Data mining
Analysis of adult dataset

Krystian Walewski, Katarzyna Gunia

November 2023

1 Problem description

This project is based on Adult dataset, also known as "Census Income". This data contains several features about socio-economic status as well as information about person's income. In the first stage of the project, we will conduct an exploratory analysis of the dataset, to investigate the data and initially assess discriminative abilities of features. We could see if there are significant differences in income between females and males, between different races, or between people in different types of relationships.

However, the main goal of this project is to create an accurate classifier which would be predicting if the yearly income exceeds \$50K, to do so we will compare several different models on different features subsets and try to find the best one. Finally, we would like to see which of the considered features are the most important in classification procedure? Which of them holds the most information about the income level?

2 Data characteristics

Adult dataset consists of 48842 instances and 14 features, both quantitate and qualitative and binary target variable — salary. The features are the following:

- age — describes the age of individuals, ranges from 17 to 90,
- workclass — Private/ Self-emp-not-inc/ Self-emp-inc/ Federal-gov/ Local-gov/ State-gov/ Without-pay/ Never-worked,
- fnlwgt — final weight, the number of people the census believes the entry represents,
- education — Preschool/ 1st-4th/ 5th-6th/ 7th-8th/ 9th/ 10th/ 11th/ 12th/ HS-grad/ Some-college/ Assoc-voc/ Assoc-acdm/ Bachelors/ Masters/ Prof-school/ Doctorate,

- education-num — encoded education column, number from 1 to 16 denote respective education level
- marital-status — Married-civ-spouse/ Divorced/ Never-married/ Separated/ Widowed/ Married-spouse-absent/ Married-AF-spouse,
- occupation — Tech-support/ Craft-repair/ Other-service/ Sales/ Exec-managerial/ Prof-specialty/ Handlers-cleaners/ Machine-op-inspct/ Adm-clerical/ Farming-fishing/ Transport-moving/ Priv-house-serv/ Protective-serv/ Armed-Forces,
- relationship — Wife/ Own-child/ Husband/ Not-in-family/ Other-relative/ Unmarried,
- race — White/ Asian-Pac-Islander/ Amer-Indian-Eskimo/ Other/ Black,
- sex — Female/ Male,
- capital-gain — profit of the person,
- capital-loss — loss of the person,
- hours-per-week — number of hours the person works per week,
- native-country — United-States/ Cambodia/ England/ Puerto-Rico/ Canada/ Germany/ Outlying-US(Guam-USVI-etc)/ India/ Japan/ Greece/ South/ China/ Cuba/ Iran/ Honduras/ Philippines/ Italy/ Poland/ Jamaica/ Vietnam/ Mexico/ Portugal/ Ireland/ France/ Dominican-Republic/ Laos/ Ecuador/ Taiwan/ Haiti/ Columbia/ Hungary/ Guatemala/ Nicaragua/ Scotland/ Thailand/ Yugoslavia/ El-Salvador/ Trinadad&Tobago/ Peru/ Hong/ Holand-Netherlands,
- salary — $> 50K$/ $<= 50K$

Adult dataset contains missing values, coded with question mark "?". Missing values appear only in columns: workclass (5.7%), occupation (5.8%) and native-country (1.8%). On the figure 1 we can see a nullity matrix. We can notice an interesting pattern, all records with missing values in workclass also miss value in occupation column. Since the number of rows with nulls is relatively small, we decided to remove those records.

We also trimmed redundant white spaces and unified label: "$<= 50K$" and "$<= 50K.$", as well as "$> 50K$" and "$> 50K.$". We noticed that some rows have un opposite information in sex and relationship columns, e.g. sex = Female and relationship = Husband. There were only 2 such records, so we removed them.

Since education and education-num hold the same information, we dropped education column. On the other hand, feature native-country contains many rarely occurring values, for example only 81 persons are from Poland. The majority, more than 91% of people in the dataset have United-States as their native country. We decided to create a new column named 'from_USA', which will be equal to 1 if the person is from the USA and to 0 otherwise.

We were also wondering why quite many rows have capital-gain equal to the maximum value 99999. We suspect that in reality those persons could have higher profits, but value 99999 was the highest possible to input.

The clean dataset contains 45220 rows and 14 columns, including target variable.

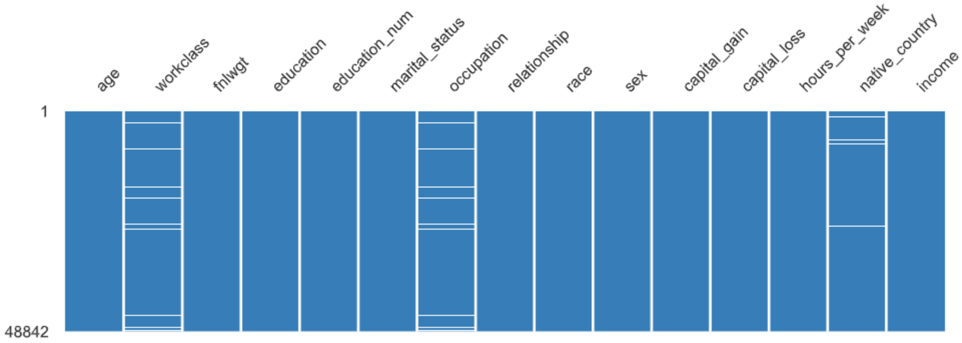

Figure 1: A nullity matrix of the dataset.

3 Exploratory Data Analysis

In this section, we will focus on the most important insights from data exploration. Additional plots can be found in the file: additional_insights.pdf.

Figure 2 shows the distribution of the target variable. We can see that the dataset is not balanced, since approx. 75% of people earn less than 50K.

