ALY6020: Predictive Analytics

Module 1 Project – Understanding Income Inequality

Smit Pareshbhai Ranpariya
Student ID: 002846259
Master of Professional Studies in Analytics

Northeastern University
Toronto

Dr. Harpreet Sharma
November 06, 2024

Predicting Income Classification Using Nearest Neighbors

Introduction

This report explores a dataset of U.S. citizens' census attributes to predict whether individuals earn low income or high income. The dataset includes attributes such as age, education level, marital status, occupation, gender, and race. This model can assist in understanding income disparities and contribute to organizations aiming to create fairer income policies. The goal is to use KNN algorithm to predict the income class based on these features, and to interpret the importance of each feature in determining income levels.

Part 1: Data Cleansing Process

Data quality is essential for building a reliable predictive model. To ensure that the dataset was ready for analysis, several data cleansing techniques were employed. Below are the steps taken:

Removing Irrelevant Columns:

The column inlwgt (individual weight) was dropped as it was irrelevant for the prediction task.

Handling Missing Values:

The dataset contained some missing values, represented by the character '?'. These values were replaced with NaN (Not a Number), making it easier to identify and handle them.

Missing values for categorical columns were imputed with the most frequent values in each column. This approach was chosen as it would fill in missing data without introducing bias or overfitting.

For this, a SimpleImputer was used with the strategy set to 'most_frequent' to replace missing categorical data with the most common values in each column.

Data Transformation:

The categorical columns (such as workclass, education, occupation, etc.) were encoded numerically using Label Encoding, which converts categories into numerical values. This was necessary since most machine learning algorithms, including KNN, require numerical data.

The target variable Salary was also encoded into binary values, where <=50K was mapped to 0 (low income) and >50K was mapped to 1 (high income).

Feature Scaling:

Numerical features such as age, education-num, capital-gain, capital-loss, and hours-per-week were standardized using the StandardScaler. This scaling process ensured that all numerical features contributed equally to the model, preventing features with larger ranges from dominating the predictions.

Final Data Quality Check:

After data transformation and imputation, a final check for any remaining missing values was performed. The dataset was found to have no missing values after the imputation process.

Part 2: Building the Nearest Neighbors Model

The KNN algorithm was chosen for this task because it is a simple yet effective machine learning method for classification. KNN works by finding the 'k' nearest data points to a given test point and using their labels to predict the label of the test point.

Splitting the Data:

The dataset was divided into two sets: the training set (70% of the data) and the test set (30%). This split was done to train the model on one portion of the data and test its accuracy on an unseen portion.

Choosing the Optimal Value of K:

The KNN model requires the selection of a hyperparameter k, which represents the number of neighbors to consider for classification. To identify the best value of k, a range of values from 1 to 19 (odd values) was tested. For each value of k, the accuracy of the model was evaluated on the test set.

The model performed best with a value of k = 19, achieving an accuracy of approximately 84%. This was the optimal choice as it struck a balance between model complexity and accuracy.

Model Training and Evaluation:

The final model was trained using k = 19, and the classification report was generated, which provided insights into the model's precision, recall, and F1-score for both income classes (<=50K and >50K).

The model performed well in predicting low-income individuals (<=50K), with a precision of 0.87 and a recall of 0.93, indicating that it accurately classified most low-income individuals. However, the model had a lower performance for high-income individuals (>50K), with a precision of 0.71 and a recall of 0.56. This imbalance may be attributed to the relatively smaller proportion of high-income individuals in the dataset.

Feature Importance:

The model's prediction is influenced by various features. Some of the most important features include age, education-num, capital-gain, and hours-per-week. These features have strong correlations with income levels. For example:

Age: Older individuals tend to have more experience and thus may earn higher salaries.

Education level: Higher educational attainment is correlated with higher income.

Capital-gain: Individuals with higher capital gains often have higher income.

Hours-per-week: Those working longer hours per week are more likely to earn higher incomes.

Model Accuracy:

- The overall accuracy of the model was found to be 84%. While this is a solid result, there is room for improvement, particularly in predicting the high-income class. To address this, more advanced techniques or model tuning could be applied.

Visualizing Model Performance:

- A plot showing the relationship between the value of k and model accuracy was created. This helped visualize how the accuracy changed as k increased. The plot indicated that the optimal value of k was 19.

Conclusion

The KNN model performed reasonably well in predicting income class, with an overall accuracy of 84%. The features that most significantly impacted income classification were age, education-num, capital-gain, and hours-per-week. Although the model accurately predicted low-income individuals, its performance for high-income individuals was less robust. The optimal choice for k was 19, based on the model's accuracy. Further refinement of the model, such as using other classification techniques or fine-tuning hyperparameters, could improve its performance.

