# Predicting Total Compensation

Mason Lee, Annie Wang,
Hans Baumberger, Sileshi Hirpa

# Content

❖ Problem statement
❖ Data cleaning and EDA
❖ Data Preprocessing and Modeling
❖ Model Selection
❖ Conclusion and Discussion

# Problem Statement

Problem Statement

Predict employee total compensation based on the professional features, companies, and macro-economy features

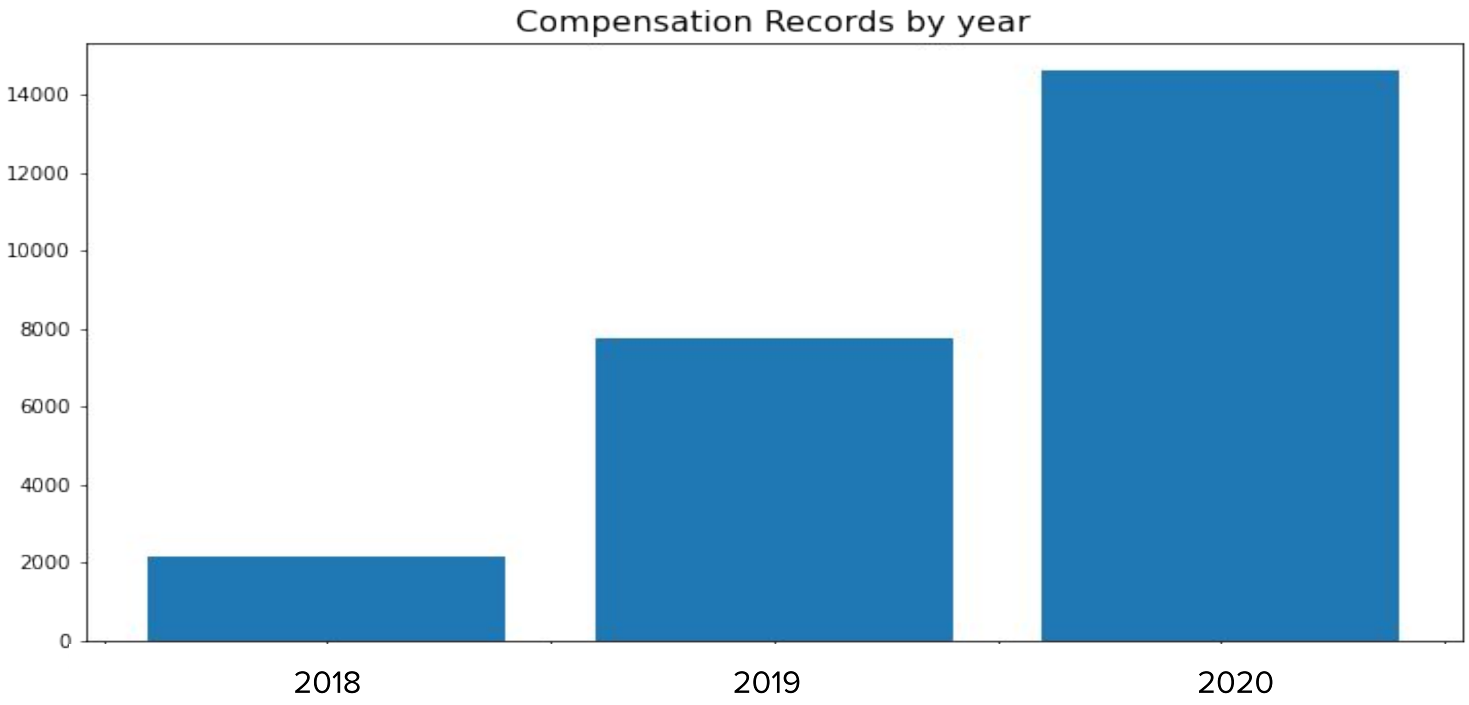
Goals:

i. Provide a reasonable expectation for compensation negotiation
ii. Provide an important benchmark for the talent competition (competitive compensation package in recruitment)

