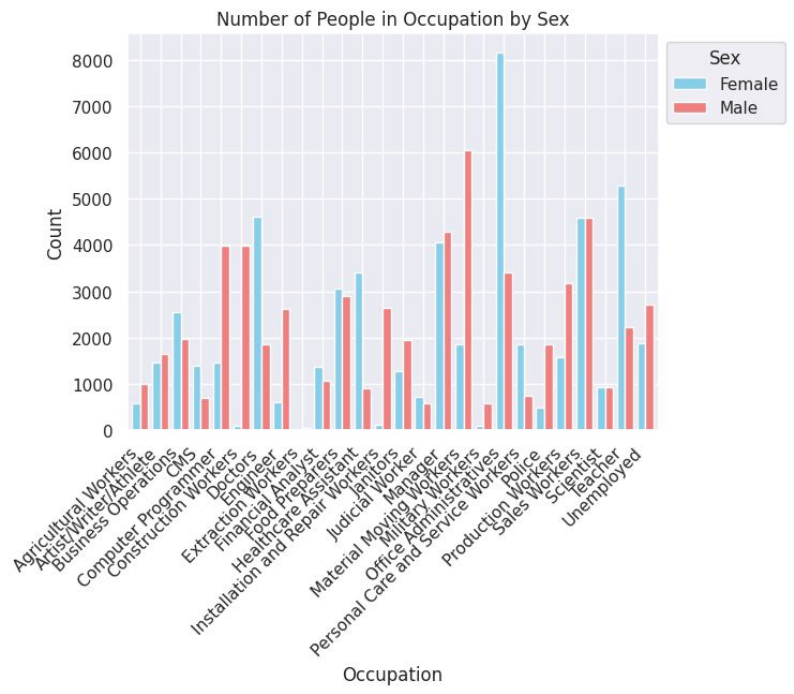
Sex



- **Males are the majority class**
  Data includes 52% males and 47% females
- **Clear wage gap between males and females.**
- **Gap between the wages increases in magnitude as education level increases.**
- **Regardless of gender, the average wage increases as education level increases.**

SAFIYA

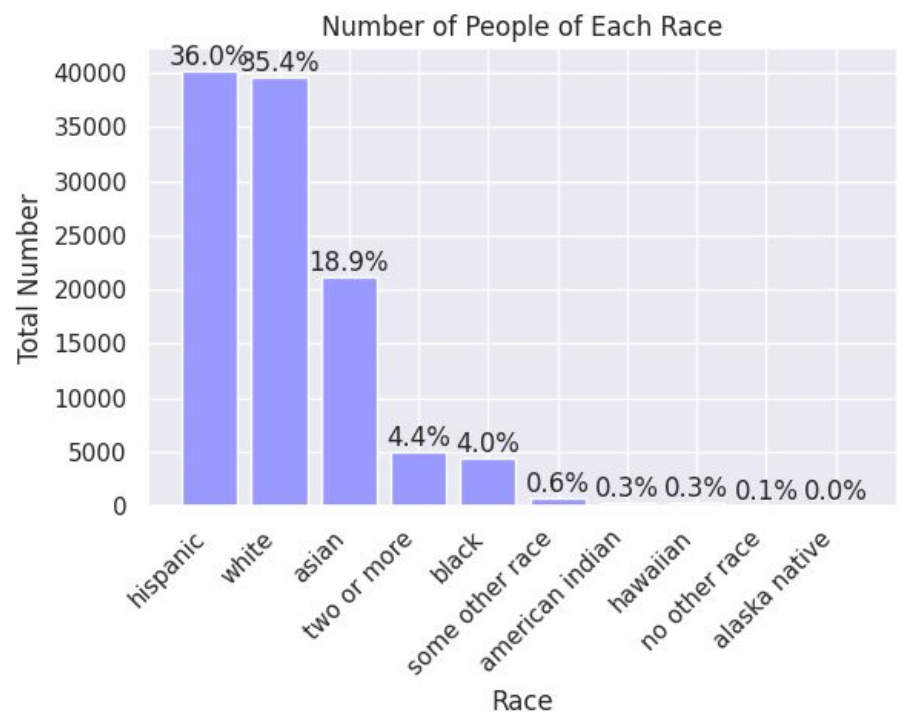
Occupation



- **Plotted the number of people in each occupation by sex**
- **Many fields that are dominated by a specific sex.**

