Project Overview: Predicting Income Levels Using a Deep Neural Network

1. Objective

The primary goal of this project is to build and optimize a binary classification model to predict whether an individual's income exceeds $50,000 annually based on demographic and employment attributes. The model leverages a deep neural network (DNN) architecture built with TensorFlow/Keras and incorporates several stages of data preprocessing, feature engineering, model tuning, and evaluation.

2. Dataset Description

The dataset originates from the U.S. Census and includes both training and testing subsets. Each record consists of features such as work class, education, marital status, occupation, race, and native country, among others.

- Target Variable: income (binary: <=50K or >50K)
- Features: A mix of categorical and numerical variables, including both continuous (e.g., age, demogweight) and ordinal/categorical (e.g., education, relationship) data types.

3. Data Cleaning and Transformation

Missing Value Treatment

Non-standard indicators of missing values, such as "?" and empty strings, were identified and replaced with NaN. These missing values were imputed using the mode of each respective column.

Feature Reduction

The feature education-num was removed due to redundancy with the education column.

Category Consolidation

To reduce the sparsity of categorical variables, countries other than the United States were grouped into a single "Other" category for the native-country feature.

One-Hot Encoding

All categorical variables were converted to binary indicators using one-hot encoding, enabling their inclusion in the neural network input.

4. Exploratory Data Analysis (EDA)

- Crosstab Visualizations were generated to assess the relationship between each categorical feature and income level.
- Bar charts and histograms were used to visualize income distribution and the distribution of key numeric predictors such as age and demogweight.
- A correlation matrix was calculated to examine linear relationships among numeric features.
- Distribution plots revealed skewness in variables like age, informing feature engineering.

5. Data Preparation for Modeling

Train-Test Split

The dataset was split into training (70%) and testing (30%) subsets using stratified sampling to preserve class distribution.

Class Imbalance Handling

The training set exhibited class imbalance, which was addressed using SMOTE (Synthetic Minority Over-sampling Technique) to synthetically balance the minority class.

Feature Scaling

A Min-Max scaler was applied to normalize feature values to the [0, 1] range, ensuring faster and more stable convergence of the neural network.

6. Model Architecture and Training

A feedforward deep neural network was built using TensorFlow's Keras API. The architecture consisted of:

- An input layer with 65 features
- Two hidden layers with 64 and 32 units, respectively
- Sigmoid activation functions for all layers
- A sigmoid output unit for binary classification
- Loss function: Binary Crossentropy
- Optimizer: RMSprop
- Performance Metric: Accuracy

The model was trained for 200 epochs with a batch size of 50.

7. Feature Importance Analysis

Post-training, weights from the first hidden layer were extracted and analyzed to assess feature importance. The average absolute weight magnitude across neurons was used to rank features. A horizontal bar chart displayed the top 10 most influential features contributing to income prediction.

8. Model Evaluation

The model's predictive performance was assessed using:

- Accuracy score
- Confusion matrix
- Classification report (precision, recall, F1-score)

These metrics were computed on the training set, which was balanced using SMOTE, ensuring reliable performance indicators.

9. Feature Relationship Analysis

Using the predicted high-income group, the following analyses were performed:

- Occupational and Educational Associations: Identified which dummy-encoded job roles and education levels were most frequently associated with incomes >$50K.
- Age Distribution: Plotted the age distribution of individuals predicted to earn more than $50K to identify income-demographic trends.

Additionally, histograms were generated to compare the distribution of an important predictor (age) and a less important one (demogweight) across income classes.

10. Model Optimization (Hyperparameter Tuning)

Two model variants were built using different activation functions and tuned via Grid Search:

Model 1: Sigmoid Activation

- Tuned parameters: optimizer, batch_size, epochs
- Optimization technique: GridSearchCV using KerasClassifier from scikeras

Model 2: ReLU Activation

- Similar tuning approach applied with ReLU activations in the hidden layers.

Both models were evaluated on a cross-validation scheme (3-fold) to select the optimal parameter configuration. The best-performing model was then tested on the hold-out set for final accuracy estimation.

11. Final Deployment on Test Data

The test dataset was subjected to the same preprocessing pipeline:

- Dropping of education-num
- Imputation of missing values
- One-hot encoding
- Feature scaling with the pre-fitted scaler

Predictions were generated using the best-tuned model, and results were exported for external evaluation or submission.

12. Conclusion

This project successfully developed a deep learning model capable of predicting income classes with high accuracy. Key accomplishments include:

- Effective handling of missing data and class imbalance
- Insightful feature importance analysis based on learned neural network weights
- Robust performance tuning using grid search and cross-validation
- Deployment-ready preprocessing and prediction pipeline

13. Recommendations for Future Work

- Introduce dropout layers to reduce potential overfitting.
- Experiment with deeper architectures or convolutional layers for pattern extraction.
- Apply ensemble methods such as bagging or stacking for performance gains.
- Incorporate additional socio-economic features (if available) to enhance model context.