CS 273P Project: determine whether a person makes over 50K a year

Student Name: Jing Tang ID: 86729684
Student Name: Jiada Ye ID: 32964504

Introduction:

We used the dataset Adult to predict whether a person makes over 50K a year through several machine learning algorithms. We first used all the features and took feature selection to improve the prediction accuracy. The we tuned the model of KNN, random forest, CVM and have the best prediction accuracy around 0.867 with CVM.

Data Analysis and Processing:

Firstly, we loaded the dataset and tested dataset and split the dataset into training data and validation data with the provided code.
The dataset contained 14 features, 7 of them were numeric features: age, fnlwgt, education-num, capital-gain, capital-loss and hours-per-week, income; and seven of them were not numeric features: workclass, education, marital.status, occupation, relationship, race, sex, native. We converted the non-numeric features into numeric ones, i.e replaced male with 0 and replaced female with 1. For the target 'income' we would predict, we used 0 to represent '<=50K' and use 1 to represent '>50k'.
Secondly, we processed some features and data. For feature 'marital.status', we combined 'Never-married',' Divorced',' Separated',' Widowed' to 'Single' and combined 'Married-civ-spouse',' Married-spouse-absent',' Married-AF-spouse' to 'Married'.

Feature Engineering and Model Comparison:

Since the meanings of some features were overlapped, we dropped 'relationship' and 'education' and kept 'marital.status' and 'education.num'. Then we explored the relationship of education and marital status with income by catplot, the figure shows that married adults have higher probability to have over 50K income, and the probability of making over 50K income increases with the increase of education.num:

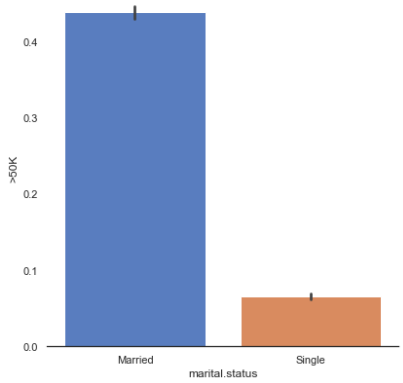

Figure 1. The Possibility Distribution of Marital Status to Income > 50k

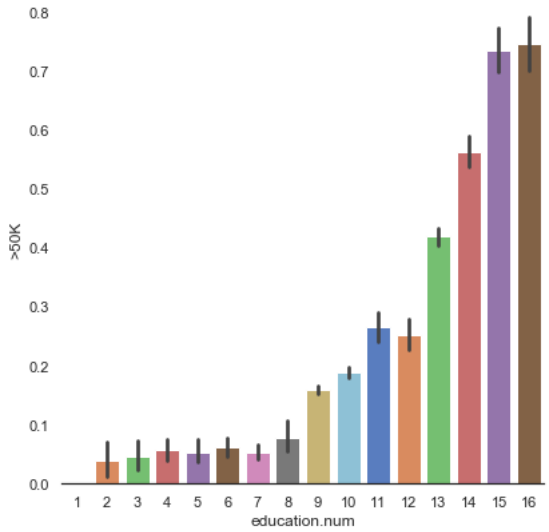

Figure 2. The Possibility Distribution of Education Num to Income > 50k

From the quantity graph of native.country, we can observe that most of the adults are from United States, so the feature is not so meaningful for the prediction and we dropped the feature.

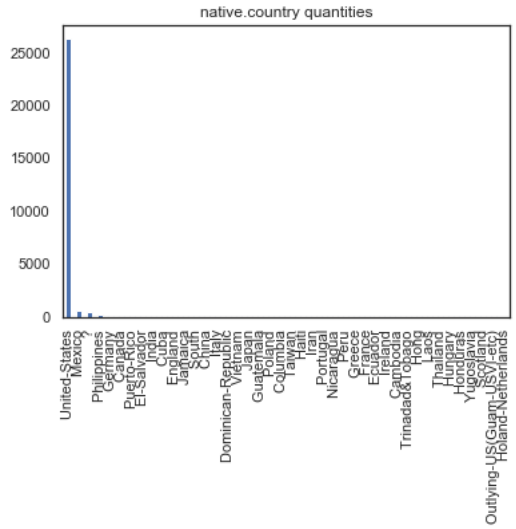

Figure 3. The Country Distribution of Dataset

We converted all other non-numeric features into numeric features and we used the data dropped “education","relationship","native.country” to train 5 models: KNN, Decision Tree, Gaussian Naive Bayes, Random Forest and SVM and use cross-validation to get the mean accuracy score of them.

KNN: 0.665087
DecisionTree: 0.717551
GaussianNB: 0.669418
RandomForest: 0.761794

SVM: 0.666484

The accuracy was not satisfying so we decided to select the features most related to ‘income’ to do the prediction. Then, we used heatmap to observe the correlation between features.