  Female dominated: Teacher, Office Administrative Work
  Male dominated: Military Work, Construction Work
- **Potential relationship sex and occupation**

Race



Population represented accurately compared to actual California census populations

Hispanic and white are the leading races

Hispanic captures anyone of Hispanic, Latino, or Spanish origin

Modeling

04

Section

Baseline Models

KNN

$R^2 = -0.73$

- Lower neighbors generalize better, too many causes memorization
- Simple visuals and analysis to understand trends

Polynomial Regression

$R^2 = 0.73$

- 7 degrees was optimal selection
- Time consuming
- Poor regression capabilities of lower wages

CatBoostRegressor Overview

Gradient Boosting Algorithm

Approach #1 - Baseline Model

$R^2 = 0.56$

- Dropped features with little correlation to wage to reduce the amount of noise in the training data.
- This did not improve the model's baseline performance, indicating that all the features add value.
- In future iterations, I used all features.

Approach #2 - Undersampling

$R^2 = 0.58$

- Removed the top 5% of the wages from the data
- Reduced dataset drastically so there was an equal amount of people in each predefined wage class
- Idea was to prevent the model from overlearning the lower wage features and be able to generalize similar features to higher wages

Approach #3 - Grid Search and Embedding

$R^2 = 0.60$

- Added in embeddings to the worker class, educational attainment, sex and the housing unit to capture additional relationships.
- Ultimately removed the embeddings as they did not improve the results.
- Used Grid Search to find optimal parameters for the model.

Grid Search Results

Optimal Configuration:

{’depth’: 7,
’iterations’: 700,
’l2_leaf_reg’: 5,
’learning_rate’: 0.1}

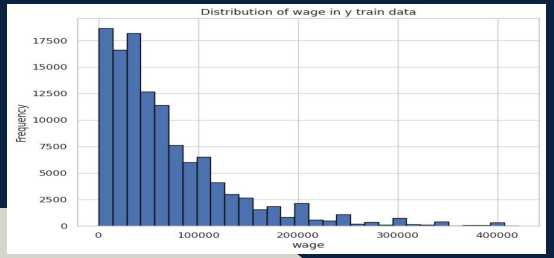
params	mean_test_score	rank_test_score
{'depth': 5, 'iterations': 500, 'l2_leaf_reg': 1, 'learning_rate': 0.05}	-890654527.5	22
{'depth': 5, 'iterations': 500, 'l2_leaf_reg': 1, 'learning_rate': 0.1}	-886437940.1	14
{'depth': 5, 'iterations': 500, 'l2_leaf_reg': 3, 'learning_rate': 0.05}	-892720349.7	24
{'depth': 5, 'iterations': 500, 'l2_leaf_reg': 3, 'learning_rate': 0.1}	-886894749	16
{'depth': 5, 'iterations': 500, 'l2_leaf_reg': 5, 'learning_rate': 0.05}	-892586231.2	23
{'depth': 5, 'iterations': 500, 'l2_leaf_reg': 5, 'learning_rate': 0.1}	-887413005.7	19
{'depth': 5, 'iterations': 700, 'l2_leaf_reg': 1, 'learning_rate': 0.05}	-887096051.3	17
{'depth': 5, 'iterations': 700, 'l2_leaf_reg': 1, 'learning_rate': 0.1}	-883968837.8	6
{'depth': 5, 'iterations': 700, 'l2_leaf_reg': 3, 'learning_rate': 0.05}	-889094449.3	20
{'depth': 5, 'iterations': 700, 'l2_leaf_reg': 3, 'learning_rate': 0.1}	-883990931.1	7
{'depth': 5, 'iterations': 700, 'l2_leaf_reg': 5, 'learning_rate': 0.05}	-889232097.2	21
{'depth': 5, 'iterations': 700, 'l2_leaf_reg': 5, 'learning_rate': 0.1}	-884236338.3	9
{'depth': 7, 'iterations': 500, 'l2_leaf_reg': 1, 'learning_rate': 0.05}	-886000313.4	13
{'depth': 7, 'iterations': 500, 'l2_leaf_reg': 1, 'learning_rate': 0.1}	-884585137.8	12
{'depth': 7, 'iterations': 500, 'l2_leaf_reg': 3, 'learning_rate': 0.05}	-886724595.2	15
{'depth': 7, 'iterations': 500, 'l2_leaf_reg': 3, 'learning_rate': 0.1}	-884464962.5	11
{'depth': 7, 'iterations': 500, 'l2_leaf_reg': 5, 'learning_rate': 0.05}	-887120778.7	18
{'depth': 7, 'iterations': 500, 'l2_leaf_reg': 5, 'learning_rate': 0.1}	-884163869.5	8
{'depth': 7, 'iterations': 700, 'l2_leaf_reg': 1, 'learning_rate': 0.05}	-882839931.5	2
{'depth': 7, 'iterations': 700, 'l2_leaf_reg': 1, 'learning_rate': 0.1}	-884346245.3	10
{'depth': 7, 'iterations': 700, 'l2_leaf_reg': 3, 'learning_rate': 0.05}	-883374548.7	3
{'depth': 7, 'iterations': 700, 'l2_leaf_reg': 3, 'learning_rate': 0.1}	-883718697.9	4
{'depth': 7, 'iterations': 700, 'l2_leaf_reg': 5, 'learning_rate': 0.05}	-883928015	5
{'depth': 7, 'iterations': 700, 'l2_leaf_reg': 5, 'learning_rate': 0.1}	-882775424	1