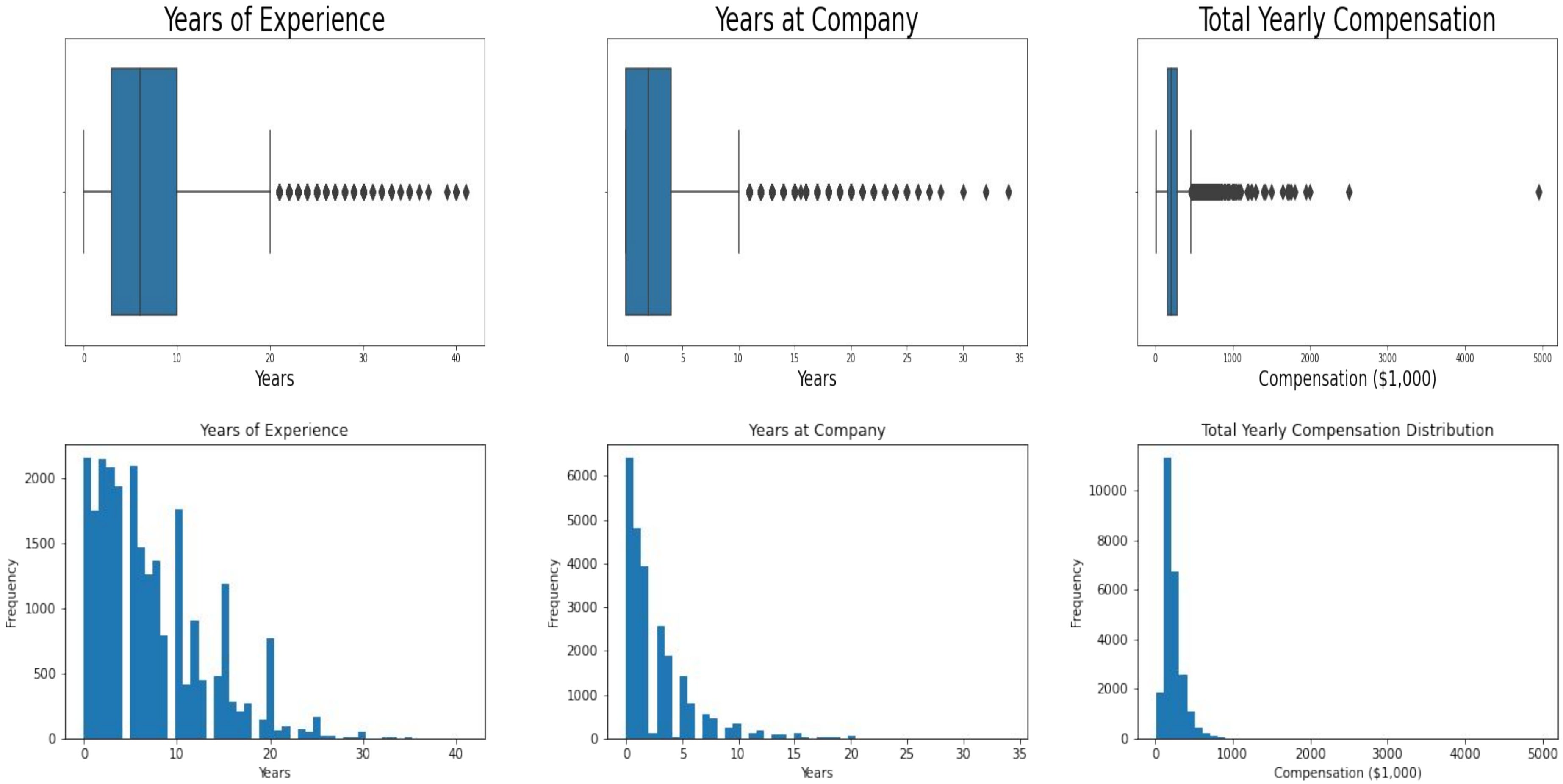
# Data

- The primary dataset from Levels.fyi by web scraping with the  permission from the company
    - Total compensation
    - Company, job type, office location
    - Years of experience, years at the company
    - Submission time
- Inflation rate and unemployment rate
- The final dataset
    - Time span: Jan 2018 to Sep 2020.
    - 24,496 records and 11 features
    - 1219 features after one hot encoding

