International Journal of Computer Science and Information Technology
ISSN: 3005-9682 (Print), ISSN: 3005-7140 (Online) | Volume 2, Number 3, Year 2024
DOI: https://doi.org/10.62051/ijcsit.v2n3.07
Journal homepage: https://wepub.org/index.php/IJCSIT/index

Analysis and Implementation of University Teacher Salary Prediction Based on Decision Tree Regression Model

Shanwen Lei

Philippine Christian University, Manila 1004, Philippines

ABSTRACT

This study is based on the decision tree regression model, utilizing the Python programming language and third-party libraries such as Scikit-learn, to mine and analyze data of university teachers over the past five years. A decision tree regression model is constructed, enabling the prediction and analysis of university teacher salaries. This achievement provides university administrators with a more scientific, objective, and efficient decision-making reference, aiding in the construction of a more scientific and systematic budget management system for universities.

KEYWORDS

Decision Tree Regression Model, Python, Data Mining, Predictive Analysis

1. RESEARCH BACKGROUND

In the current rapid transformation of higher education, the role of university teachers is becoming increasingly important. They not only impart knowledge but also lead and inspire the development of academic fields. Therefore, university management needs to establish reasonable salary budgets to fully recognize the work of teachers and improve their motivation and work quality. At the same time, fair salary distribution helps retain a stable teaching staff, reduce talent turnover and recruitment costs, and improve the efficiency of university operations.

However, in formulating salary budgets for teachers, universities first need to understand and predict the level of teachers' salaries, and then derive specific budget amounts through statistics. However, predicting teacher salaries often faces various challenges, including the complexity of salary types and the diversity of teacher information. The accuracy of prediction may be affected by various factors, leading to bias and affecting the accuracy of the budget. Therefore, there is an urgent need for a scientific method to help universities accurately predict the salary levels of teachers.

2. APPLICATION ANALYSIS

In the past, when predicting salary levels manually, the following difficulties were often encountered:

- From a subjective perspective, due to the different ways in which different managers handle data and make decisions, differences in prediction results may occur, making it susceptible to personal bias and subjective judgments.
- From an objective perspective, managers have limited ability to process and analyze large amounts of data, making it difficult to comprehensively consider all factors affecting salaries, especially in situations where the data volume is relatively large and complex, making it even more difficult to discover deep relationships within the data.
- From a cost-benefit perspective, manual analysis and prediction of salaries require a significant

amount of time and effort, with low efficiency, and it is difficult to adapt quickly to market changes, resulting in a significant cost without significant benefits.

In today's information age, with the explosive growth of data and the continuous emergence of various effective data processing algorithms, people have begun to realize that traditional manual prediction methods are inadequate when facing large-scale and complex data. Therefore, more and more attention has turned to data mining technology, hoping to use its powerful data processing and analysis capabilities to solve various management and decision-making challenges.

Particularly in the field of salary level prediction, data mining technology has begun to receive widespread attention due to its objectivity, efficiency, and accuracy. Among them, the decision tree regression model, as a common and effective data mining algorithm, is based on a tree-like structure for decision-making, dividing the dataset into multiple subsets according to different features until certain conditions or purity thresholds are reached. During the prediction process, the decision tree model constructs a tree based on the existing data features and target values to represent the relationship between different feature values and target values, thereby predicting new data based on this.

Compared to the difficulties encountered previously, the decision tree regression model is able to effectively address and solve these challenges:

- The decision tree regression model predicts based on data and algorithms, without being influenced by subjective factors, which can improve the objectivity and consistency of the prediction results.
- The decision tree model can automatically process large amounts of data and discover deep relationships within the data by constructing a tree-like structure, thus more comprehensively considering factors that affect salaries.
- The decision tree regression model can conduct data analysis and prediction more quickly, thereby improving prediction efficiency and reducing cost inputs.

3. FUNCTION IMPLEMENTATION METHOD

Once the data mining algorithm to be used is determined, how can it be effectively applied to salary prediction? Python is the most suitable for data mining work! Python has a rich ecosystem of third-party libraries, such as Scikit-learn, which focuses on data mining applications. It provides users with numerous data mining algorithm APIs and analysis tools. With the help of these APIs and tools, the efficiency of data processing and model building has been greatly improved.