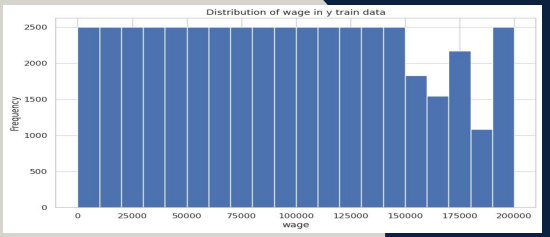
SAFIYA

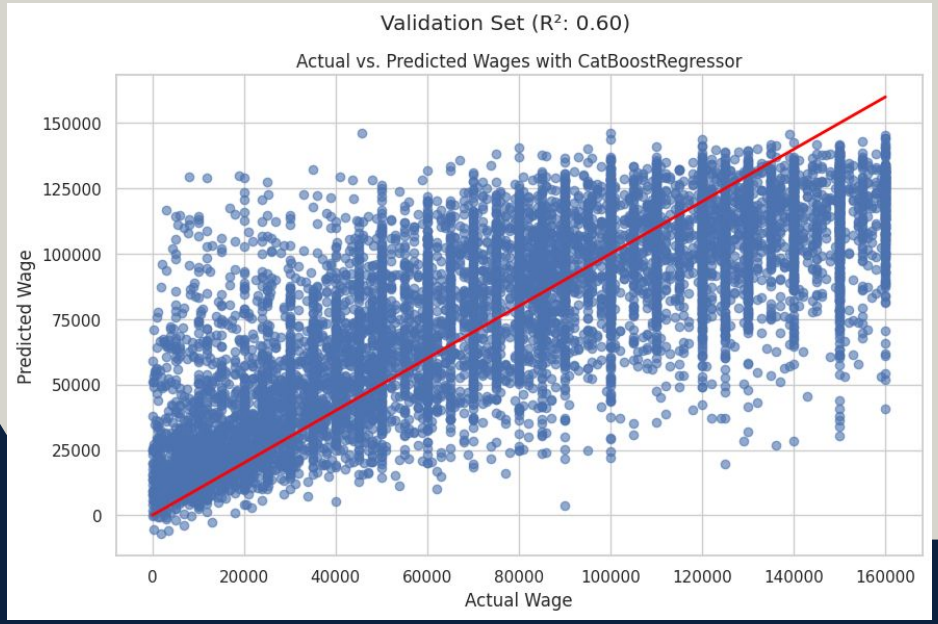
Final Model Results
CatBoostRegressor

$R^2 = 0.60$

MSE = 1,312,826,079







- Undersampling data shown to be most effective in improving the model performance
- MSE so high because model is not predicting higher wages very well but does decently at predicting the lower wages

