Project Income Prediction Report

The goal of this project is to predict the income level of individuals based on various features. The dataset used for this project is "train_data.csv." The following report provides an overview of the project workflow, including data preprocessing, feature selection, and model evaluation.

1. Data Preprocessing

The initial step in the project is to preprocess the data to ensure its quality and suitability for analysis. The following preprocessing steps were performed:

1. Data Import: The dataset "train_data.csv" was imported using the pandas library, and the data was stored in a dataframe named **`df`**.
2. Handling Missing Values: Any missing values in the dataset were replaced with appropriate values to facilitate analysis. The code **`df=df.replace(' ?', None)`** was used to replace question marks, which were considered null values, with actual null values.
3. Exploratory Data Analysis: The data was explored to gain insights into its structure and distribution. The data types of each column were checked using the **`df.dtypes`** function. Additionally, a count plot and distribution plots were created to visualize the income distribution based on different features like age and sex.
4. Handling Categorical Data: Categorical variables were encoded using one-hot encoding to convert them into numerical representations. The **`pd.get_dummies`** function was used to create dummy variables for categorical columns like workclass, education, occupation, and relationship.
5. Feature Selection: A feature selection technique called chi-square was applied to select the most relevant features for predicting income. The GenericUnivariateSelect class from sklearn.feature_selection module was used with chi2 as the score function and k_best as the mode. The selected features were education-num, capital-gain, capital-loss, and hours-per-week.
6. Train-Test Split: The preprocessed data was split into training and testing sets using the train_test_split function from sklearn.model_selection. The split was performed with 75% of the data used for training and 25% for testing.

2. Model Training and Evaluation

Multiple models were trained on the preprocessed data, and their performance was evaluated using various evaluation metrics. The models used for income prediction were:

1. Logistic Regression: A logistic regression model was trained on the training data using the LogisticRegression class from sklearn.linear_model. The trained model was then used to make predictions on the testing data.

2. Support Vector Machine (SVM): A support vector machine model with a linear kernel was trained on the training data using the SVC class from sklearn.svm. Similar to logistic regression, predictions were made on the testing data.

3. Decision Tree: A decision tree classifier was trained on the training data using the DecisionTreeClassifier class from sklearn.tree. Predictions were made on the testing data using the trained decision tree model.

For each model, the following evaluation metrics were calculated:

- Accuracy: The accuracy score measures the overall correctness of the predictions.
- Precision: Precision is the ratio of true positives to the sum of true positives and false positives. It represents the model's ability to correctly identify positive instances.
- Recall: Recall, also known as sensitivity or true positive rate, is the ratio of true positives to the sum of true positives and false negatives. It measures the model's ability to correctly identify positive instances out of all actual positive instances.
- F1-score: The F1-score is the harmonic mean of precision and recall. It provides a balanced measure of a model's performance.
- Confusion Matrix: The confusion matrix provides a detailed breakdown of the model's predictions, showing the number of true positives, true negatives, false positives, and false negatives.

3. Results

The performance metrics of each model on the testing data are as follows:

Logistic Regression:

- Accuracy: 0.812
- Precision: 0.670
- Recall: 0.507
- F1-score: 0.577

Support Vector Machine (SVM):

- Accuracy: 0.813
- Precision: 0.678
- Recall: 0.504
- F1-score: 0.578
-

Decision Tree:

- Accuracy: 0.798
- Precision: 0.596
- Recall: 0.571
- F1-score: 0.583

4. Conclusion

Overall, this project provides insights into income prediction based on various factors and demonstrates the effectiveness of machine learning models in addressing such prediction tasks.