Next, taking China L University as a case study, we will analyze in detail how to implement the prediction of salary for university teachers at L University.

3.1. Data Preparation:

The foundation of data mining lies in having comprehensive and complete data, which is the key to ensuring the accuracy of mining results. Taking China L University as an example, it has more than 500 faculty and staff. Researchers collected information data of in-service teachers from 2019 to 2023. The data includes teachers' age, work experience, educational background, job category, job level, and salary. The data covers five consecutive years, providing temporal continuity for the study.

Next, researchers conducted data cleaning and preprocessing to eliminate potential data quality issues and ensure the accuracy and consistency of the data. After preprocessing, a total of 2185 valid data points were obtained. Each data point contains 7 features such as 'AGE', 'DEPARTMENT', 'WORKING EXPERIENCE', 'EDUCATIONAL EXPERIENCE', 'WORKING CATEGORY', 'WORKING POSITION', and 'PROFESSIONAL TITLE', with a total of 15295 feature points.

'SALARY' is used as the target variable in the analysis and is used together with the previous 7 features to build the model.

Some of the features in the data, such as 'DEPARTMENT', 'EDUCATIONAL EXPERIENCE', 'WORKING CATEGORY', 'WORKING POSITION', and 'PROFESSIONAL TITLE', are text data. Directly using these text data for regression analysis may cause issues because text data are typically categorical variables and cannot be directly processed numerically. Therefore, researchers need to first transform these text data into numerical form for use in regression model analysis.

NO.	ID NO.	AGE	DEPARTMENT	WORKING EXPERIENCE	EDUCATIONAL EXPERIENCE	WORKING CATEGORY	WORKING POSITION	PROFESSIONAL TITLE	SALARY
1	803149	47	5	25	3	1	13	5	243168.4
2	803142	44	5	19	3	1	12	5	200547.8
3	803301	30	5	5	4	1	9	4	195185.03
4	803277	32	5	8	4	1	9	4	183301
5	803271	34	5	9	4	1	8	3	160207.64
6	803270	35	5	8	4	1	9	4	205632.56
7	803232	50	5	30	4	1	12	5	257645.54
8	803230	36	5	13	4	1	10	4	185559.63
9	803162	49	5	24	3	1	11	4	236407.43
10	803161	55	5	36	4	1	13	5	236889.29
11	803154	54	5	29	3	1	14	5	253362.69
12	805049	33	5	6	4	1	7	3	110459
13	803378	29	5	5	4	2	7	3	169939.87
14	803373	31	5	3	4	2	7	3	174055.9
15	803361	35	5	9	4	2	7	3	184961.24
16	802024	59	1	40	3	4	14	5	281169.23
17	803371	36	5	11	4	1	7	3	167028.65
18	803370	27	5	3	4	1	7	3	157463.65
19	803131	45	5	21	3	1	11	4	210770.19
20	803116	47	5	25	3	1	12	5	253217.57
21	803115	46	5	24	3	1	12	5	252946.45
22	802011	59	5	39	3	5	16	6	361367.85
23	905025	45	5	22	3	1	9	4	202362.71
24	803310	42	5	17	3	1	9	4	171601.64
25	803303	33	5	9	4	2	9	4	179730.44
26	803342	30	5	6	4	1	7	3	174076.61
27	803341	34	5	9	4	1	7	3	162352
28	803381	33	5	7	4	1	7	3	158423.65
29	803385	26	5	2	4	1	7	3	108945.44
30	805041	42	4	23	3	2	19	1	205687.28
31	803390	29	4	6	4	1	7	3	134636.8
32	803391	35	4	8	4	1	7	3	125873.2
33	803072	59	7	39	3	5	15	6	315641.36
34	905066	33	4	9	4	2	7	3	154090.03
35	803359	27	4	5	3	1	7	3	151676.8
36	803360	27	4	2	4	2	7	3	165060.42