Tree Based Models

XGB
$R^2 = 0.71$

- Good initial results showed promise
- Tuning was time consuming, lead to explore LGBM
- LGBM would serve as intermediary to identify model config

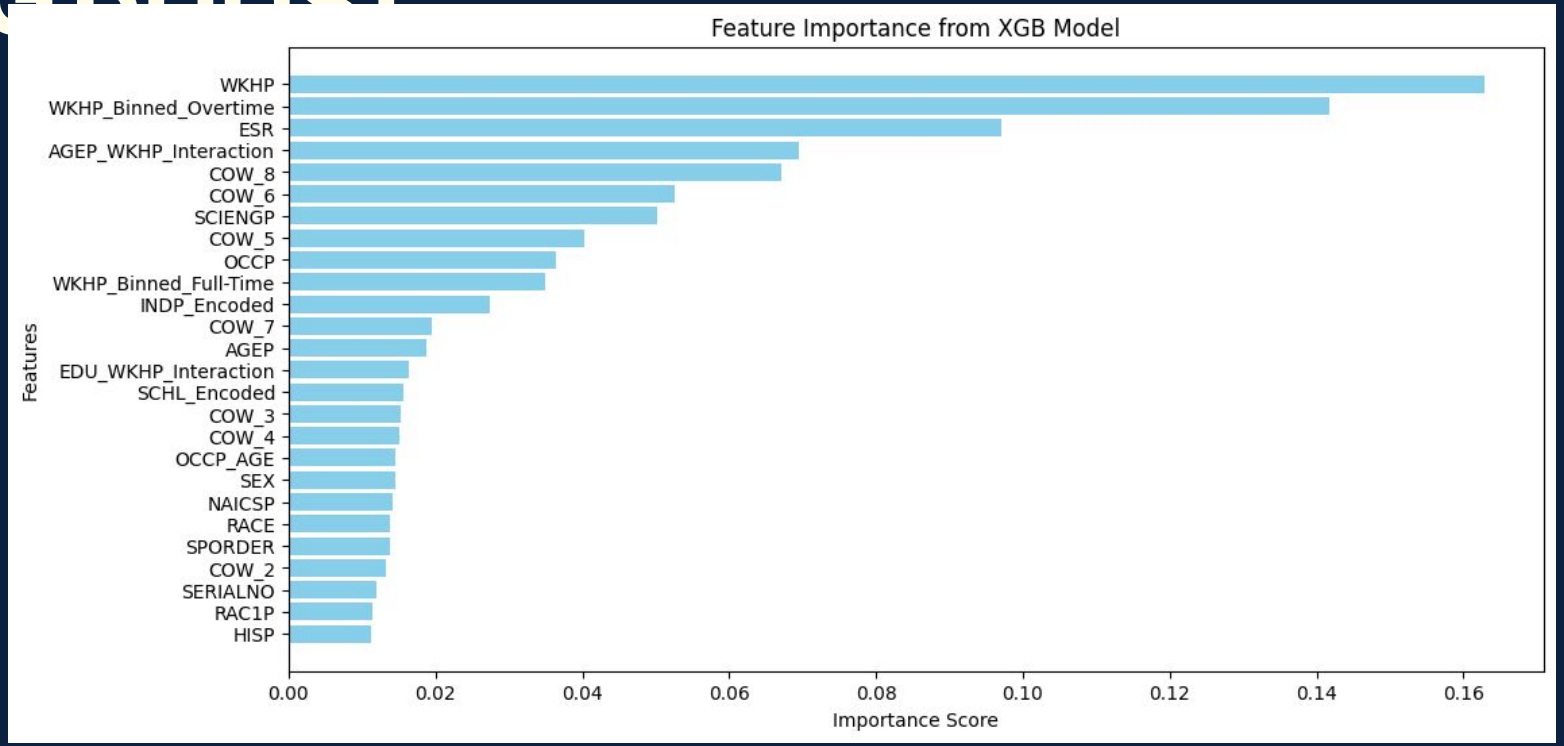
LGBM
$R^2 = 0.637$

- More efficient XGB with large dataset
- Facilitated hyperparameter tuning
- Yields average results, great for experimentation