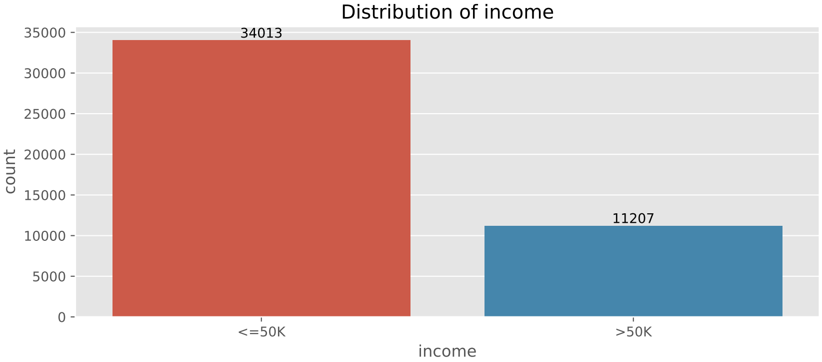

Figure 2: Distribution of income.

By investigating the distribution of the feature depending on the target variable, we can initially assess the discriminative ability of this feature. Plot 3 shows the difference in age distribution between people earning less than 50K and more than 50K. We see that younger people have usually smaller income. The plot 4, with boxplots of age depending on income also show that people with higher income are on average older. However, both boxplots overlap, so it might not be the best feature to distinguish between target variable.

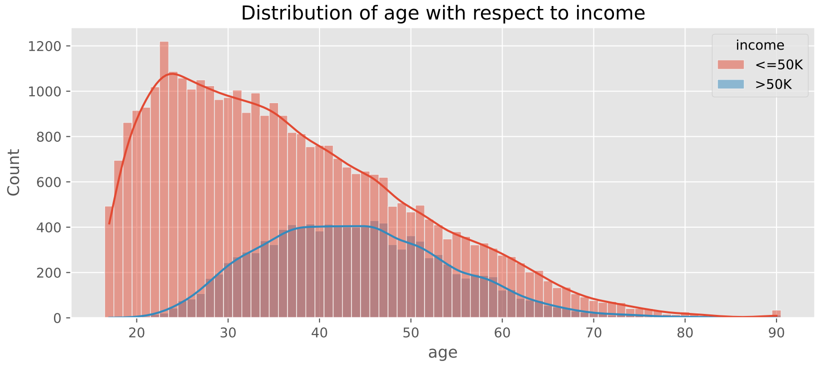

Figure 3: Histogram of age depending on the income.

The next plot 5 contains the similar boxplots, but of feature hours-per-week. That feature has many outliers, but we can also observe that people working more hours tend to earn more.

Education level might be useful when predicting the class label, since people with higher income usually have higher education level, which can be seen on the plot 6. According to that plot, bachelors are the biggest group with income larger than 50K. On the other hand, the largest education group with income lass than 50K are high-school graduates.

We also observe big differences in between groups for marital status and relationship. Showed on plot 7 and 8 respectively. Married people, especially husbands, tend to earn more money, compared to other groups.

We can also observe some differences between occupations and income level, shown on figure 9. However, these differences are not so clear, because occupations: “Prof-speciality” and “Exec-managerial” which have the highest number of income above 50K also have the similar level number of income below 50K.

When it comes to capital gain and capital loss, those features are in 91.6% and 95.3% zeros, hence it is difficult to visualize them. However, we can observe the differences in mean gain and loss between income groups, displayed in the table 1. In both cases, the group with higher income has definitely larger mean gain and loss.

We didn’t notice significant differences between groups in workclass, sex, race, fnlwgt and from_USA features. Appropriate figures can be found in the additional file.

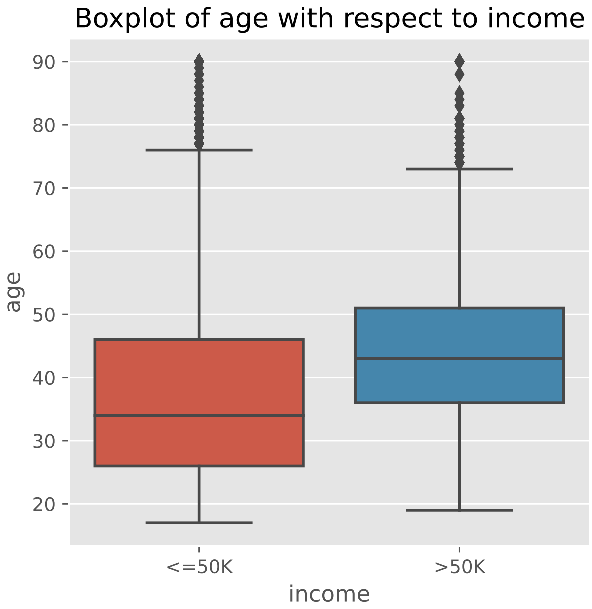

Figure 4: Boxplot of age depending on the income.

income	capital_gain	capital_loss
<=50K	149.023	54.03
>50K	3991.79	193.49

Table 1: Mean value of capital gain and capital loss in income groups.

To investigate correlation between features, we transformed categorical features into numerical ones, using ordinal encoding and computed Pearson correlation, shown on figure 10. Workclass, fnlwgt, race and from_USA are the least correlated features with our target variable. We didn't actually see the differences in distributions between groups for these features, so deleting them shouldn't harm the model.

4 Methods and algorithms used in the project

To create a classification model, we considered several algorithms including linear and quadratic discriminant analysis, k-nearest neighbours, logistic regression, random forest and XGBoost. Our dataset was split in a stratified way into train (75%) and test (25%) parts, to ensure that the distribution of labels is the same in both sets.