Figure 1 Partial Display of Data Set after Data Assignment

3.2. Construction of Decision Tree Regression Model

After data preprocessing, we can proceed to construct the decision tree regression model. Here, we will directly call the decision tree regression algorithm API from the Scikit-learn library and use Python language for data mining.

```
def load_data_and_split(file_path, features, target, test_size=0.2, random_state=42):
    all_data = pd.read_excel(file_path)

    X = all_data[features]
     y = all_data[target]

    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=test_size, random_state=random_state)

    train_features_path = file_path.replace('.xlsx', '_train_features.xlsx')
    test_features_path = file_path.replace('.xlsx', '_test_features.xlsx')
    train_target_path = file_path.replace('.xlsx', '_train_target.xlsx')
    test_target_path = file_path.replace('.xlsx', '_test_target.xlsx')

    X_train.to_excel(train_features_path, index=False)
    X_test.to_excel(test_features_path, index=False)
    y_train.to_excel(train_target_path, index=False)
    y_test.to_excel(test_target_path, index=False)

    root = tk.Tk()
    root.title("Dataset Size")

    label_train_features = tk.Label(root, text=f"Train Features Size: {X_train.shape}")
    label_train_features.pack()

    label_train_target = tk.Label(root, text=f"Train Target Size: {y_train.shape}")
    label_train_target.pack()

    label_test_features = tk.Label(root, text=f"Test Features Size: {X_test.shape}")
    label_test_features.pack()

    label_test_target = tk.Label(root, text=f"Test Target Size: {y_test.shape}")
    label_test_target.pack()

    root.mainloop()

    return X_train, X_test, y_train, y_test

file_path = '/Volumes/Temporary/学习/project1/test.xlsx'
features = ['AGE', 'DEPARTMENT', 'WORKING EXPERIENCE', 'EDUCATIONAL EXPERIENCE', 'WORKING CATEGORY',
'WORKING POSITION', 'PROFESSIONAL TITLE']
target = 'SALARY'
X_train, X_test, y_train, y_test = load_data_and_split(file_path, features, target)
```

Figure 2 Dataset Splitting Code

Through the above code, we first split the dataset into training and testing sets in a ratio of 80% to 20%. Additionally, the code exports and saves the split training and testing sets separately for convenience, enabling repeated testing using the same sets.

Following the code split, we obtain 1748 data entries for the training set and 437 data entries for the testing set.

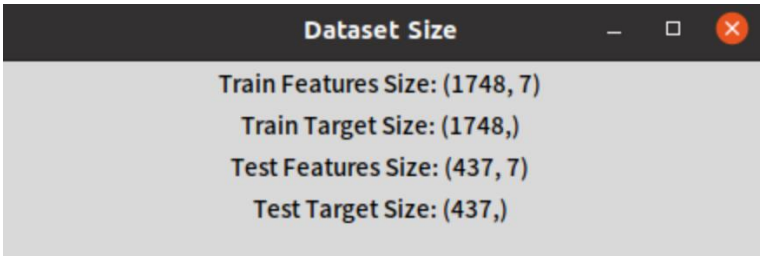

Figure 3 Data Set Split Result Display

Next, we applied decision tree regression to analyze the dataset. The training set was used to build the decision tree regression model, while the testing set was employed to evaluate the predictive performance of the generated decision tree model.

```
decision_tree_reg = DecisionTreeRegressor()


decision_tree_reg.fit(X_train, y_train)


visualizer = PredictionError(decision_tree_reg)
visualizer.fit(X_train, y_train)
visualizer.score(X_test, y_test)
g = visualizer.poof()


prd_decision_tree_reg = decision_tree_reg.predict(X_test)
plt.figure(figsize=(10, 8))
x = range(len(prd_decision_tree_reg))
plt.grid(ls=':', lw=1)
plt.scatter(x, prd_decision_tree_reg, color='#FEB64D', marker='o', facecolor='None', lw=2)
plt.scatter(x, y_test, color='#9287E7', marker='o')
label = ["Prediction", "Actual"]
plt.legend(label, loc=2, markerscale=0.85, ncol=1, fontsize=10, framealpha=1)
plt.title('Decision Tree Regression', fontsize=12)
plt.savefig("/Volumes/Temporary/学习/project1/decision_tree_regression.png")
plt.show()
print("Decision Tree Regressor Model Score: ", decision_tree_reg.score(X_test, y_test))
```

Figure 4 Decision Tree Regression Model Construction Code

This is a plot illustrating the distribution of prediction errors obtained from testing the model with the test dataset. On the x-axis are the actual values of the target variable from the test set, while the y-axis represents the predicted values of the target variable. The closer the fitted line is to the diagonal, the more accurate the predictions. Here, the prediction error is measured using the R2 testing method, where a score closer to 1 indicates better testing performance.