XGB-LGBM Blend
$R^2 = 0.64$

- Idea: combine two best performing models to achieve improved results
- Interesting to play with blend weights
- Overly complex

Feature Importance XGBoost

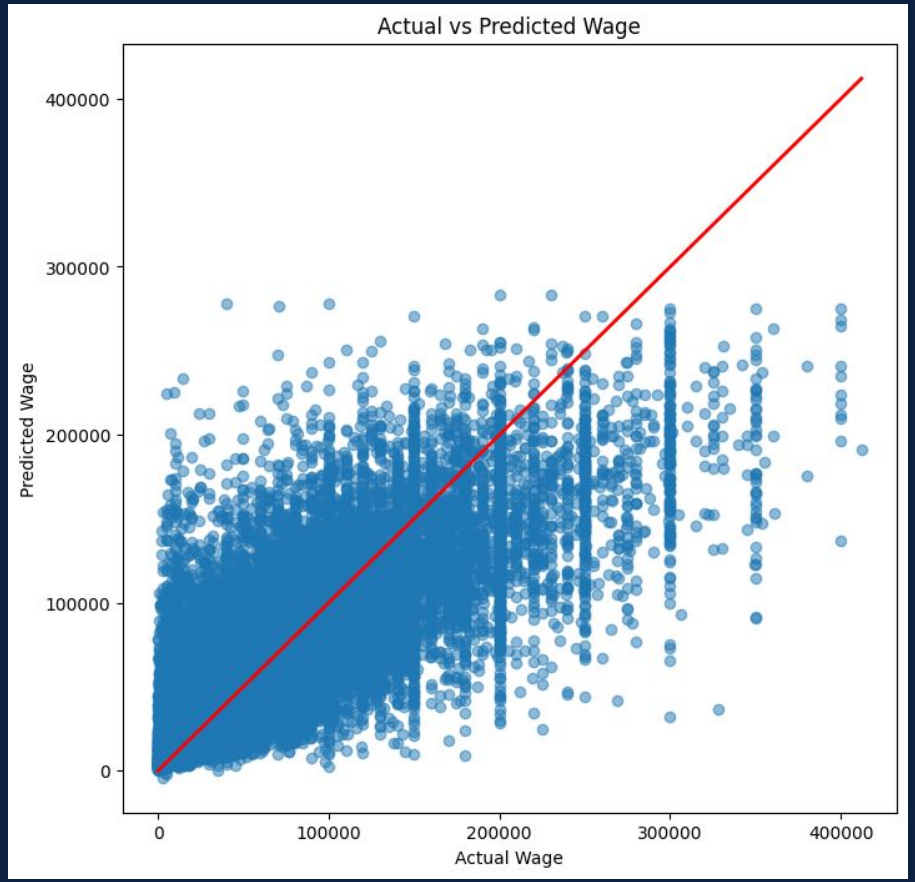


Feature Engineering

Square influential features	Surprisingly little to no improvement to model accuracy Cubing and raising to the fourth did not assist either Adding unnecessary complexity to the model
Combine top features	Helped model tremendously, captured hidden relationships Boosted accuracy most of the feature engineering tactics attempted Only relevant for ~ top 5 base input features
Remove bottom features	Little to no improvement to model accuracy Emphasized that more features allowed for greater accuracy Features that may seem redundant can add significant, hidden value

Blend Results

- R-squared = 0.64
- Poor outlier prediction
- Ideal blend found to be: LGBM = 0.52, XGB = 0.48
- Exhaustive feature engineering did not yield great improvement