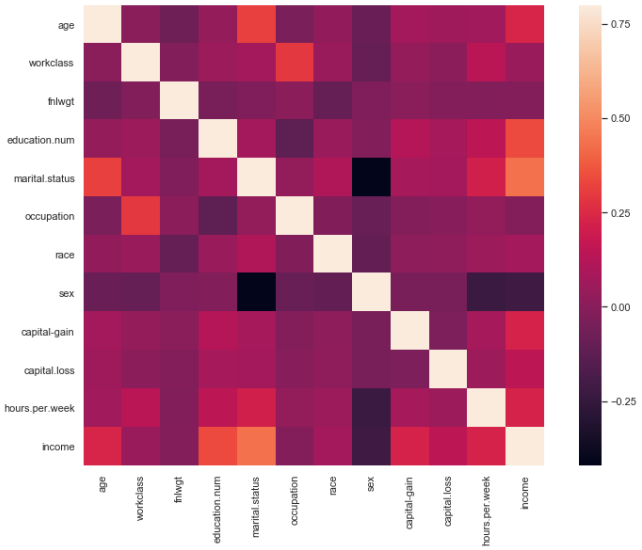

Figure 4. The Heatmap About the Relevance of Income to Different Features

From the heatmap, it can be observed that some features are not strongly related to income, such as ‘sex’ and ‘fnlwgt’. To find the features most related to ‘income’, we used ‘nlargest’ to list the top ten income-related features and we can see that the features ‘marital.status’, ‘education.num’, ‘age’ have the strongest relationship with ‘income’.

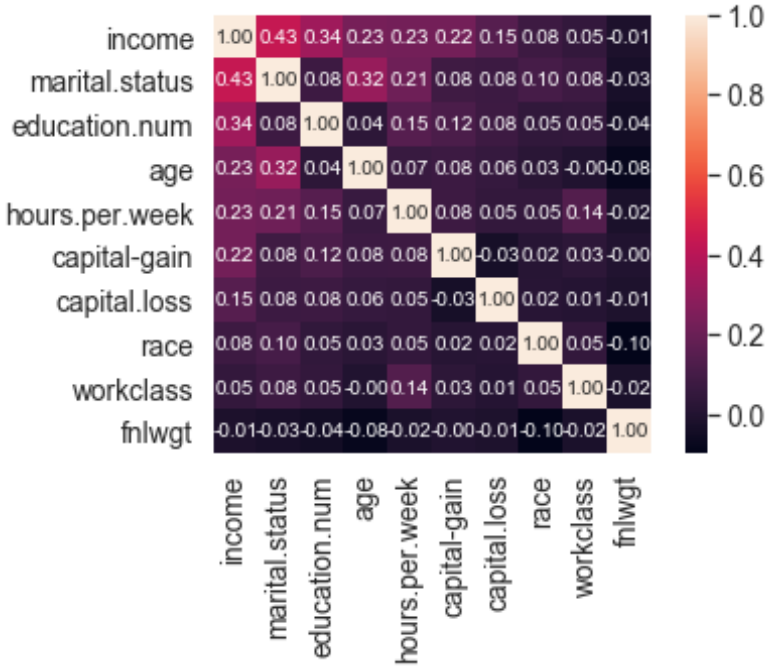

Figure 5. The 10-largest Heatmap of Relevance of Income to Features

Then we dropped the features with weakest relationship with 'income', which were 'sex', 'occupation','fnlwgt', and 'workclass'.
We used the train dataset to train the 5 models: KNN, Decision Tree, Gaussian Naive Bayes, Random Forest and SVM and used the validation dataset to calculate the accuracy of the 5 models.
KNN: 0.838428
DecisionTree: 0.829945
GaussianNB: 0.801602
RandomForest: 0.840727
SVM: 0.829636
All the results from selected features seem improved significantly compared to those from the original features.

Model Tuning:

We chose three algorithms (KNN, SVM and Random Forest) from the five algorithms above to take some tuning and comparison.

KNN:

KNN can be used for both classification and regression predictive problems. At K=1, we were overfitting the boundaries. Hence, accuracy initially increased and reached to a maximum. By varying the k from 1 to 30, we found the k that resulted in the best accuracy was 12 with accuracy 0.8508934072704868.

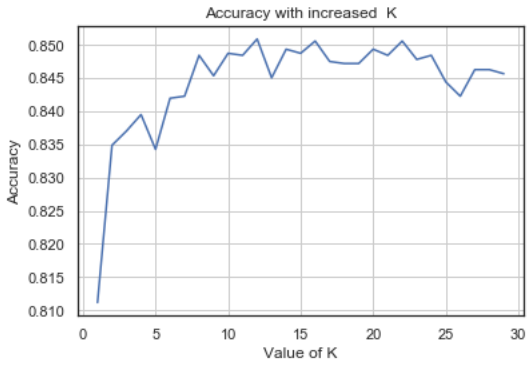

Figure 6. KNN Accuracy with increased K

And we tested the model on test dataset, the accuracy of the model was 0.8484122596892083.