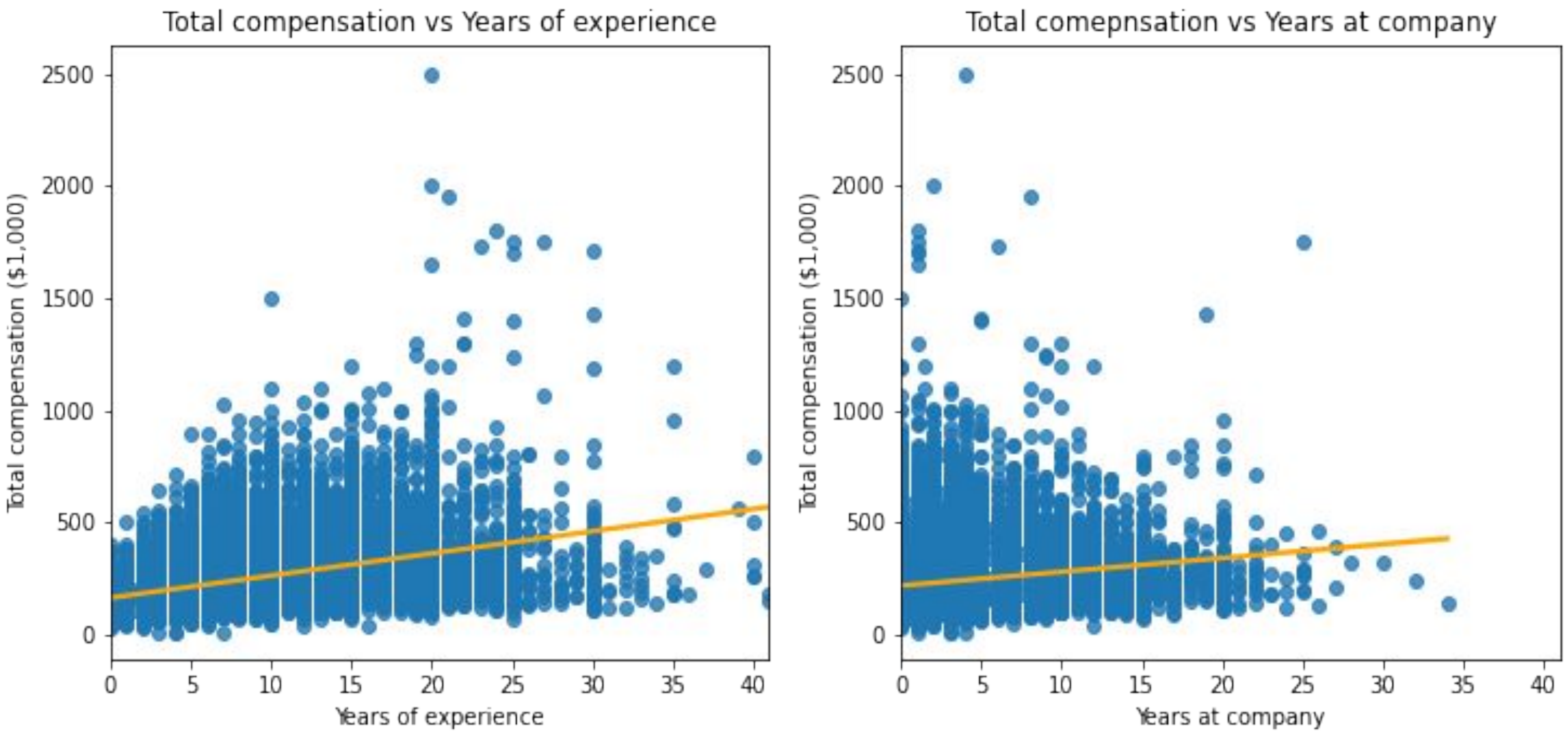
# Record Distribution



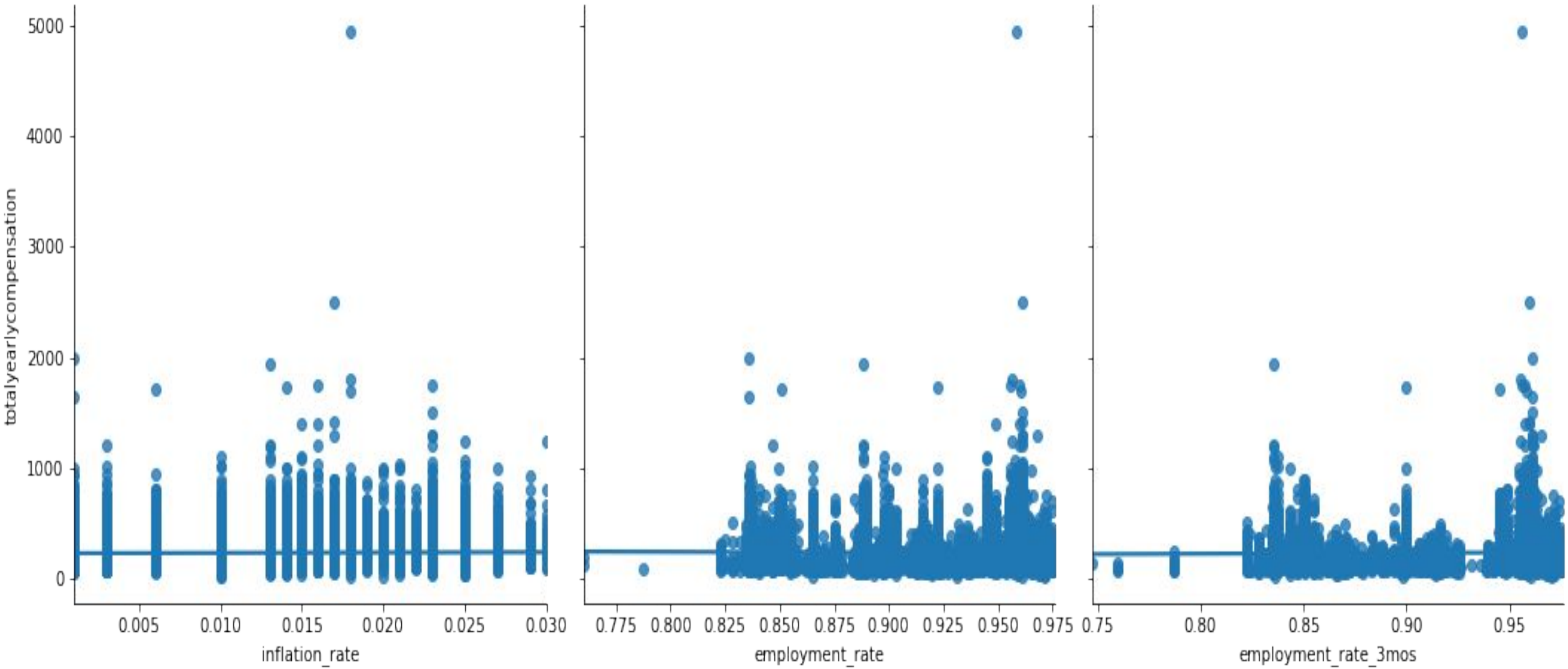
# EDA (cont.)