We prepared a few versions of input data depending on the model. For LDA,

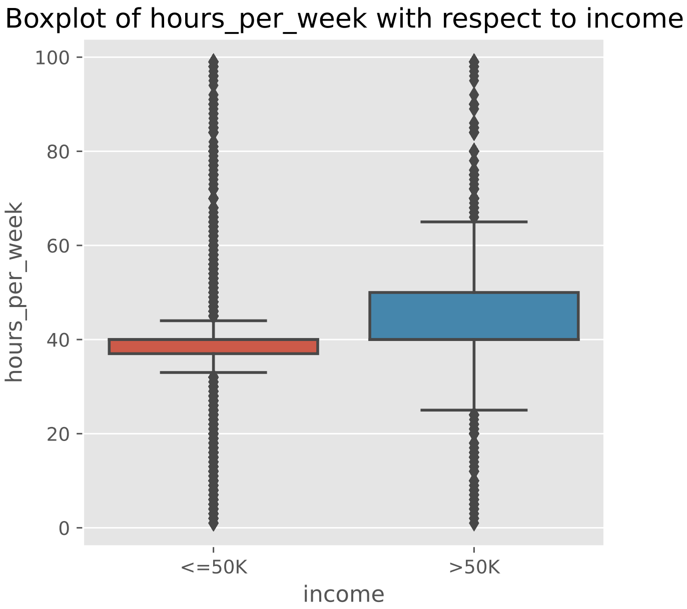

Figure 5: Boxplot of worked hours per week depending on the income.

QDA, KNN, logistic regression and random forest we used one-hot encoding to encode categorical variables. Only XGBoost doesn't require it, so we trained it on numerical and categorical features.

Since KNN is based on distance metrics, we decided to standardize the features, so each of them could have the same impact on the model. We also tried to use Gower'r distance, but we failed and had the same problem as the person writing this mail. We compared performance of logistic regression on standardized and not standardized data. We were also testing min-max scaling, however it gave us either the same or worse results, so we didn't include them in the model.

To check performance of the models on train data, we deployed 10-fold stratified cross validation with 10 repeats. We compared the results for four different measures: accuracy, recall, precision and AUROC. 10-fold stratified cross validation with 5 repeats was also used to find the optimal number of neighbours in KNN algorithm. We performed grid search and compared accuracy for number of neighbours, raging from 1 to 45. The optimal number of neighbours was chosen as 38.

Then we decided to check what will happen if we get rid of not important attributes. Columns fnlwgt, workclass and from_USA were dropped as they were most uncorrelated with salary. We used also feature importance from XGBoost model to find that fnlwgt, *race* and from_USA were considered as the least important based on average gain across all splits the feature is used in. It

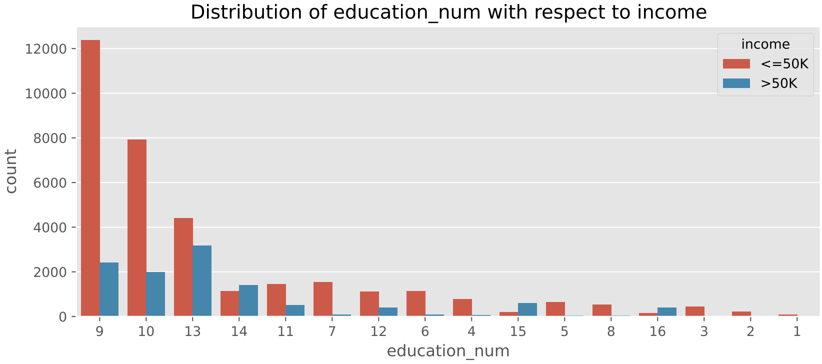

Figure 6: Differences in education levels between income labels.

resulted in the same or even better accuracy scores.

The last stage of developing models was to find the best hyperparameters. We used once again grid search and cross-validation to find the best set of them, comparing accuracy score. We tuned the following parameters:

- LDA — 'solver',
- KNN — 'weights' and 'p',
- Random Forest — 'n_estimators' and 'criterion,
- XGBoost — 'scale_pos_weight', 'eta'and 'max_depth'.

At the end, we trained the classifiers with the optimal parameters on the whole training set and validated them on the test set. We compared their performance measures as well as ROC and precision-recall curves.

5 Results

On the figure 11 are the boxplots of accuracy calculated on train data using cross validation for initial models. We can see that all the models except QDA have similar variance. For QDA we obtained a lot of small outlying scores. We should keep in mind that in our dataset around 25% of people earn more than 50K, and we have an unbalanced dataset. So even if our classifier would predict that everyone earn less than 50K, it will still have quite decent accuracy around 75%. So QDA model is indeed very bad. On the other hand, XGBoost seems to perform the best. We can also observe that logistic regression perform surprisingly better for standardized data. It is because by default we use the L2 regularization. We should always standardize the data before any kind of regularization to ensure that the regularization is applied uniformly across all features.

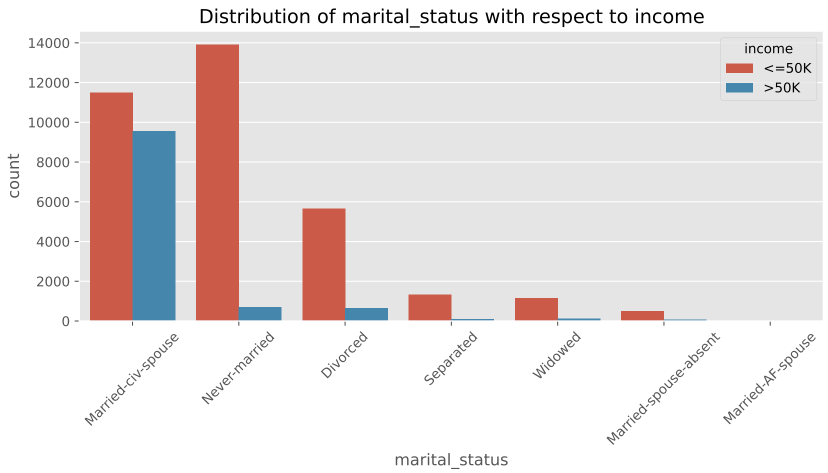

Figure 7: Differences in marital status between income.