Reference

GeeksforGeeks. *(KNN) algorithm*. https://www.geeksforgeeks.org/k-nearest-neighbours/

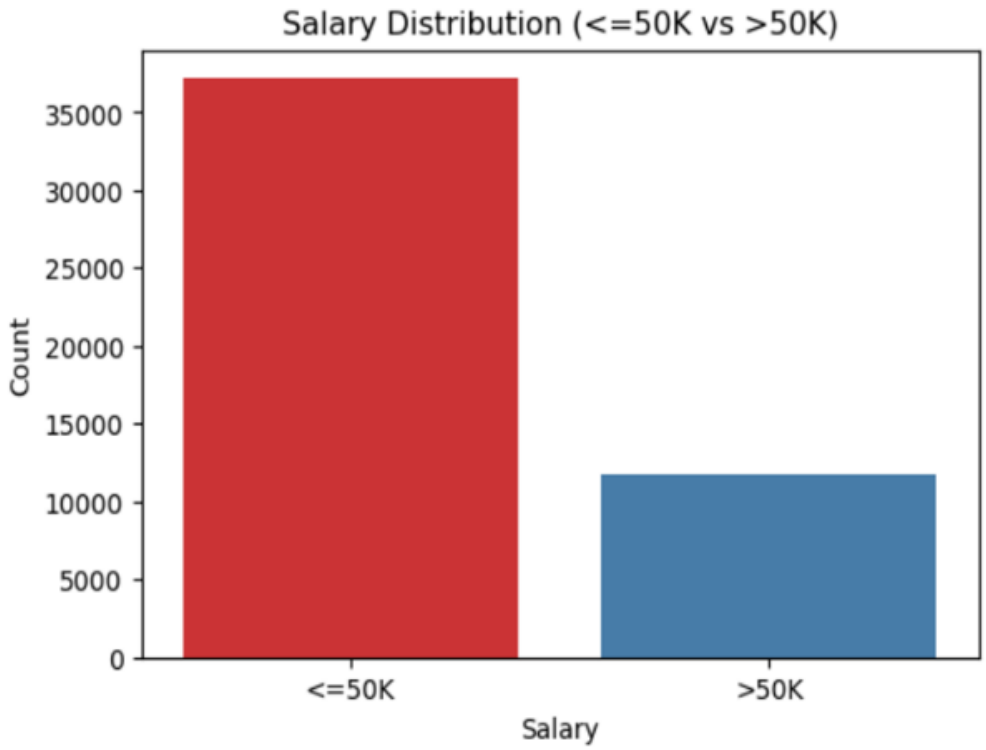
IBM *(KNN)*. https://www.ibm.com/topics/knn

Java point *K-nearest neighbor algorithm for machine learning* https://www.javatpoint.com/k-nearest-neighbor-algorithm-for-machine-learning

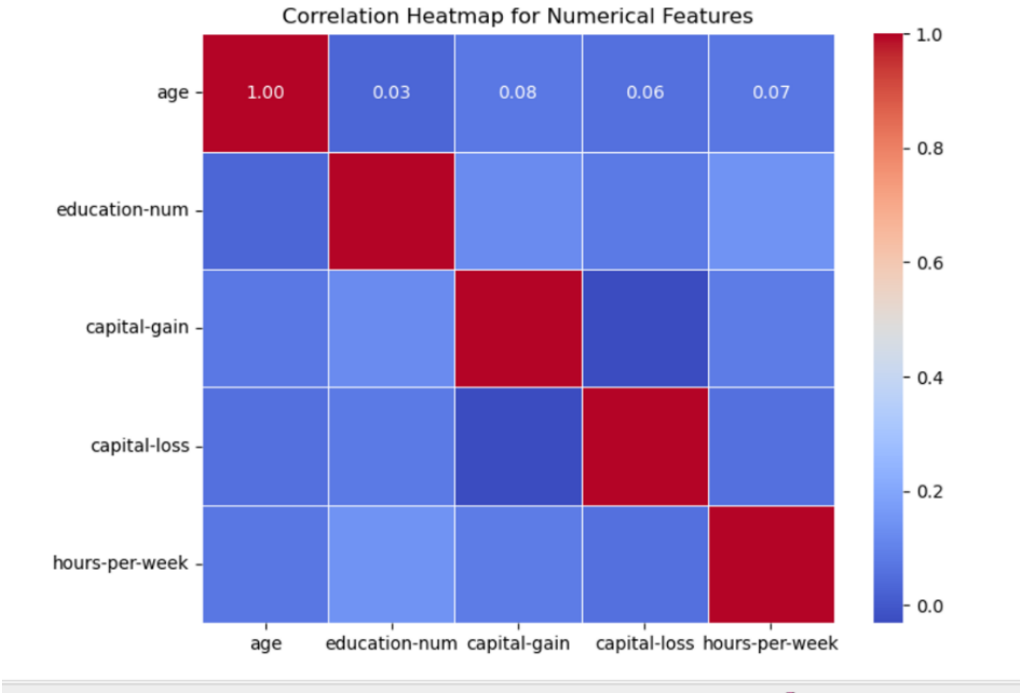
Appendix

Figures:

1. **Salary Distribution Plot**
    - Displays the count of low-income (<=50K) and high-income (>50K) individuals in the dataset.



2. **Correlation Heatmap**
    - Shows the correlations between numerical features in the dataset.



3. **Accuracy vs. K Value Plot**
    - Visualizes the relationship between different k values and the corresponding model accuracy.

```
Best k value: 19
Classification Report:
              precision    recall  f1-score   support

           0       0.87      0.93      0.90     11166
           1       0.71      0.56      0.63      3487

    accuracy                           0.84     14653
   macro avg       0.79      0.75      0.76     14653
weighted avg       0.83      0.84      0.83     14653

Model accuracy: 0.8410564389544803
```