Diversity in the Workplace

William Baron, Tiffany Lee, Emma Peatfield, Yudhvir Singh, Sterling Tarng
Computer Engineering Department
San Jose State University (SJSU)
San Jose, CA, USA
Email: {william.baron, tiffany.lee, emma.peatfield, yudhvir.singh. sterling.tarng}@sjsu.edu

***Abstract*—Diversity, unconscious bias in the workplace and, in general, the way companies treat their employees are a very important topic. Data science can help discover potential discrimination by looking at the data and see if there are segments of employees that are treated wrong.**

***Index Terms*—Workplace discrimination, Data Science, Preprocessing data, Prediction model, Random Forest prediction**

I. Introduction

The main goal of the project is to find signs of discrimination or bias in how companies treat their employees. Due to large number of employees in different companies it is not easy to find the discrimination or bias. but with the help of data sciences we can process a large number of employees and check the discrimination by checking the deviation.

We pre-processed the given data so it is easier to train and test large number of employee data. Then we identified for each employee rank and categorized them into their respective ranks. Also we identified who's working under who. We predicted the salary of each of the employee which is then will be tested to their actual salary to identify the discrimination or bias in the employees of the company. At last using random tree forest we calculated compared the predicted salaries to the actual salaries to check the bias.

II. System Overview

For our machine learning model we first pre-processed the data. After pre-processing the data: - Determine the relevant features from our employee data - Build and run our prediction model to predict salary Determine the importance of each feature relative to the others

A. Motivation

Bias and discrimination is very important in all of the companies because people with different backgrounds work for the same company.Specially in places like Bay area people come from all over the world. So to maintain the fairness in the companies, all the employees should be treated fairly and equally. But to check the discrimination or bias its not easy to manually go over through large number of employees. So to solve this important issue data sciences can help identify the problem

B. Initial Solution and Failure

Our initial solution was to use a collaborative filtering recommendation approach in a parallel way in order to determine whether or not there was bias. First we would preprocess our data and isolate key features that we wanted to compare. Then, we were going to perform a series of collaborative filtering recommendations on salary in parallel based on just a singular feature at a time. The idea was that, if we can predict a salary based on a single feature for each of the features and then compare them, any extreme variance in predicted salary indicates bias. However, the problem was that recommendation and prediction are very different. We need a way to validate our prediction and a recommender system does not provide that for us.

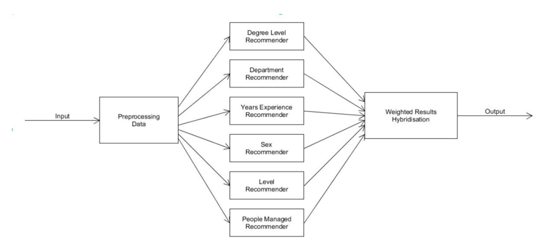

Fig. 1. Initial System Architecture.

C. System Design

Equations: Given training set X with responses Y. B is number of samples hatf is regression tree

$$\hat{f} = \frac{1}{B} \sum_{1}^{B} f_b \left(x' \right) \tag{1}$$

After training, predictions for unseen samplesx' can be made by averaging the predictions from all the individual regression trees on x'

III. Technical Discussion of Pre Processing of Data

For our given data set of employees we have to tabulate the data in format which is easily then trained and tested by the machine learning algorithm. We also checked which employee has what rank and what employees work under each employee.

- Determined the level of each employee
- Identified how many employees each person manages
- Encoded the values for non-numeric features

```
In [15]: df = pd.merge(df_emp,
                       df_company[['emp_id','boss_id', 'dept']],
                       on='emp_id')

         encode_education = {'education':{'High_School': 0, 'Bachelor': 1, 'Master': 2, 'PhD': 3}}
         encode_level = {'level': {'CEO': 0, 'Exec': 1, 'VP': 2, 'Director': 3, 'MM': 4, 'IC': 5}}

         df.replace(encode_education, inplace=True)
         df.head(20)
```