Looking at figure 12 with boxplots of AUROC, XGBoost is again the best model. Random forest and logistic regression for standardized features also performed well. This time, logistic regression for non standardized data is definitely the worst one.

Plots 11 and 12 contains results of training basic models, with all possible features, without any hyperparameters tuning. In the next step, we decided two test our model after dropping some features. At first, we dropped three most uncorrelated features: fnlwgt, workclass and from_USA. All models' parameters stayed the same.

Figure 13 present the same boxplots of accuracy as before but for models without three most uncorrelated features. It may look like the interquartile range greatly increased, this is because the scale changed. However, for logistic regression, the variance indeed increased. Models like QDA and logistic regression benefited from that feature drop. While the accuracy for random forest decreased. As previously, XGBoost is performing the best.

Plot 14 shows the comparison of AUROC for considered models, but without least correlated features. The results for logistic regression greatly improved, even though it is still the worst performing model.

In the next step, we tried different approach of feature selection, we deleted features that could mislead our best model XGBoost. Plots 15 and 16 present two different types of feature importance. On the first plot, we see the number of times each feature is used to split the data, across all trees. The second plot, 16, shows the average information gain due to that feature. We can see the inconsistent ordering on both plots. Feature fnlwgt occurs most often — 583 times, but gives very little information — approx. 2.66. This is why we decided to drop it.

Plots 17 and 18 present results of cross validation of the considered models

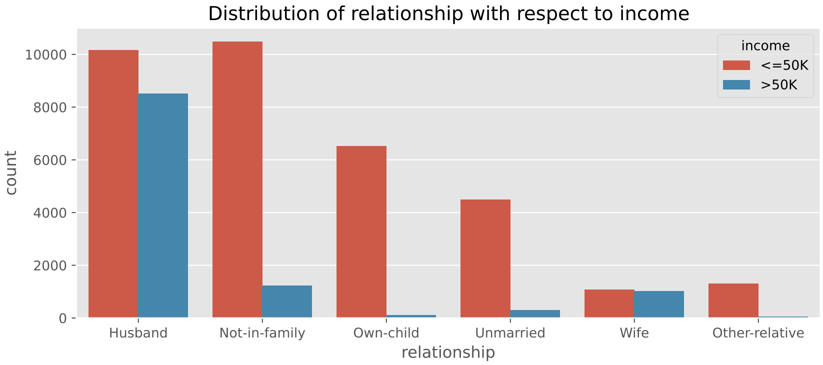

Figure 8: Differences in relationship status between income.

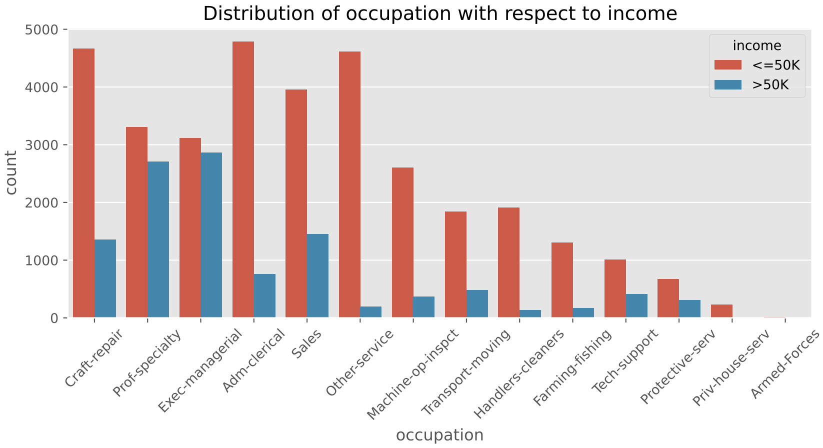

Figure 9: Differences in occupations and income.

without feature fnlwgt.

The performance of models didn't change that much compared to previous features drop. However, the dispersion of accuracy and AUROC decreased compared to previous feature selection. That is why we used this set without column fnlwgt in the next stage of model building — hyperparameters tuning.

We tuned various parameters of the models. Optimal values of these parameters can be found in the file with additional results. In XGBoost we tuned the parameter 'scale_pos_weight', which might be useful in case of unbalanced datasets, like ours. However, since we decided to compare our models mostly on accuracy, hence the optimal value turned out to be equivalent to no scaling.

Finally, we trained our tuned models on the whole training dataset without feature fnlwgt and then tested it on the test dataset, which were previously unused. The values of scoring measures can be found in the table 5. For easier

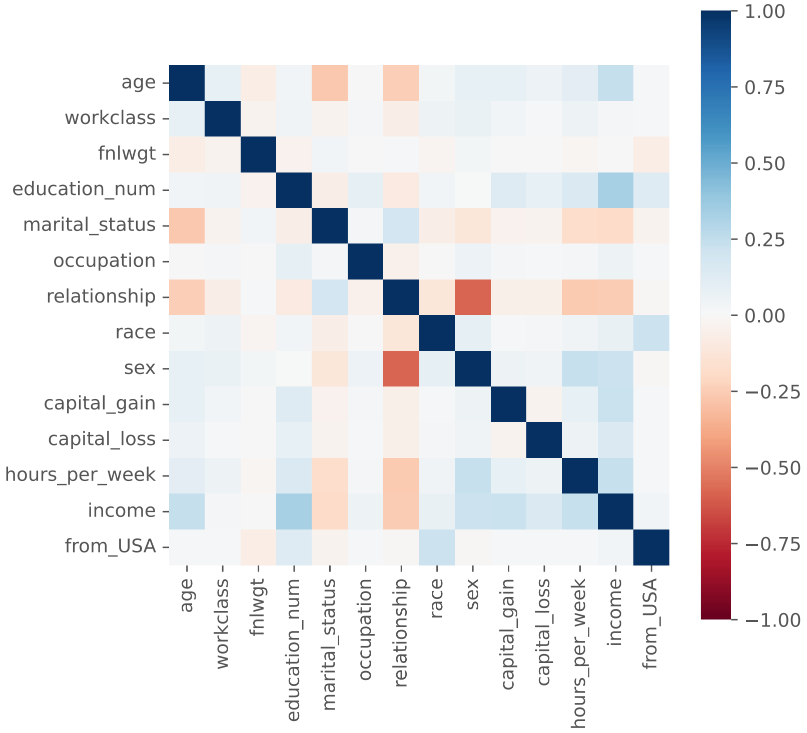

Figure 10: Differences in occupations and income.

