Salary Prediction Based on 1994 US Census Data

Reem Alessa
dept. Computer Sciences
King Saud University
Riyadh, Saudi Arabia
441200983@student.ksu.edu.sa

Mohrah Aljafar
dept. Computer Sciences
King Saud University
Riyadh, Saudi Arabia
441201214@student.ksu.edu.sa

Abeer Alshuaibi
dept. Computer Sciences
King Saud University
Riyadh, Saudi Arabia
439200693@student.ksu.edu.sa

***Abstract*— Salary prediction is the problem of training a model on historical salary information to estimate people's future salaries. The purpose of the research paper is to examine the features that influence a person's income using the dataset retrieved by Barry Becker from the US 1994 Census database. In this work, Weka software is used to build classification models for Salary prediction using the Decision Tree J48 and Nave Bayesian. The results of this paper reveal that the Decision tree J-48 algorithm outperformed the Nave Bayesian algorithm in terms of accuracy.**

***Keywords*—**Data Mining, Naïve Bayes, Decision Tree, Salary Prediction, J-48, Weka.

I. INTRODUCTION

Data mining refers to the process of uncovering patterns and trends in datasets that are difficult to spot using standard statistical techniques. The aim is to discover new insights and knowledge that can be used to develop predictive models [1]. This paper uses various data mining techniques to address the problem of salary prediction.

Salary prediction is the problem of training a model on historical salary data to predict future salaries for individuals. A person's income can be influenced by numerous factors, including age, education, race, and sex. Understanding the tendencies and patterns that determine a person's wage is crucial, particularly given the potential for biases such as race and gender influencing a person's compensation. In this paper, we will employ classification techniques to address this problem and provide insights into the key factors that influence a person's salary.

II. LITERATURE REVIEW

In this section, we will review recent research on salary prediction models and summarize how different prediction and classification techniques were used to classify or predict a person's salary.

A. "Statistical Machine Learning Regression Models for Salary Prediction Featuring Economy Wide Activities and Occupations"

This paper [2] discusses the development of a framework for predicting labor salaries in the Saudi Arabian economy using statistical machine learning algorithms. Their proposed framework enables fine-grained salary prediction with limited survey data by considering both occupational features and organizational characteristics. They evaluated the performance of five different supervised machine learning algorithms in predicting mean annual salaries across economic activities and major occupational groups. Their results show that non-linear machine learning algorithms such as Bayesian Gaussian process regression and artificial neural networks outperform traditional linear models such as multiple linear regression. The authors suggest that the use of machine learning algorithms can reduce the cost of salary benchmarking and improve accuracy, particularly when estimating salaries for similar occupations in different industries or different occupations within the same sector.

B. "Salary Prediction Using Machine Learning"

This paper [3] proposes a salary prediction model using a linear regression algorithm with second-order polynomial transformation. The goal is to predict the salary of an employee based on various parameters such as job type, degree, major, years of experience, industry, and miles from a metropolitan area. Their methodology involved data collection, cleaning, manual feature engineering, automatic feature selection, model selection, training, and validation, and model comparison. The authors found the most relevant five features for salary prediction and compared their model's performance with other algorithms based on standard scores and curves like the F1 score, ROC curve, and Precision-Recall curve. The authors also explored how to add more attributes to the basic model and determined the most appropriate method for doing so, further increasing the accuracy of the model. They found that their model achieved a 76% accuracy rate, which is relatively high compared to other models.

C. "Salary Prediction Using Regression Techniques"

This paper [4] presents an interesting approach to predicting salary growth and creating a user-friendly graphical representation of the data. The system takes the salary data from an organization's database and creates a graph that shows the salary growth for a particular position based on qualifications and experience. The graph is created using linear and polynomial regression models. Their findings reveal that the model's prediction is true up to a certain percentage, and they conclude that k-nearest regression can increase prediction accuracy.

D. "Salary Prediction in the IT Job Market with Few High-Dimensional Samples: A Spanish Case Study"

This research [5] focuses on predicting salaries offered by companies through job advertisements on a Spanish e-recruitment website that specializes in IT positions for young people. The work is difficult due to the small number of samples, high dimensionality, and presence of noise. The study examines major areas of the job market and identifies factors that influence final salary, such as experience, job stability, and certain job roles like Team Leader and IT Architect. Several models, including logistic regression, closest neighbors, MLPs, SVMs, random forests, adaptive boosting, and voting classifiers, are compared in the study. Ensembles based on decision trees outperform and attain an accuracy of around 84.