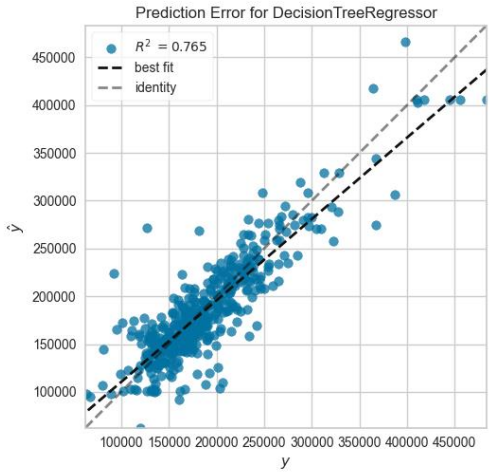

Figure 5 Prediction Error for Decision Tree Regression

This graph displays the distribution of predicted values and the original values from the test data when tested on the model.

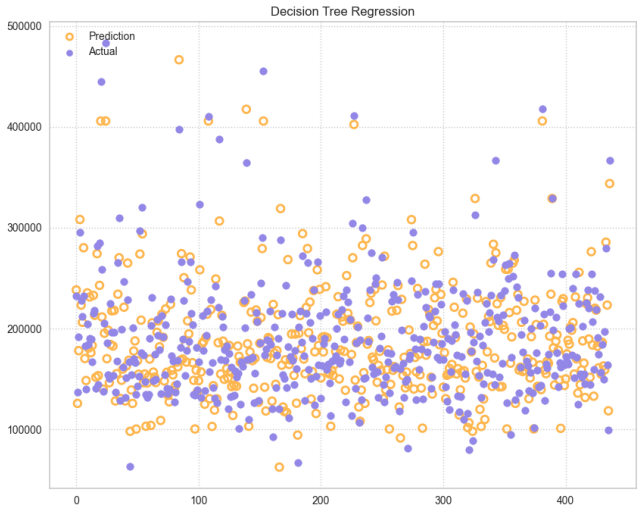

Figure 6 Display of Decision Tree Regression Test Results

After testing with the test data, it was evident that the decision tree regression model constructed using the dataset exhibited a good predictive performance. The fitted line of the predicted data was very close to the line of the standard data, and the R2 test score was also relatively high.

Finally, we used Python code to visualize the tree structure of the decision tree regression model.

```
decision_tree_reg = DecisionTreeRegressor()


decision_tree_reg.fit(X_train, y_train)


tree_depth = decision_tree_reg.get_depth()


plt.figure(figsize=(50, 30))
plot_tree(decision_tree_reg, filled=True, feature_names=features, rounded=True, fontsize=8)
plt.title(f'Decision Tree (Depth: {tree_depth})', fontsize=16)
plt.savefig("/Volumes/Temporary/学习/project1/decision_tree.png")
plt.show()
```

Figure 7 Visualized Decision Tree Regression Model Code

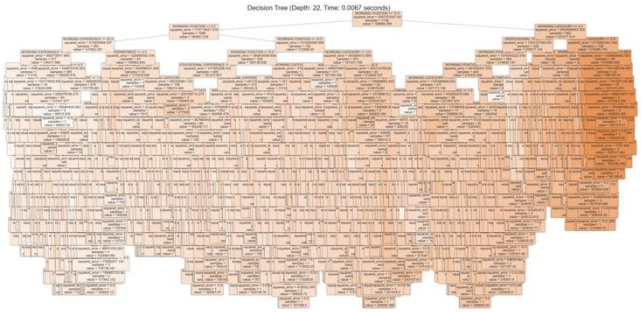

Figure 8 Decision Tree Regression Model Diagram

3.3. Salary Prediction Application

After constructing the decision tree regression model, we will proceed to test its actual predictive performance.