comparison, we also created tables with mean scores from previous stated of model building. Performance of the initial models can be found in a table 2, of models after first feature selection based on correlation in a 3 and of models after dropping feature fnlwgt in a table 4. We didn't include in the main report boxplots of recall and precision, they can be found in the additional report. Comparing these tables, we can for instance notice that initially QDA had the highest recall.

Comparing different scoring measures, we get slightly different order of best performing models. However, XGBoost is the best for all measures. The next two plots present ROC curve — figure 19 and precision-recall curve — figure 20. They once again proof that XGBoost performs the best on our dataset. The second place takes ex aequo logistic regression for standardized data and random forest. On the other hand, QDA performed the worst.

Lastly, let's also look at the feature importance returned from our best model, XGBoost. On the plot 21 we can see that relationship turned out to be the most important feature. Marital status, which holds very similar information as the first feature, also turned out to be very substantial. One could think that education would be a crucial factor when it comes to earning, but is

model	accuracy	roc_auc	recall	precision
KNN	84.31	89.63	59.20	72.57
LDA	83.70	89.03	58.73	70.68
QDA	72.57	86.74	87.87	47.84
Logistic regression	79.03	58.32	26.48	70.82
Logistic regression (std)	84.75	90.30	60.65	73.30
Random Forest	85.01	90.24	62.08	73.47
XGBoost	86.65	92.46	66.03	76.93

Table 2: Comparison of mean accuracy scores of models in percentages.

model	accuracy	roc_auc	recall	precision
KNN	83.55	87.97	60.23	69.46
LDA	83.55	88.91	58.52	70.27
QDA	80.44	86.67	35.32	72.64
Logistic regression	82.01	86.25	52.33	67.89
Logistic regression (std)	84.58	90.16	60.40	72.86
Random Forest	83.95	88.49	61.92	70.13
XGBoost	86.65	92.45	65.92	77.01

Table 3: Comparison of mean accuracy scores [%] of models after dropping not correlated features: fnlwgt, workclass and from_USA.

model	accuracy	roc_auc	recall	precision
KNN	83.84	88.18	61.09	70.02
LDA	83.69	89.01	58.71	70.65
QDA	80.02	86.76	43.51	72.54
Logistic regression	82.45	86.84	53.43	68.93
Logistic regression (std)	84.65	90.28	60.45	73.07
Random Forest	84.21	88.93	61.99	70.74
XGBoost	86.83	92.57	66.15	77.53

Table 4: Comparison of mean accuracy scores [%] of models after dropping feature fnlwgt.

model	accuracy	roc_auc	recall	precision
KNN	84.60	76.00	59.00	73.34
LDA	83.53	74.74	57.39	70.36
QDA	82.05	67.51	38.81	77.01
Logistic Regression (std)	84.93	76.62	60.22	73.83
Random Forest	85.05	77.93	63.88	72.28
XGBoost	87.11	80.01	66.00	78.33

Table 5: Comparison of models' scores [%] on test set.

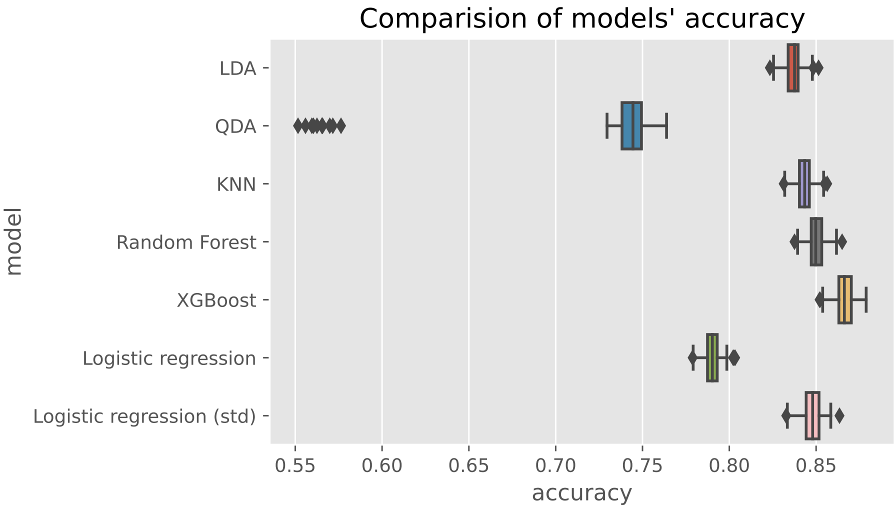

Figure 11: Boxplots of accuracy for considered models.

only third. Happily, race and native country have the smallest gain.

6 Conclusions

We managed to reach the main goal of this project and created an accurate classifier predicting weather, yearly income exceeds $50K. From all the considered models, XGBoost turned out to be the best, reaching 87% of accuracy and 80% of AUROC. It outperformed other classifiers for all comparison methods and all performance measures.

We also learned which features are the most and least important in predicting income level. Race and native country turned out to be the least informative features. On the other hand, relationship and marital status were the most important.