III. DESCRIPTION OF THE DATASET

A. Data Set Information

Barry Becker extracted the data from the 1994 Census database. The 1994 US Census database has 48,842 records that make up the US Adult Census dataset [6]. The dataset provides 14 input variables that are a mixture of categorical, ordinal, and numerical data types, The US Adult Census dataset is a widely used dataset in the field of machine learning for classification tasks. The dataset has been used to evaluate the performance of various machine learning algorithms and to compare the accuracy of different classification models. The dataset is challenging due to its mixed data types, missing values, and class imbalance. Therefore, pre-processing techniques such as data cleaning, feature selection, and handling missing values are essential to obtain reliable and accurate results.

B. Objective

The primary goal of the dataset is to categorize individuals earning less than $50,000 or more than $50,000 based on a variety of explanatory variables, such as age, occupation, education, and others. Table 1 lists the attributes in the dataset along with their descriptions.

Table 1 Attributes Description

Attribute	Description	Type	Possible values	Min	Max
Age	The age of an individual	Numeric-Discrete	-	17	90
Workclass	A general term to represent the employment status of an individual.	Categoric - Nominal	Private, Self-emp-not-inc, Self-emp-inc, Federal-gov, Local-gov, State-gov, Without-pay, Never-worked		
Fnlwgt	Final weight. In other words, this is the number of people the census believes the entry represents	Numeric	-	77516	1484705
Education	The highest level of education achieved by an individual	Categoric - Ordinal	Bachelors, Some-college, 11th, HS-grad, Prof-school, Assoc-acdm, Assoc-voc, 9th, 7th-8th, 12th, Masters, 1st-4th, 10th, Doctorate, 5th-6th, Preschool		
Education-num	The highest level of education achieved in numerical form	Numeric-Discrete	-	9	16
Marital-status	Marital status of an individual	Categoric - Nominal	Married-civ-spouse, Divorced, Never-married, Separated, Widowed, Married-spouse-absent, Married-AF-spouse		
Occupation	The general type of occupation of an individual	Categoric - Nominal	Tech-support, Craft-repair, Other-service, Sales, Exec-managerial, Prof-specialty, Handlers-cleaners, Machine-op-inspct, Adm-clerical, Farming-fishing, Transport-moving, Priv-house-serv, Protective-serv, Armed-Forces		
Relationship	Represents what this individual is relative to others	Categoric - Nominal	Wife, Own-child, Husband, Not-in-family, Other-relative, Unmarried		
Race	Descriptions of an individual's race	Categoric -Nominal	White, Asian-Pac-Islander, Amer-Indian-Eskimo, Other, Black		
Sex	The sex of the individual	Categoric - Binary	Female, Male		
Capital-gain	Capital gains for an individual	Numeric-Discrete	-	0	99999
Capital-loss	Capital loss for an individual	Numeric-Discrete	-	0	4356
Hours-per-week	The hours an individual has reported to work per week	Numeric-Discrete	-	1	40
Native-country	Country of origin for an individual	Categoric - Nominal	United-States, Cambodia, England, Puerto-Rico, Canada, Germany, Outlying-US(Guam-USVI-etc), India, Japan, Greece, South, China, Cuba, Iran, Honduras, Philippines, Italy, Poland, Jamaica, Vietnam, Mexico, Portugal, Ireland, France, Dominican-Republic, Laos, Ecuador, Taiwan, Haiti, Columbia, Hungary, Guatemala, Nicaragua, Scotland, Thailand, Yugoslavia, El-Salvador, Trinadad&Tobago, Peru, Hong, Holand-Netherlands		
Salary	Whether or not an individual makes more than $50,000 annually	Categoric - Binary	<=50K , >50K		

IV. TECHNIQUE

This research will employ the data mining classification technique to categorize individuals based on their wages. Classification is a type of supervised machine learning that utilizes various methods to train a model on labeled data, enabling it to classify new input data into one of the labels. In this paper, the labels are based on the "Salary" attribute, which has two values: "<=50K" or ">50K." The algorithms employed in this research will classify people as one of these two labels based on the various attributes in the salary prediction dataset.

