MODELING BENEFITS PROGRAMS FOR CITY EMPLOYEES

HARINI LAKSHMANAN

UYEN PHAM

STEPHEN REAGIN

THE PROBLEM

Compensation is an important subject for all workers, including city employees in the state of California. There are fair pay and equality considerations, i.e. *equal pay for equal work*, but budgets are funded through taxes and voters don't want to overspend

- Workers deserve to know if their benefits are generally in line with similarly-situated peers
- Cloud computing resources, combined with appropriate data, could provide a solution for employees and taxpayers to ensure people are paid fairly, equitably, and responsibly

We propose using city compensation and payroll data to assess whether employee *benefits* programs are predictably valued and aligned with work or job performed

COMPENSATION DATA

We analyzed **1.4 million employee records** across 10 years of payroll data from multiple city sources:

- Los Angeles
- San Francisco
- San Jose

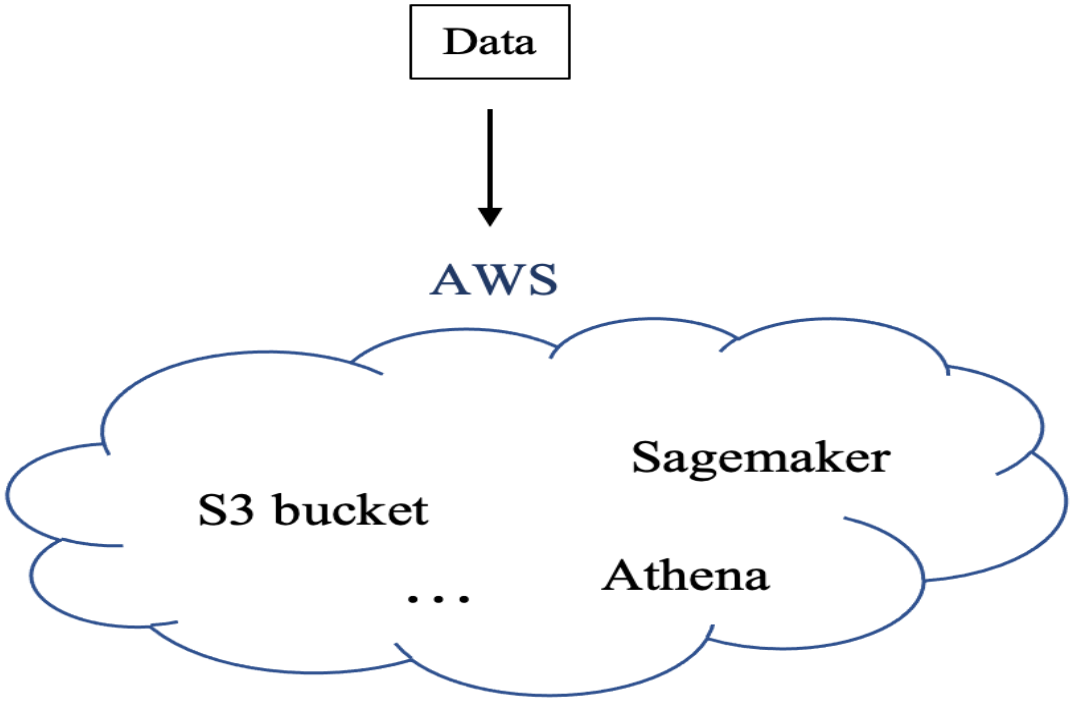
Common components of compensation include:

- Cash compensation
- Salary / Overtime / Bonuses / Irregular cash
- Other program
- Benefits / Medical / Retirement / Pension

We built **regression models** to assign a predicted value for each employee's Total Benefits

These models were developed and deployed in the cloud through Amazon Web Services (AWS)

DATA IN THE CLOUD



- Data was loaded into AWS, an Amazon cloud service which is cost effective and provides useful tools such as
  - S3 bucket (storage)
  - Sagemaker (machine learning platform)
  - Athenal (interactive query service)

- Data was preprocessed, explored, transformed and feature engineer, etc. preparing for modeling. Various techniques (data cleaning, normalization, etc.) were applied to ensure that our results were accurate and reliable

DATA EXPLORATION

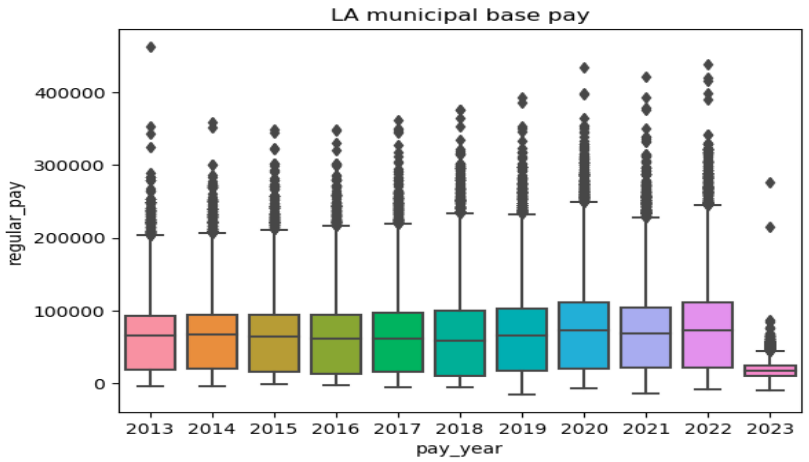


Figure 1 - *Boxplot of base pay over the years in LA*

- Base pay average over the year in L.A was around $70K
- Numerous outliers. The same is true for the other cities. They were considered important for our training