Random Forest:

Random forests are an ensemble learning method for classification, regression and other tasks that operates by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees [1].
n_estimators represents the number of trees in the forest. Usually the higher the number of trees the better to learn the data. However, adding a lot of trees can slow down the training

process considerably. max_features represent the number of features to consider when looking for the best split.
By varying the n_estimator in the range [50,100,150,200,250], we got the best accuracy 0.8450 When n_estimator= 150.

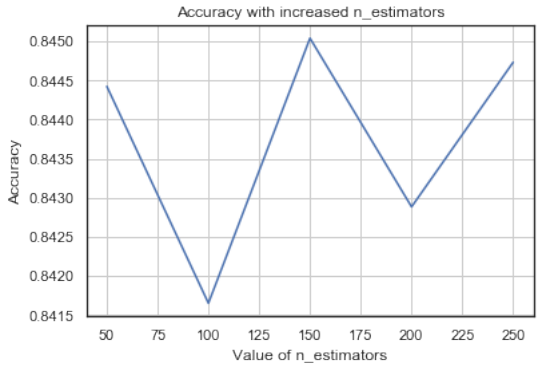

Figure 7. Random Forest Accuracy with increased n_estimators

By varying the max_feature int [1,2,3,4,5,6,7], we got the best accuracy 0.8435 when max_feature = 5

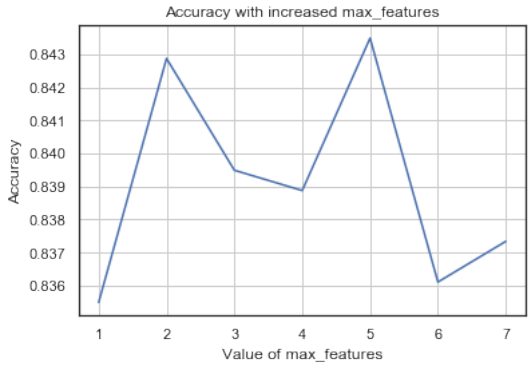

Figure 8. Random Forest Accuracy with increased max_features

Then we tested the model on test dataset, the accuracy was 0.8427615011362939.

SVM:

We used different kernels of support vectors: linear, rbf and sigmoid to train the model and got accuracy:

Linear ACC: 0.7735674676524954
rbf ACC: 0.8296364756623537
sigmoid ACC: 0.7578558225508318

Then we tuned the parameter gamma of SVM with RBF kernel. The gamma parameter controls the distance of a single training example can influence. For lower gamma, its influences can reach far and for higher values its influence can reach close. The gamma parameters can be seen as the inverse of the radius of influence of samples selected by the model as support vectors [4].

First we varied the gamma in the range [1.0e-08,1.0e-07,1.0e-06,1.0e-05,1.0e-04,1.0e-03,1.0e-02,1.0e-01], SVM got the highest accuracy 0.8539741219963032 when gamma = 1.0e-03.

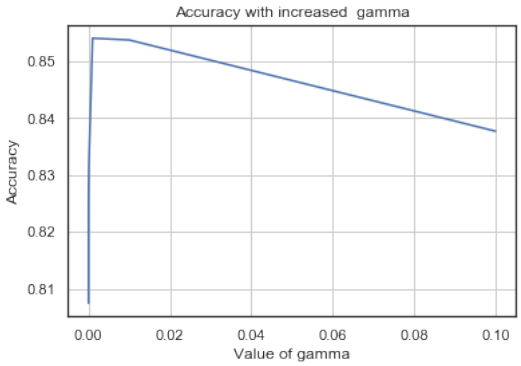

Figure 9. SVM Accuracy with increased Gamma

Then, we tested the model on test dataset, the accuracy of the model was 0.8572569252502917.
To further improve the performance of SVM model, we adjusted the C which is the penalty parameter of the error term. We tried 1 (default), 10, 100, 200, 300 and 400, and found it was at C = 300 that the score was highest (0.865978748234138). Then we tuned the C from 290 to 310 and found C=300 still the best.

Conclusion

In this project, we analyzed the dataset and the features and made a feature selection according to their relevance with the prediction target, the we used the five models to do the prediction on the selected features. Among the five models, knn, random forest and SVM had higher accuracies, therefore, we chose them to do the parameter tuning. For KNN, the model got the highest accuracy 0.8484 with k=12. For random forest, the model got the highest accuracy 0.8428 with the parameter: {n_estimators = 150, max_features= 5}. For SVM, first we made a kernel selection, and tuned the parameter gamma and C with RBF kernel, the model gets the highest accuracy 0.866 with gamma = 1.0e-03 and C = 300.
We choose SVM with RBF kernel, C = 300 and gamma = 1.0e-03 as our best model.

Reference:

1. https://en.wikipedia.org/wiki/Random_forest
2. https://www.kaggle.com/iranneto/a-simple-knn-application
3. https://www.kaggle.com/ipbyrne/income-prediction-84-369-accuracy
4. https://scikit-learn.org/stable/auto_examples/svm/plot_rbf_parameters.html

Appendix

The code of the project is in the jupyter notebook file project.ipynb submitted to the EEE dropbox. The dataset files and data_loader.py should be put in the same path with project.ipynb.