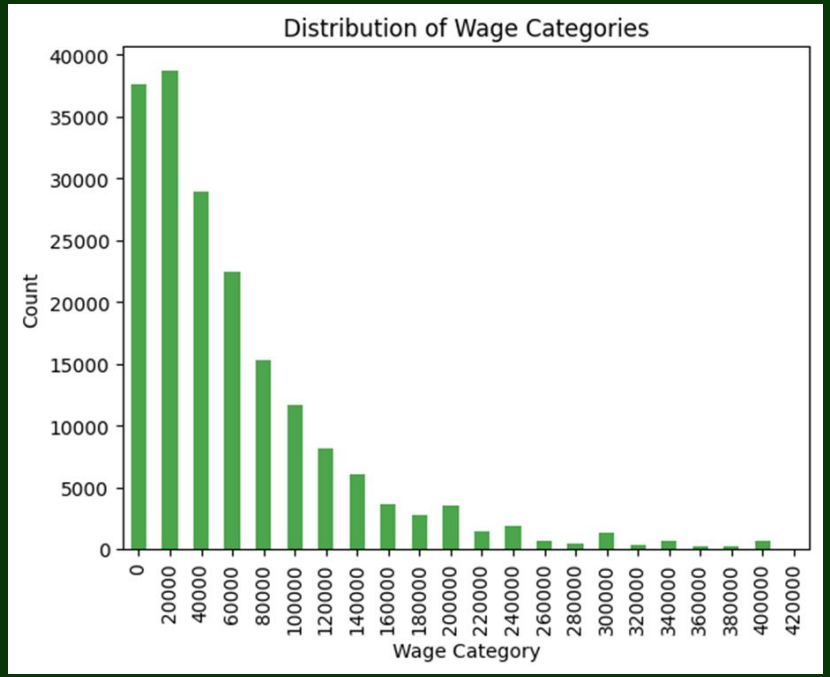
Experiments

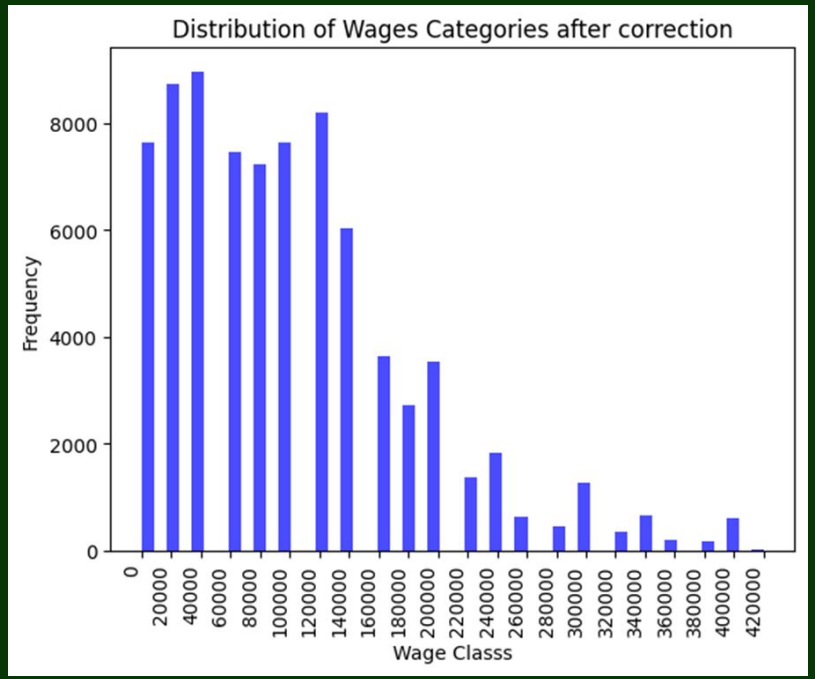
05

Further Data Analysis

Imbalance correction

We deleted sufficient number of records from the lower wage categories to create a more balanced dataset. Wage distribution after this exercise was more balanced.





Hyperparameter Tuning Table

- Influential hyperparameters on XGBoost shown here
- Train and test R squared selected as main target metric
- Ideal model: max_depth = 6, LR = 0.05, n_estimators = 300
- Facilitated model development greatly