A. Decision Tree - J48

Decision Tree - J48 is a classification technique that uses a tree-like structure to split the data into subsets based on the

values of the features. J48 is an open-source Java implementation of the C4.5 algorithm in the Weka data mining tool. It can handle both categorical and numerical attributes and can deal with missing values and prune the tree to avoid overfitting [7].

B. Naïve Bayesian

Naïve Bayesian is a classification technique that uses Bayes' theorem and the assumption of conditional independence of features. It can be trained very efficiently and often outperforms more complex methods. It can be represented by a Bayesian network with a single root and no edges between the other nodes. It calculates the posterior probability of a class given the input by learning the prior probability of the class and the conditional probability of the features [8].

V. Relationship Between Attributes

To find the relationships between the attributes. We used Weka's "CorrelationAttributeEval" filter in the "Select Attributes" tab with the "Ranker" search method. The highest positive correlation was spotted between the marital-status attribute and the relationship attribute with a result equal to 0.4545, while the highest negative correlation spotted is equal to -0.076646 between the fnlwgt attribute and the age attribute.

VI. Preprocessing

A. Missing values

From the Preprocess tab in Weka, we applied the Unsupervised filter called RemoveWithValues using the Instance filter option. This filter allows us to remove instances (rows) that contain specific values. In our case, we used this filter to remove rows that contained missing values, which were represented by the symbol "?" in Weka. After applying the filter, we were able to identify and remove the rows that contained the missing values. In total, we deleted 2,399 rows that contained the missing values.

This pre-processing step is important to ensure that our model is not affected by the presence of missing values in the data. Missing values can significantly impact the performance of our models by introducing bias, reducing the accuracy of our predictions, and leading to incorrect conclusions.

B. Outliers

We used the InterquartileRange filter to detect the outliers with the outlierFactor set to 1.5. After identifying all the outliers, we removed these rows using the RemoveWithValues filter. In total, we deleted 8246 rows that contained the outlier values.

Removing outliers is important to ensure that our model is not affected by the presence of outliers in the data. Outliers can significantly impact the performance of our model by skewing the data distribution, increasing the variance, and reducing the accuracy of our predictions.

C. Feature Selection

After using the CorrelationAttributeEval AttributeEvaluator with the Ranker method and the full training data set, we found that the 'Fnlwgt' attribute had the last rank with a 0.0000443 value. As a result, we decided to remove this attribute from the dataset. This decision was because the 'Fnlwgt' attribute had the least correlation with the class attribute 'salary'. By removing this attribute, we were able to simplify our dataset and improve performance.

VII. Experiments

Both the Decision Tree - J48 and Naïve Bayesian classifiers were trained on 75% of the dataset and tested on the remaining 25%.

A. Naïve Bayesian classifier

The Naïve Bayesian model took 0.05 seconds to build and 0.3 seconds to test; 4433 instances were correctly classified, representing 80.9098% of the dataset, while 1046 instances were incorrectly classified, representing 19.0911%. See Table 2, 3, 4, and 5.

Table 2 Naïve Bayesian Model Build & Test Time

Metric	Time (seconds)
Build model	0.05
Test model	0.03

Table 3 Naïve Bayesian Classifier Summary

Metric	Value
Correctly Classified Instances	4433 (80.91%)
Incorrectly Classified Instances	1046 (19.09%)
Kappa Statistic	0.4554
Mean Absolute Error	0.1968
Root Mean Squared Error	0.4004
Relative Absolute Error	50.74%
Root Relative Squared Error	97.78%
Total Instances	5479

Table 4 Naïve Bayesian Classifier Detailed Accuracy by Class

Metric	<=50K	>50K	Weighted Avg
TP Rate	0.926	0.484	0.809
FP Rate	0.516	0.074	0.399
Precision	0.833	0.701	0.797
Recall	0.926	0.484	0.809
F-Measure	0.877	0.573	0.797
MCC	0.468	0.468	0.468
ROC Area	0.877	0.877	0.877
PRC Area	0.954	0.714	0.890

Table 5 Naïve Bayesian Classifier Confusion Matrix

Actual \Predicted	<=50K	>50K
<=50K	3731	299
>50K	747	702

B. Decision Tree - J48 classifier

Building the J48 model took 0.27 seconds, while testing the model took only 0.01 seconds. The size of the decision tree is 484 with 372 leaves, and the correctly classified instances were 4670, representing 85.2345% of the dataset, while the incorrectly classified instances were 809, representing the remaining 14.7655%. See Table 6, 7, 8, and 9.