7 Further research suggestions

In our analysis we compared only a few classification models, we could expand this analysis by including models like SVM or neutral networks.

We also tuned only part of hyperparameters. It might be beneficial to check how big improvement we can obtain by tuning the remaining hyperparameters.

Another broad topic that could be studied in more details is the feature selection. We tested our data on the dataset with all features, without one feature and without three features, but we could increase the number of subsets. It would also be interesting to compare the feature importance returned from

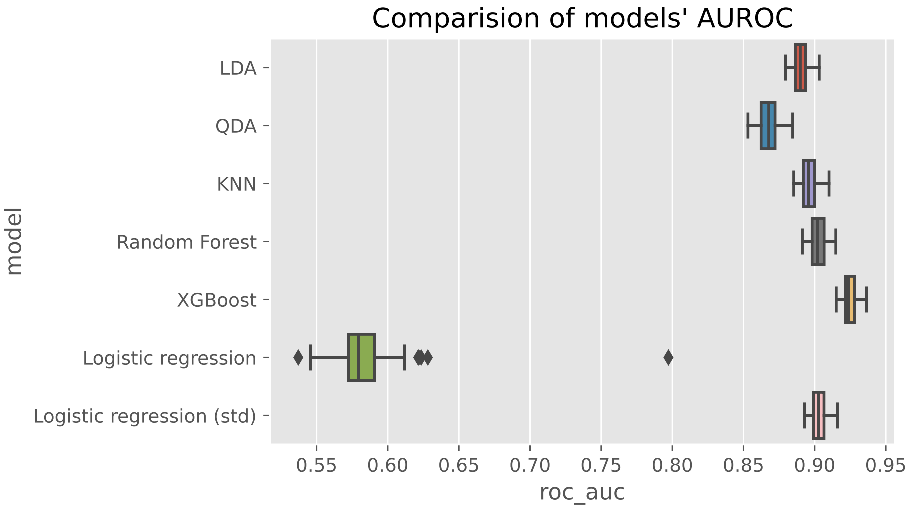

Figure 12: Boxplots of AUROC for considered models.

XGBoost with feature importance returned from e.g. random forest or logistic regression.

Even though, our dataset is not extremely unbalanced. We could check if thresholding or any sampling method could significantly improve our model.

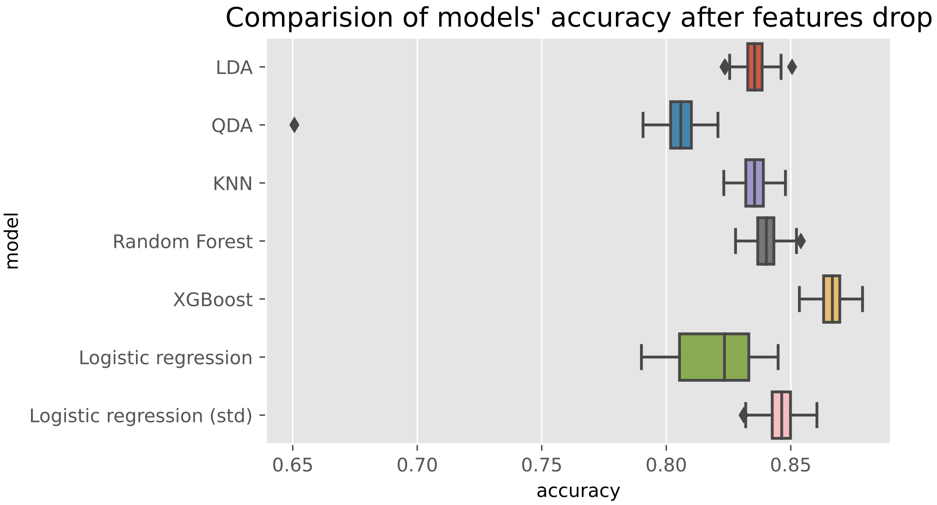

Figure 13: Boxplots of accuracy for considered models after dropping features: fnlwgt, workclass and from_USA.

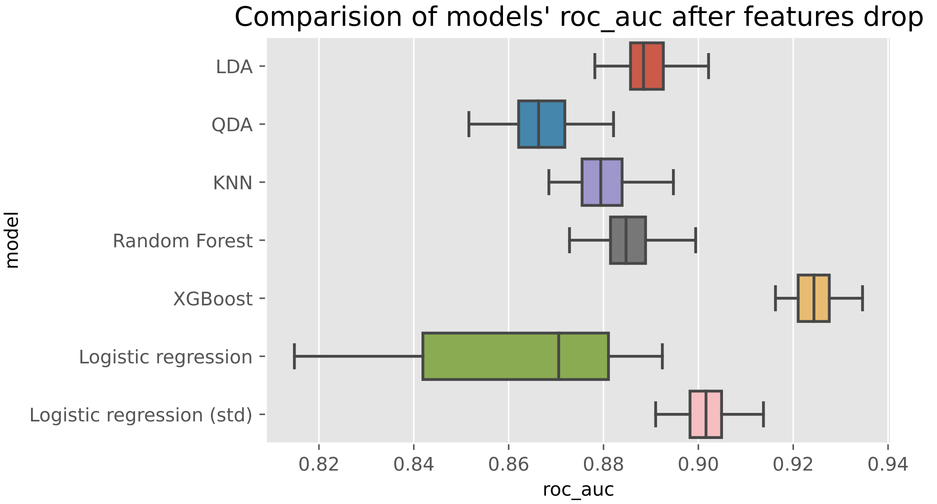

Figure 14: Boxplots of AUROC for considered models after dropping features: fnlwgt, workclass and from_USA.

