An income prediction project is a supervised Artificial Intelligence project that aims to predict the income level or salary of individuals based on various demographic, economic, and employment-related features. The project involves collecting and preprocessing a dataset that contains information about the individuals, such as age, gender, education level, occupation, marital status, race, as well as their salary. The dataset is then used to train an artificial intelligence model, with logistic regression, SVM, decision tree and random forest, which can then predict the income level of new individuals based on their features. The model can be evaluated using various evaluation metrics, such as accuracy, precision, recall, and F1 score, as well as confusion matrix of each classification algorithm to assess its performance and identify areas for improvement.

1st step, preprocessing:

The model preprocesses the two csv files train\_data and test\_data, the preprocessing drops null values, drops duplicates, drop illegal formats, converts values to their right format (example: age to float) and encodes the categorical values into numerical values.

2nd step, feature selection:

After preprocessing, we can now start feature selection. The model uses Lasso regression to select features based on how they affect the Income, the lasso regression uses two hyper-parameters: Alpha and Max iteration. By manually trying different values for the hyper-parameters we get different accuracy and precision values but it becomes hard to precisely pinpoint the exact values which can give the best accuracy and precision, so we used a grid search to let the model automatically try a wide range of alpha and max iteration values to identify the best performing values.

After the model uses grid search with this dataset we can see it chose value of alpha 0.01 and max iteration value of 1000.

3rd step, classification:

Now that we found the best hyper-parameter values and used them to select features, we can then train our model using logistic regression, SVM, decision tree and random forest.

4th step, evaluation:

After training the model, we can now test it with our test data and evaluate the performance of our model, to measure the performance we can use a lot of helpful statistical variables such as Accuracy, precision, recall and F1-score. It’s also worth mentioning that all these variables are each derived from a mathematical equation that uses the variables within the confusion matrix, the confusion matrix is the non-abstract way of measuring out performance, the confusion matrix consists of four elements, true positive, false positive, false negative and true negative, each of these elements represent the relation between the actual value and the predicted value, for instance, giving a correct answer to a right test is considered true positive.

* Here are the performance measurements:

-Logistic Regression: Accuracy = 0.814546662238459, Precision = 0.7288306451612904, Recall = 0.3908108108108108, F1-score = 0.5087966220971147

-SVM: Accuracy = 0.8049817336433078, Precision = 0.7298435619735258, Recall = 0.3278378378378378, F1-score = 0.45244311823946287

-Decision Tree: Accuracy = 0.764530056459648, Precision = 0.5213675213675214, Recall = 0.5110810810810811, F1-score = 0.5161730585505664

-Random Forest: Accuracy = 0.8181335104616406, Precision = 0.7224791859389454, Recall = 0.4221621621621622, F1-score = 0.5329239167519619

* And we get the following confusion matrix values:

-Logistic Regression Confusion Matrix:

[[10817 538]

[ 2254 1446]]

-SVM Confusion Matrix:

[[10906 449]

[ 2487 1213]]

-Decision Tree Confusion Matrix:

[[9619 1736]

[1809 1891]]

-Random Forest Confusion Matrix:

[[10755 600]

[ 2138 1562]]