Table 6 J48 Model Build & Test Time

Metric	Time (seconds)
Build model	0.27
Test model	0.01

Table 7 J48 Classifier Summary

Metric	Value
Correctly Classified Instances	4670 (85.23%)
Incorrectly Classified Instances	809 (14.77%)
Kappa statistic	0.6026
Mean absolute error	0.2102
Root mean squared error	0.3334
Relative absolute error	54.201%
Root relative squared error	75.5967%
Total Number of Instances	5479

Table 8 J48 Classifier Detailed Accuracy by Class

Metric	<=50K	>50K	Weighted Avg.
TP Rate	0.925	0.651	0.852
FP Rate	0.349	0.075	0.277
Precision	0.880	0.757	0.848
Recall	0.925	0.651	0.852
F-Measure	0.902	0.700	0.849
MCC	0.606	0.606	0.606
ROC Area	0.878	0.878	0.878
PRC Area	0.935	0.747	0.885

Table 9 J48 Classifier Confusion Matrix

Actual \Predicted	<=50K	>50K
<=50K	3727	3727
>50K	506	506

VII. EVALUATION

To evaluate the two models, we compared their accuracies, error rates, true-positive rates, false-positive rates, precisions, recall, ROC areas, mean absolute errors, and root mean squared errors, as shown in Table II. Although the J48 took more time to build, it gave better overall results than the Naïve Bayesian model.

Table 10 Results

Algorithm/evaluation metric (weighted avg.)	Naïve Bayesian	Decision Tree - J48
Time to build the model (in seconds)	0.05	0.27
Time to test the model (in seconds)	0.03	0.01
Accuracy	80.9089%	85.2345%
Error Rate	19.0911%	14.7655%
TP Rate	80.9%	85.2%
FP Rate	39.9%	27.7%
Precision	79.8%	84.8%
Recall	80.9%	85.2%
ROC Area	0.877	0.878
MAE	0.1968	0.2102
RMSE	0.4004	0.3334

VIII. CONCLUSION

In conclusion, using the adult data set with Weka and applying the J48 and Naive Bayes algorithms have shown promising results. After analyzing the performance of both algorithms, it can be concluded that J48 provides more accurate results than Naive Bayesian. J48's accuracy can be attributed to its ability to handle both categorical and numerical data, as well as its ability to handle missing data. Naive Bayesian, on the other hand, assumes that all attributes are independent of each other, which may not always hold true in real-world scenarios. Therefore, when working with the adult data set, J48 can be considered a more suitable algorithm for predictive modeling.

REFERENCES

[1] Calders, T. and Custers, B. (2013) 'What is data mining and how does it work?', *Studies in Applied Philosophy, Epistemology and Rational Ethics*, pp. 27–42. doi:10.1007/978-3-642-30487-3_2.

[2] Matbouli, Y.T. and Alghamdi, S.M. (2022) 'Statistical machine learning regression models for salary prediction featuring economy wide activities and occupations', *Information*, 13(10), p. 495. doi:10.3390/info13100495.

[3] Lothe, P. D., Tiwari, P., Patil, N., Patil, S., and Patil, V. (2021). 'Salary Prediction using Machine Learning', *International Journal of Advance Sciencetic Research and Engineering Trends*, 6(5), 199-202.

[4] Das, S., Barik, R. and Mukherjee, A. (2020) 'Salary prediction using regression techniques', *SSRN Electronic Journal*. doi:10.2139/ssrn.3526707.

[5] Martín, I. et al. (2018) 'Salary prediction in the IT job market with few high-dimensional samples: A Spanish case study', *International Journal of Computational Intelligence Systems*, 11(1), p. 1192. doi:10.2991/ijcis.11.1.90.

[6] UCI Machine Learning Repository: Census Income Data Set, https://archive.ics.uci.edu/ml/datasets/Census+Income (accessed May 29, 2023).

[7] Chandrasekar, P. *et al.* (2017) 'Improving the prediction accuracy of decision tree mining with data preprocessing', *2017 IEEE 41st Annual Computer Software and Applications Conference (COMPSAC)*. doi:10.1109/compsac.2017.146.

[8] Yang, F.-J. (2018) 'An implementation of naive bayes classifier', *2018 International Conference on Computational Science and Computational Intelligence (CSCI)*. doi:10.1109/csci46756.2018.00065.