Years of Experience
Years
Years at Company
Years
Total Yearly Compensation
Compensation ($1,000)
Years of Experience
Frequency
Years
Years at Company
Frequency
Years
Total Yearly Compensation Distribution
Frequency
Compensation ($1,000)

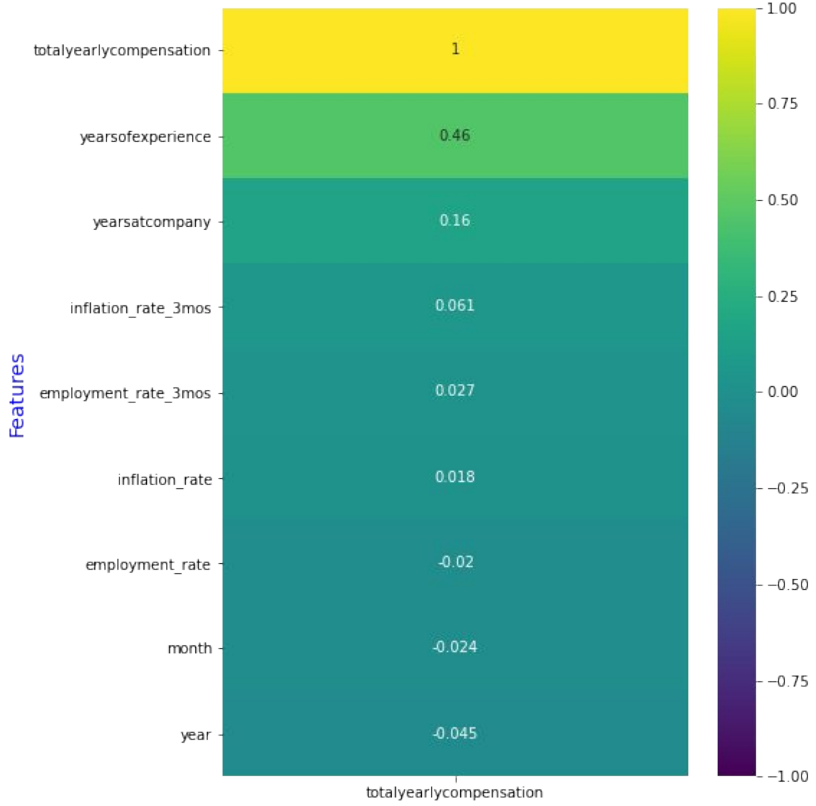
# Total Compensation & Experience



# Total Compensation & Macro-Econ Features

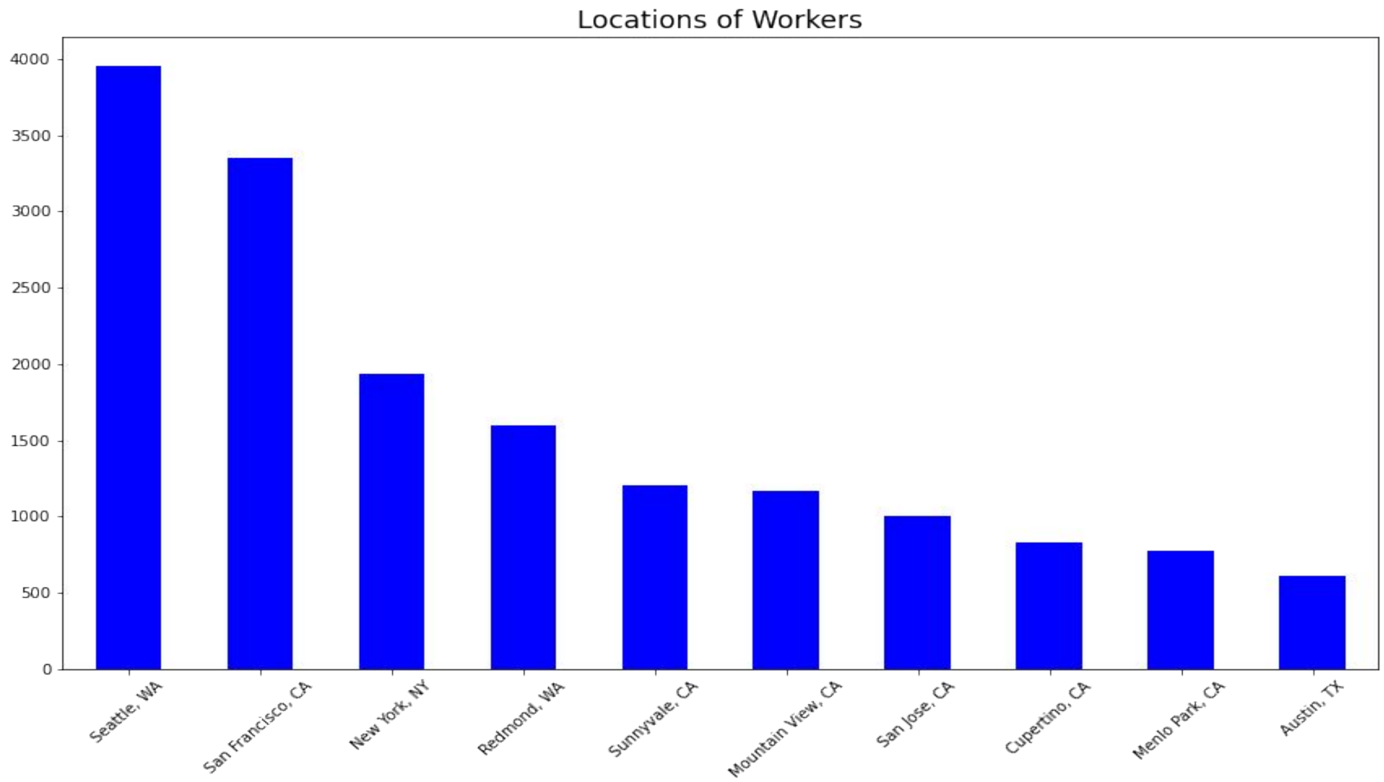


# EDA (cont.): Correlation with Total Comp

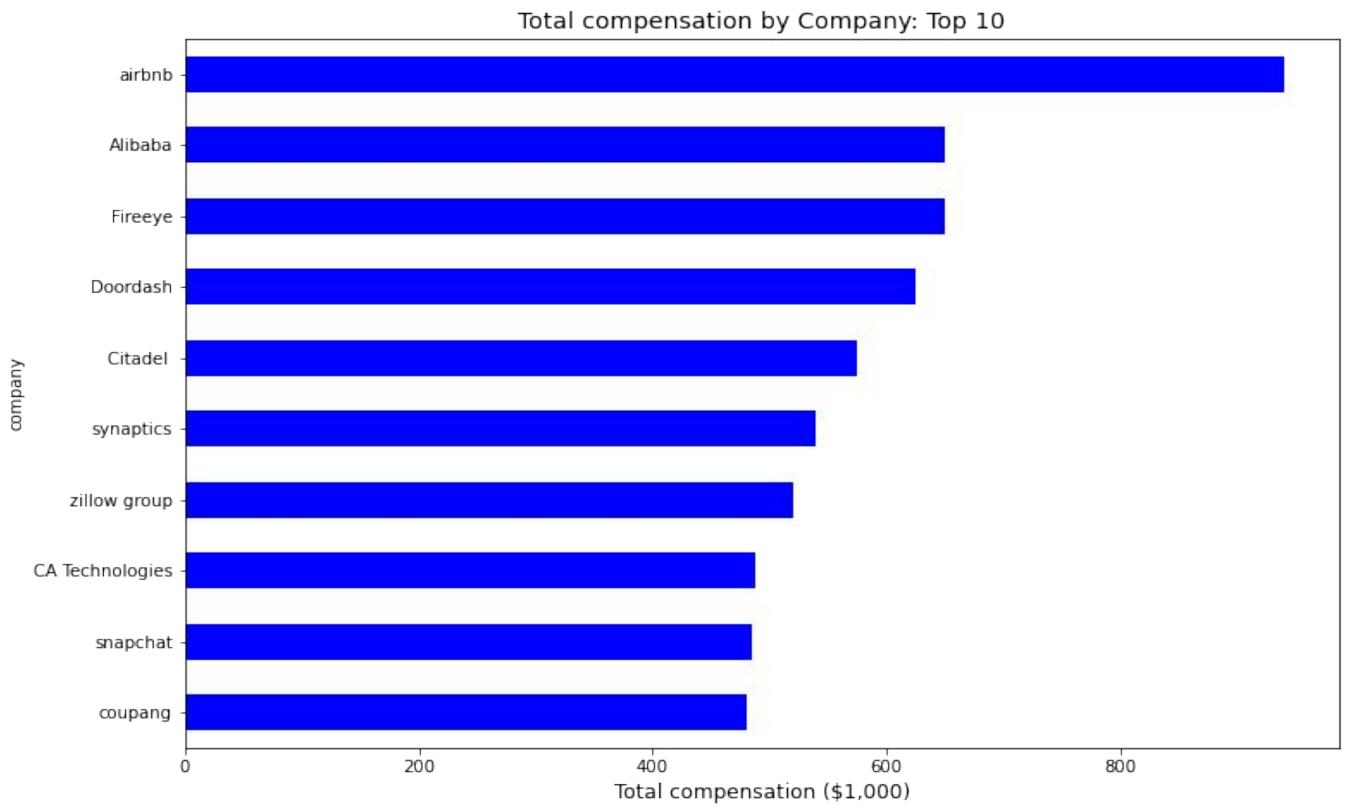