Firstly, researchers separated the data from China L University's dataset for the year 2023, summarizing and aggregating the actual target variables for 2023. Then, they inputted the features of the university teachers for the year 2023 into the decision tree regression model to generate corresponding predicted values. Subsequently, all predicted values were aggregated, and the final calculation involved determining the difference between the actual and predicted values, as well as the percentage deviation in prediction.

Here, we also exported the successfully constructed decision tree regression model as a pkl file using Python code, facilitating direct model invocation during testing. Additionally, researchers added code for a simple GUI display based on tkinter for the prediction program, thereby significantly enhancing its usability and display clarity.

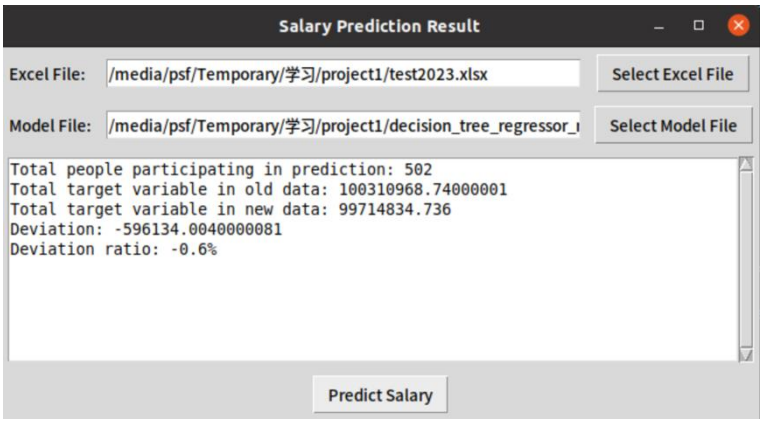

Figure 9 Salary Prediction Program With GUI

As shown in the above figure, after testing and calculation by the model, the total salary expenditure for 502 university teachers in 2023, as per the original table, amounted to over 100,310,968 yuan. The predicted total salary for university teachers was over 99,714,834 yuan, resulting in a difference of only about 596,134 yuan, with a prediction deviation of merely -0.6%.

4. CONCLUSION

This study, based on the decision tree regression model and supported by Python programming language and third-party libraries like Scikit-learn, has achieved predictive analysis of university teacher salaries. Through testing with actual data, the prediction deviation was only -0.6%, meeting the researchers' expectations for model accuracy. This fully demonstrates the good practical effect of the decision tree regression model in predicting university teacher salaries. Leveraging the predictive results of the decision tree regression model can provide university managers with scientific, objective, and efficient decision-making references, thereby assisting universities in constructing a more scientific and systematic budget management system.

REFERENCES:

[1] Chen, J., Mao, S., & Yuan, Q. (2022, March). Salary prediction using random forest with fundamental features. In Third International Conference on Electronics and Communication; Network and Computer Technology (ECNCT 2021) (Vol. 12167, pp. 491-498). SPIE.

[2] Kushwah, J. S., Kumar, A., Patel, S., Soni, R., Gawande, A., & Gupta, S. (2022). Comparative study of regressor and classifier with decision tree using modern tools. Materials Today: Proceedings, 56, 3571-3576.

[3] Eichinger, F., & Mayer, M. (2022). Predicting salaries with random-forest regression. In Machine Learning and Data Analytics for Solving Business Problems: Methods, Applications, and Case Studies (pp. 1-21). Cham: Springer International Publishing.

[4] Asaduzzaman, A., Uddin, M. R., Woldeyes, Y., & Sibai, F. N. (2024, January). A Novel Salary Prediction System Using Machine Learning Techniques. In 2024 Joint International Conference on Digital Arts, Media and Technology with ECTI Northern Section Conference on Electrical, Electronics, Computer and Telecommunications Engineering (ECTI DAMT & NCON) (pp. 38-43). IEEE.

[5] Alao, D. A. B. A., & Adeyemo, A. B. (2013). Analyzing employee attrition using decision tree algorithms. Computing, Information Systems, Development Informatics and Allied Research Journal, 4(1), 17-28.

[6] El-Rayes, N., Fang, M., Smith, M., & Taylor, S. M. (2020). Predicting employee attrition using tree-based models. International Journal of Organizational Analysis, 28(6), 1273-1291.