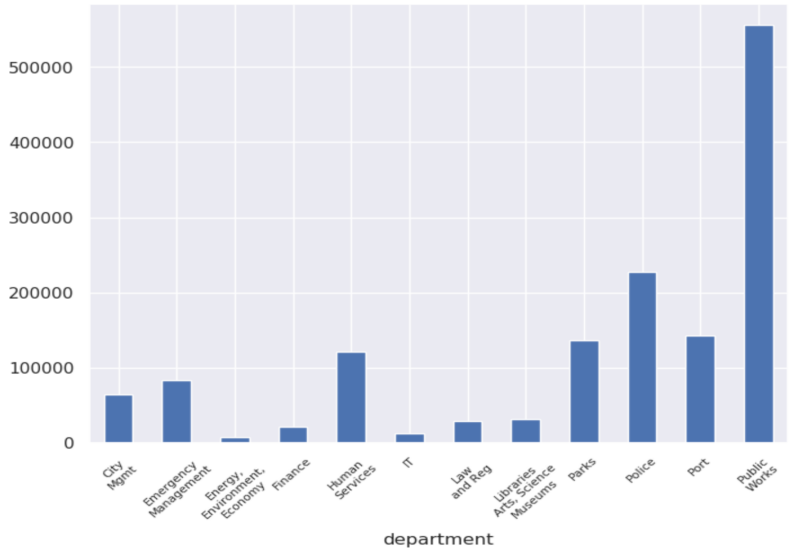


Figure 2- *Department distribution*

Department was condensed from >5K categories into 12 categories with public work having the most employees

The final data input for modeling includes features such as ***base salary, paid year, department, overtime, irregular cash, city id, average annual cpi and total benefits*** with total benefit being the target feature.

Train and test split is 75: 25 ratio

DATA ANALYSIS

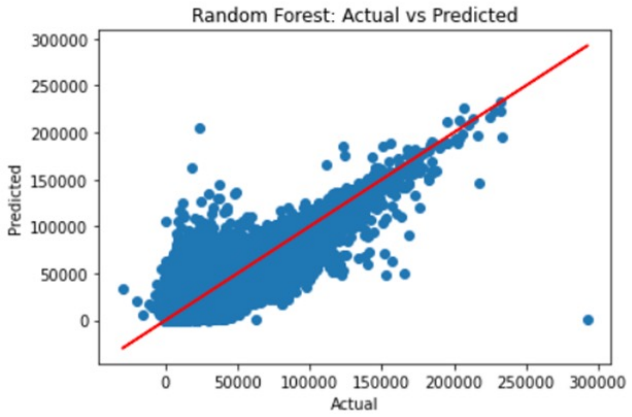
- Objective: Predict Total Benefits for a given job description based on various metrics
- Total benefits is a metric expressed in dollars and hence a continuous variable. So, we used various regression techniques to predict the benefits namely,
  - Linear learner
  - Random Forest
  - Decision Tree
  - XGBoost
  - Ridge regression
- Also, we tried using Auto pilot in Sage Maker, to get the three best models.
- We used the Test data to train our various models and used the trained regression model to predict the Total benefits of the Train data and compared with the actual Total Benefits.
- To understand how accurate our prediction is we used Statistical measures like **R-squared value** and **Root mean square error**.

RESULTS

Table 1:Random Forest Results

R2 square:	0.935
MAE:	3410.7898
MSE:	37859078.3365
RMSE:	6152.9731

Fig 3: Random Forest Actual vs Predicted Benefits in $USD:



- The initial model used was the AWS-native estimator "Linear Learner", which took roughly 36 minutes to train and roughly 5 minutes to make predictions RMSE of ***11278*** value is quite high.
- Decision tree, XGBoost and Ridge regression had RMSE's of ***8313.3, 6416.2*** and ***11262*** respectively
- Based on our results Random Forest showed the best goodness of fit with the lowest RMSE of ***6152.9*** and an R-squared value of ***0.935***.
- Just to explain R-square of 1.0 is the perfect fit and we can say that our predicted values of Random forest and the actual outcome are highly correlated.
- We have chosen to deploy the ***Random Forest*** model for future predictions.

RECOMMENDATIONS AND FUTURE WORK

Our AWS models predicted a city employee's Total Benefits value with greater accuracy than the overall standard deviation of $22K

However, this is not accurate enough for a given employee to determine whether or not they are receiving fair benefits

In addition, the use of AWS-native algorithms did not improve model performance beyond those of local Python libraries

We recommend:

- Disclosing additional compensation inputs
  - Years of experience or tenure
  - Career levels
  - Salary bands
  - Itemized benefits programs
- Bring your own algorithm script
  - Cloud-native modeling solutions do not offer performance improvement over standard libraries
  - Using cloud servers, storage, and compute resources for modeling is a cost effective solution

THANK YOU!

Please visit our GitHub to learn more:

https://github.com/unpham/ads-508-project/