Out[15]:

	emp_id	signing_bonus	base_salary	education	gender	experience	boss_id	dept
0	138719	0	273000.0	2	M	2	43602.0	engineering
1	3192	0	301000.0	1	F	1	87847.0	sales
2	114657	0	261000.0	2	F	2	180854.0	sales
3	29039	0	86000.0	0	F	4	88370.0	HR
4	118607	0	126000.0	1	F	3	23565.0	sales
5	91334	0	221000.0	3	F	2	62990.0	sales
6	101901	1	194000.0	3	M	12	7558.0	engineering
7	171111	1	119000.0	2	F	1	30468.0	engineering
8	23443	1	217000.0	3	M	8	130585.0	sales
9	24195	0	230000.0	2	M	4	1050.0	engineering
10	184295	0	218000.0	3	M	5	18627.0	marketing
11	153551	0	232000.0	3	M	10	47755.0	sales
12	104709	0	320000.0	3	M	6	94683.0	marketing
13	175864	0	394000.0	2	M	5	110091.0	engineering
14	113738	1	204000.0	1	M	4	112643.0	sales
15	23737	0	76000.0	1	F	13	198893.0	HR
16	27476	0	358000.0	1	F	3	56451.0	sales
17	25181	0	171000.0	3	M	2	154381.0	sales

Fig. 2. Pre processing the data.

IV. Technical Discussion of Training and Testing of Data

For training and testing of employees data we decided to use random forest because it helps in training and testing better.

Random Forest:

- A random forest is a combination of decision trees
- Decision trees are build as a series of questions that narrows the possibility to make a confident decision
- The order of the questions can change and thus the final decision may change with it.
- A random forest is the collection of many decision trees by put together in a single model that is able to combine decision of all the decisions trees created

V. Results and Evaluation

After running our random forest model on the data, we get the mean absolute error of our predictions followed by a list of the different features and their importance score.

```
In [19]: # Use the forest's predict method on the test data
         predictions = rf.predict(test_feature)
         # Calculate the absolute errors
         errors = abs(predictions - test_label)
         # Print out the mean absolute error (mae)
         print('Mean Absolute Error:', round(np.mean(errors), 2), 'degrees.')

         Mean Absolute Error: 57531.89 degrees.

In [20]: # Get numerical feature importances
         importances = list(rf.feature_importances_)
         # List of tuples with variable and importance
         feature_importances = [(feature, round(importance, 2)) for feature, importance in zip(data_features, importances)]
         # Sort the feature importances by most important first
         feature_importances = sorted(feature_importances, key = lambda x: x[1], reverse = True)
         # Print out the feature and importances
         [print('Variable: {:20} Importance: {}'.format(*pair)) for pair in feature_importances];

         Variable: dept_HR              Importance: 0.58
         Variable: dept_engineering     Importance: 0.12
         Variable: experience           Importance: 0.1
         Variable: people_under         Importance: 0.07
         Variable: education            Importance: 0.05
         Variable: signing_bonus        Importance: 0.03
         Variable: level                Importance: 0.01
         Variable: gender_F             Importance: 0.01
         Variable: gender_M             Importance: 0.01
         Variable: dept_marketing       Importance: 0.01
         Variable: dept_sales           Importance: 0.01
         Variable: dept_CEO             Importance: 0.0

In [ ]:
```

Fig. 3. Random forest prediction results .

Our results showed that within our given set of employee data, there was not enough of an extreme variance in terms of feature important to warrant evidence for any discrimination or bias. At a glance, while the mean absolute error looks high, it is actually within the standard deviation of salaries in our given employee data. With that in mind, we can safely say that our prediction model was effective and that it helped support our conclusion. Additionally, when we look at our predicted salaries vs our test salaries (shown in Figure 4), we can see a similar spread pattern showing us that the feature importance of our predictions matches pretty closely to the actual feature importance.

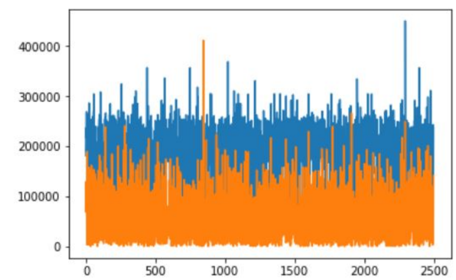

Fig. 4. Predicted vs. Test Data.

VI. Conclusion

A diverse workplace is a reflection of a changing world and marketplace. Diverse work team bring high value to the organization. Respecting individual differences will benefit the workplace by creating a competitive edge and increasing work productivity. Therefore treating the diverse workplace fairly will lead to high value of the company.

VII. references

- http://cs231n.github.io/python-numpy-tutorial/
- https://www.scipy.org/docs.html
- http://pandas.pydata.org/pandas-docs/version/0.15/tutorials.html
- https://networkx.github.io/documentation/stable/
- https://www.labri.fr/perso/nrougier/teaching/matplotlib/
- https://towardsdatascience.com/hands-on-introduction-to-scikit-learn-sklearn-f3df652ff8f2
- https://towardsdatascience.com/the-random-forest-algorithm-d457d499ffcd