Features' correlation with total yearly compensation

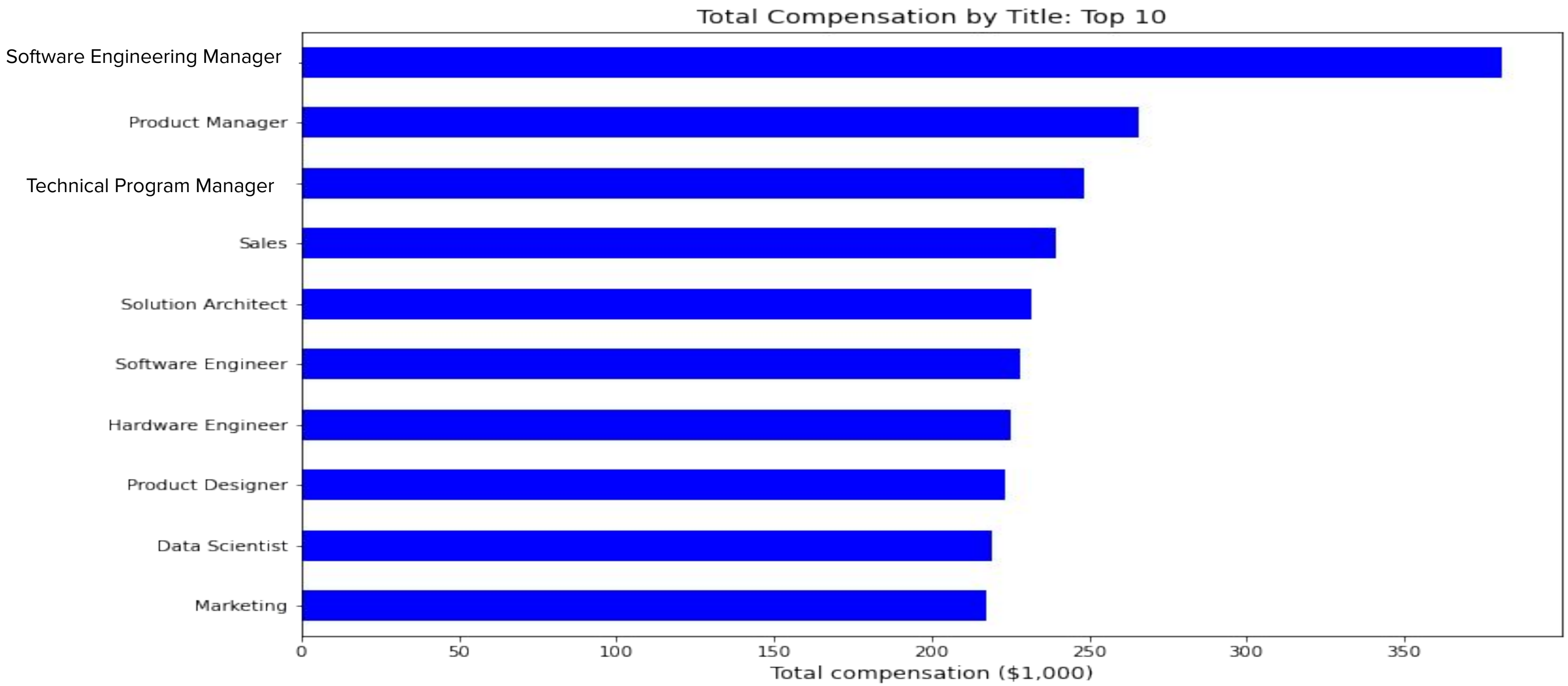
# Distribution of records by office location



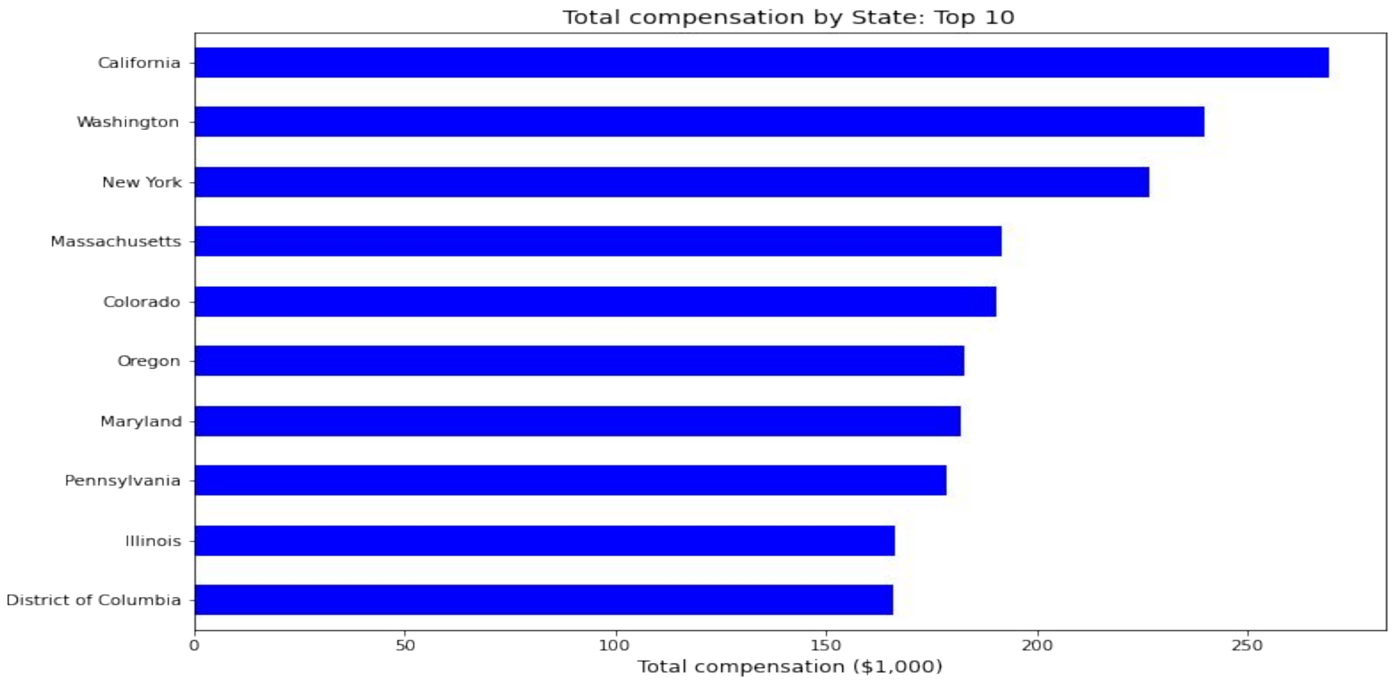
# Total Compensation of Top 10 Companies