Hyperparameter Effects on R^2 Score

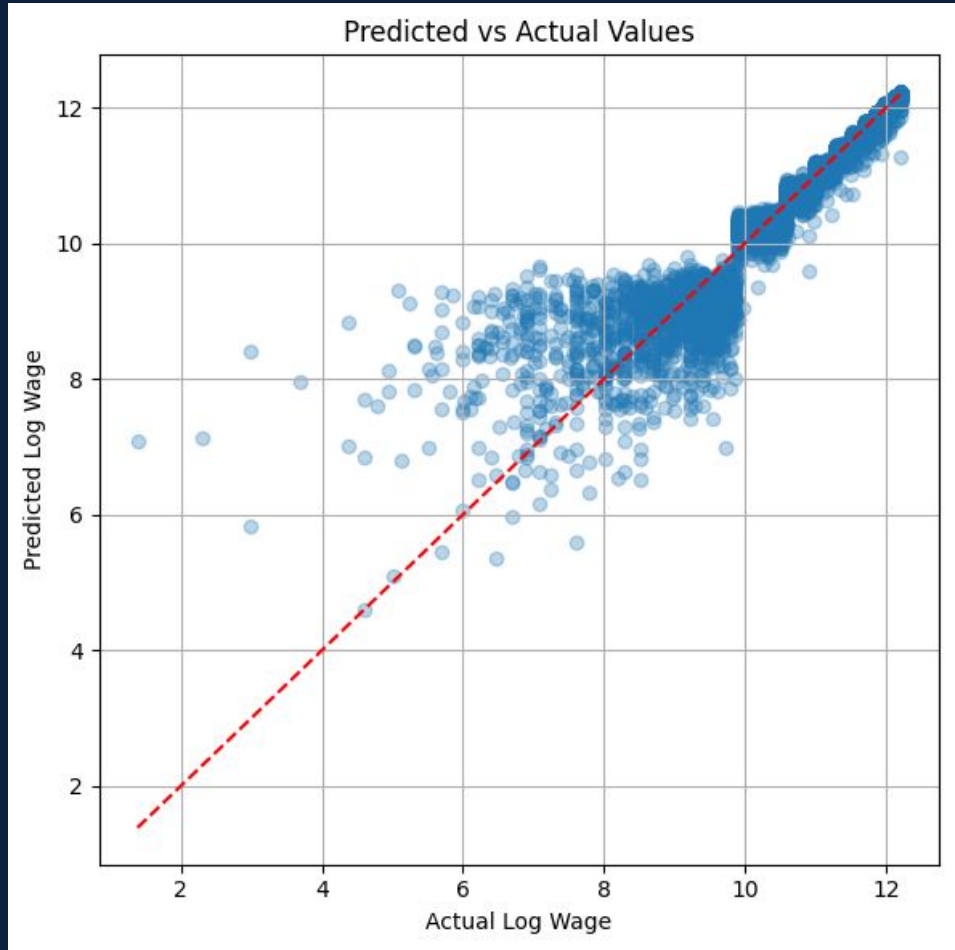
	hyperparameter_tested	max_depth	learning_rate	n_estimators	mean_train_r2	mean_test_r2
0	max_depth & learning_rate	6	0.01	300	0.9192	0.9054
1	max_depth & learning_rate	8	0.01	300	0.9366	0.9048
2	max_depth & learning_rate	10	0.01	300	0.9540	0.9026
3	max_depth & learning_rate	12	0.01	300	0.9676	0.8992
4	max_depth & learning_rate	6	0.05	300	0.9545	0.9115
5	max_depth & learning_rate	8	0.05	300	0.9789	0.9091
6	max_depth & learning_rate	10	0.05	300	0.9933	0.9065
7	max_depth & learning_rate	12	0.05	300	0.9984	0.9051
8	max_depth & learning_rate	6	0.10	300	0.9713	0.9078
9	max_depth & learning_rate	8	0.10	300	0.9914	0.9050
10	max_depth & learning_rate	10	0.10	300	0.9981	0.9044
11	max_depth & learning_rate	12	0.10	300	0.9997	0.9026
12	n_estimators	12	0.05	100	0.9819	0.9036
13	n_estimators	12	0.05	200	0.9938	0.9043
14	n_estimators	12	0.05	300	0.9970	0.9037
15	n_estimators	12	0.05	400	0.9983	0.9033

Final Model

06

XGBoost Model

- Key steps taken to achieve model :
    - Imbalance correction
    - Log applied to target wage
    - Outlier handling
- XGBoost identified to be optimal model (R^2=0.91) through experimentation of tree based models
- Extreme outliers remain, further handling to improve accuracy further



Thank You!

Github repo:
https://github.com/UC-Berkeley-I-School/w207FinalProject