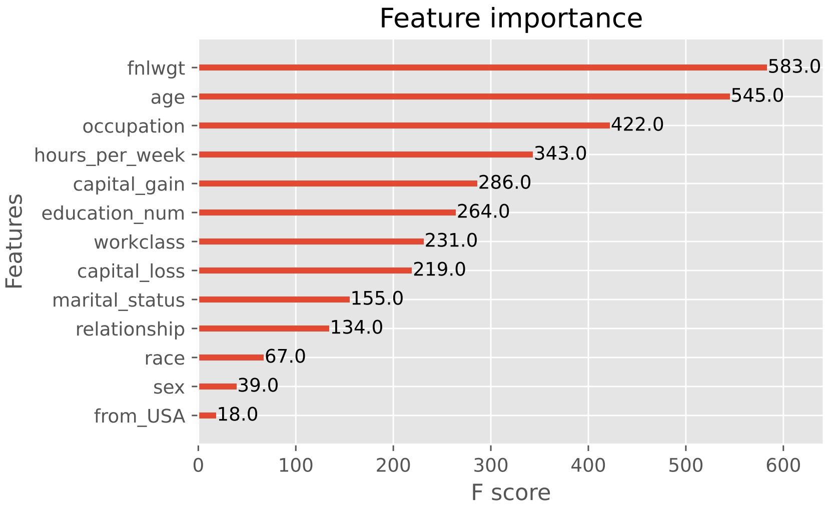

Figure 15: Number of times each feature in XGBoost model is used to split the data across all trees.

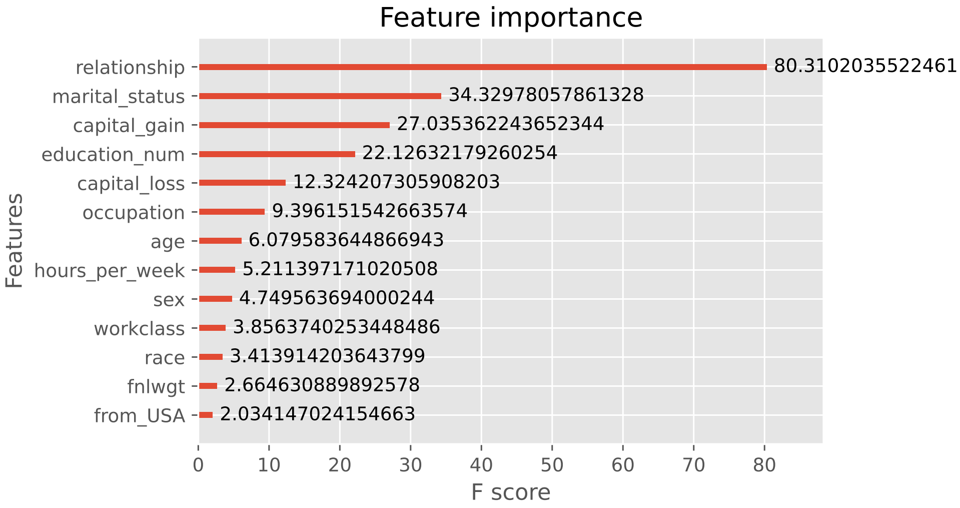

Figure 16: The average gain across all splits each feature in XGBoost model is used.

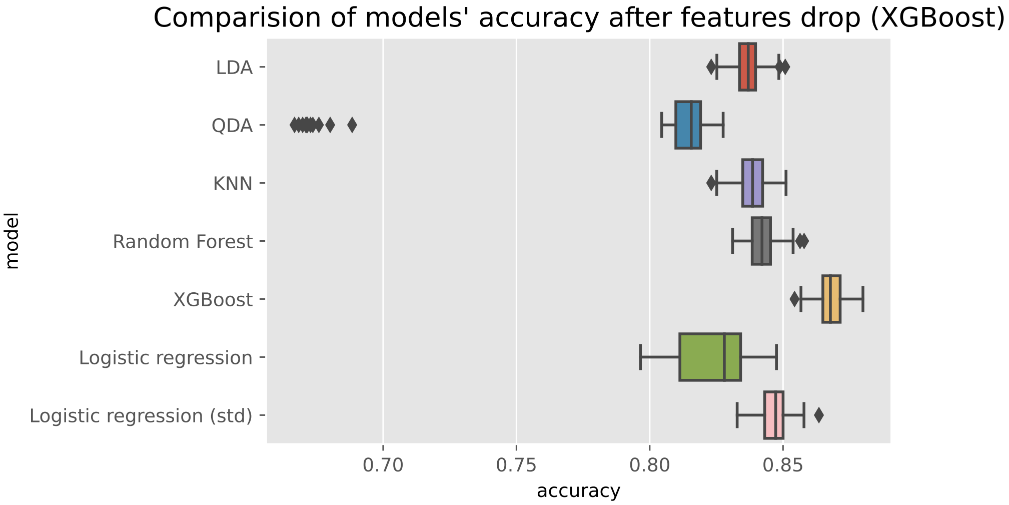

Figure 17: Boxplots of accuracy for considered models after dropping features: fnlwgt.