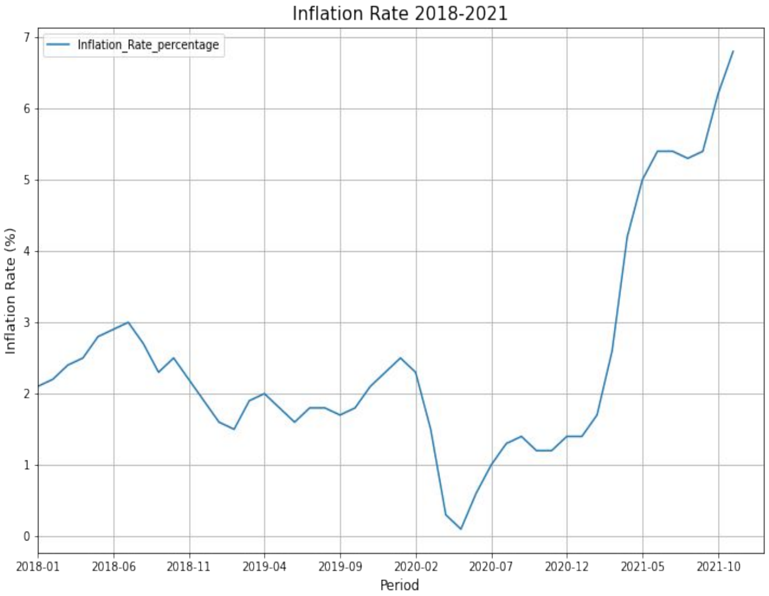
# Total Compensation of Top 10 Titles

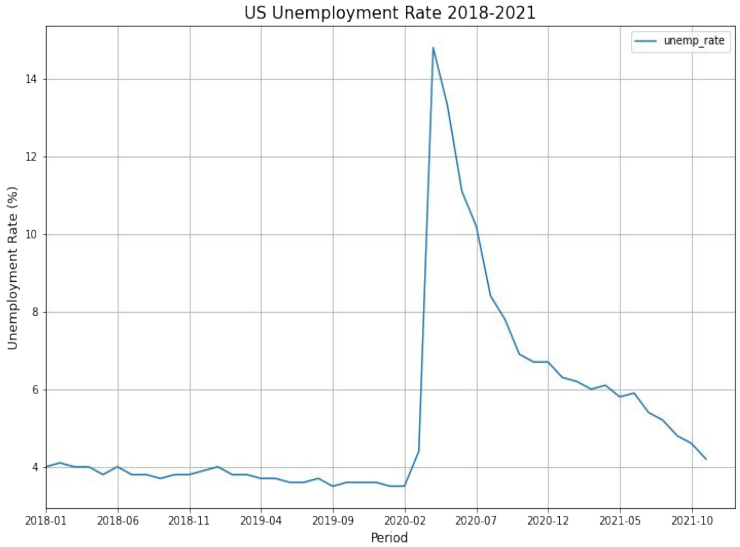


# Total Compensation of Top 10 States



# Macroeconomic Factors



# Models

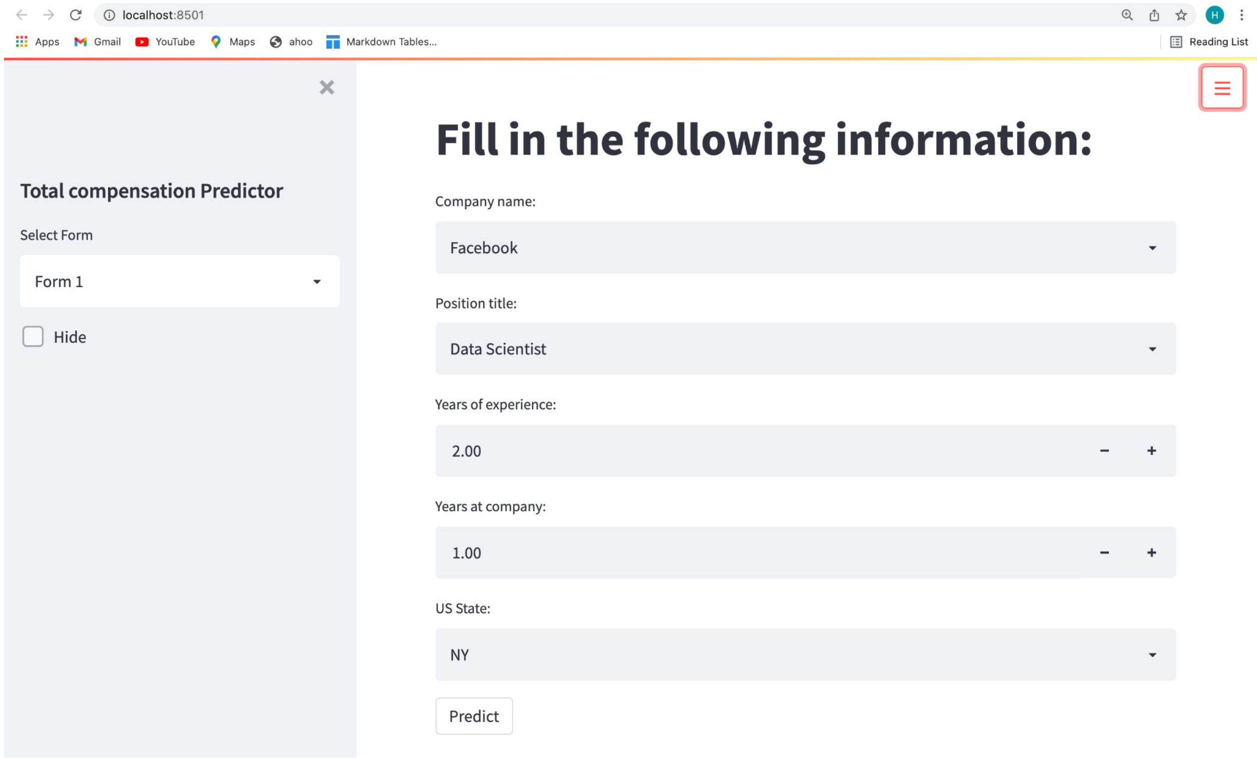
- Linear Regression
  - LASSO, Ridge, Elastic Net
- KNN Regressor
- Gradient Boosting Regressor
- RandomForest Regression (**worked on AWS**)
- AdaBoost Regression
- Support Vector Regression
- ~~Neural Network~~

# Models Performance

| Model | Train (R2) | Test (R2) | Train (MSE) | Test (MSE) | Comment |
|---|---|---|---|---|---|
| **Linear Regression** | 0.5193 | - 7.2931E^28 | 8286.35 | 1.22E27 | |
| **Lasso Regularization** | 0.5182 | 0.5143 | 8305.30 | 8157.45 | |
| **Ridge Regularization** | 0.52 | 0.5097 | 8274.20 | 8234.28 | |
| **Elastic Net Regularization** | 0.4483 | 0.4499 | 9511.19 | 9238.88 | |
| **Random Forest Regression** | 0.466 | 0.410 | 9060 | 10319 | |
| **AdaBoost Regression** | 0.1930 | 0.1276 | 13693 | 15276 | |
| **KNN Regressor** | 0.9907 | 0.4762 | 158.3478 | 9172.5505 | |
| **Gradient Boosting Regressor (without GridSearch)** | 0.5973 | 0.5318 | 6834.12 | 8198.40 | |
| **Gradient Boosting Regressor (with GridSearch)** | 0.7131 | 0.5477 | 4867.52 | 7919.98 | **BEST MODEL** |
| **Support Vector Regression** | 0.1368 | -0.1287 | | | |
| **SVR (with GridSearch)** | 0.5029 | 0.4745 | | | |

# Compensation Prediction

Streamlit



# Highlights

- A very interesting and relevant business question
    - Interesting (and rare) datasets
- Heterogeneous data (categorical + numeric)
    - ColumnTransformer vs. OneHotEncoder vs get_dummies
    - 1219 features with over 1000 as dummy variables
- Complex models and transformers
    - AWS
    - Pickle them
- Streamlit
    - Built the webpage
    - Worked with unpickled models in notebook but stuck with streamlit code on ColumnTransformer
- Team collaboration
    - Each member participated in key stages: collecting data, EDA, modeling and creating presentation materials.
    - Each member volunteered to take charge in one aspect

# Discussion & Next Step

- Rooms for Improvement
  - More features about the employee, the company and industry, and macro-economy
  - Include more current data
  - More hyperparameter search (grid search, randomized search, bayes search)

# Thank you!