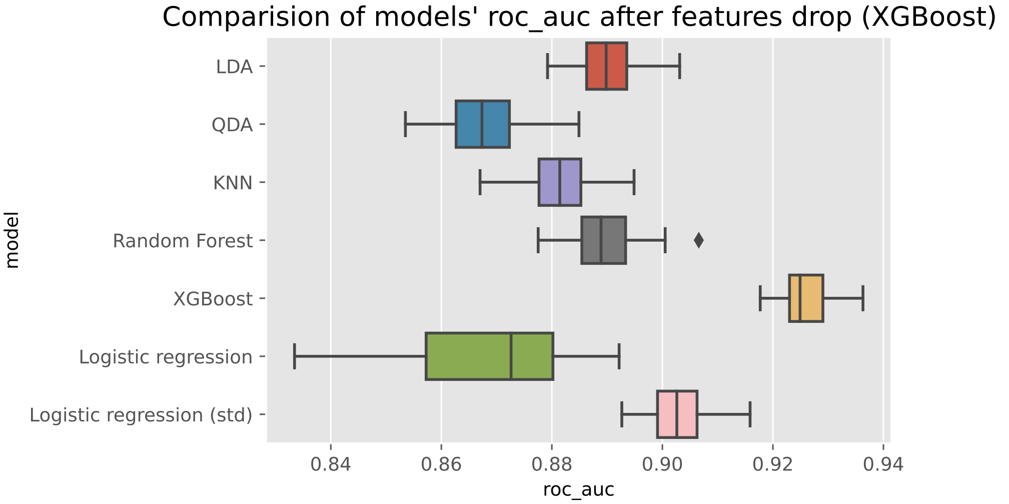

Figure 18: Boxplots of AUROC for considered models after dropping features: fnlwgt.

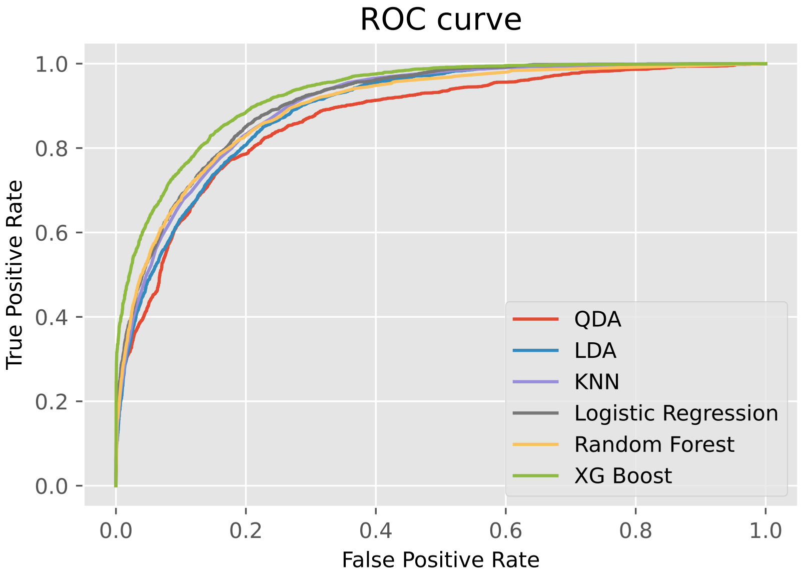

Figure 19: ROC curves of final models.

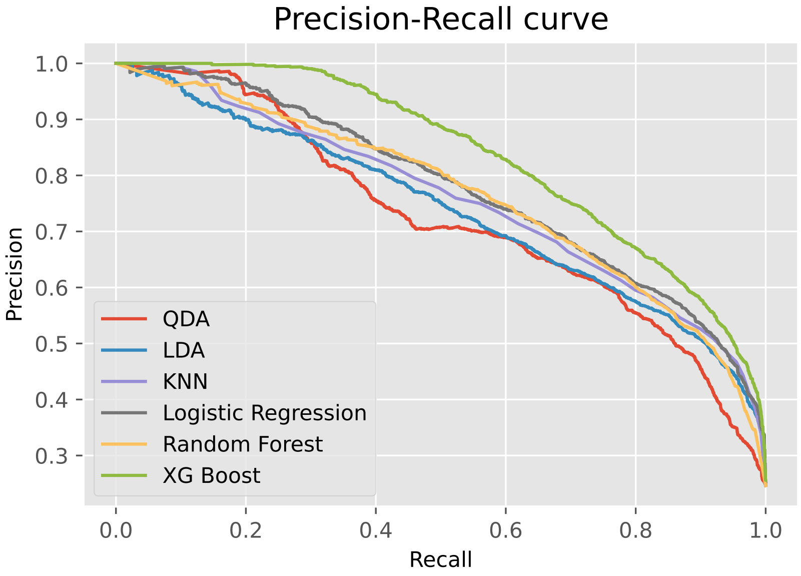

Figure 20: Precision-Recall curves of final models.

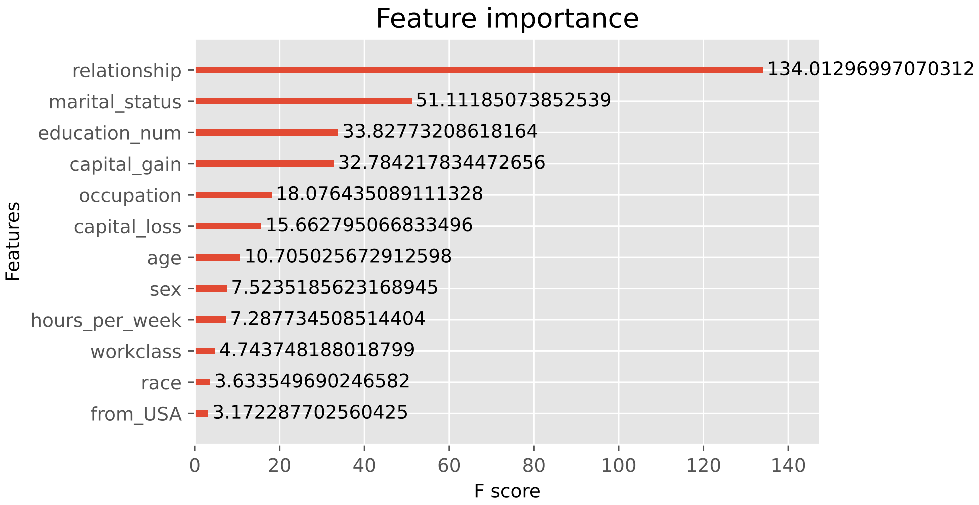

Figure 21: The average gain across all splits each feature in XGBoost model is used.