Kexin Chen (30205253), Yushang Lai (74082916), Zizhao Han(86049685)

CS 273A Project Report
Data set: Adult
Team member: Kexin Chen, Yushang Lai, Zizhao Han

1. Data Set Introduction

The Adult dataset we use aims to predict whether a person could make over 50K a year through fourteen features which respectively are: 1.Age; 2. Workclass; 3. Fnlwgt (Final Weight, which is the number of units in the target population that the responding unit represents); 4. Education level; 5. Total Years of Education; 6. Marital-status; 7. Occupation Categories 8. Relationship in family; 9. Race; 10. Gender; 11. Capital-gain (profit from investment) 12. Capital-loss 13. Hours per week for Work; 14. Native Country

The dataset contains 48842 instances (train=32561, test=16281) whose features are a mix of continuous and discrete values.

2. Data Preprocessing and Visualization

2.1 Impute Missing Values

At the beginning, we check for the missing values from the data both in training set and test set: Feature workclass and categories have around 6% missing value while feature native country has around two percent of that. Both removing and replacing by most frequent value methods are applied to this data set for the missing data; after several tests for model evaluation, we find that replacing by most frequent value method outperform removing the data for those .

	Training Missing number	Training Missing Percent	Test Missing number	Test Missing Percent
workclass	1836.0	5.64%	963.0	5.91%
occupation	1843.0	5.66%	966.0	5.93%
native-country	583.0	1.79%	274.0	1.68%

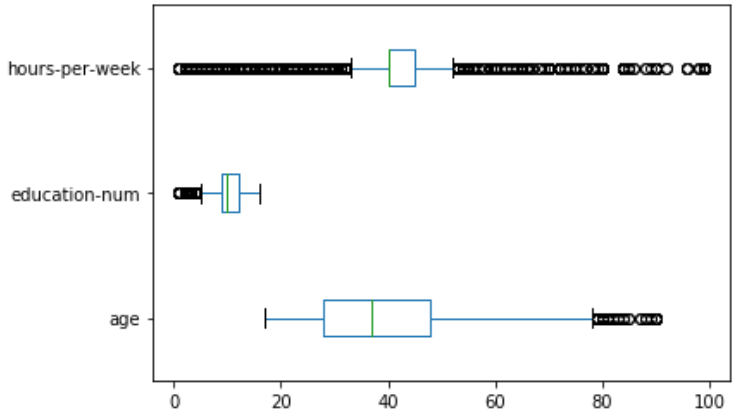
2.2 Feature Selection via Data Summary

After filling out the data, we transform gender to binary 0(female) and 1(male) as well as salary (0: salary <= 50k) and (1 : salary > 50k)make a summary for each continuous feature as the figure describes below. This table tells us that more than three quarters of people in this dataset don't have any investment gain or loss; and most of the people work around 40 hours per week. This motivates us to try whether our predictor performs better without those features.

	age	fnlwgt	education-num	sex	capital-gain	capital-loss	hours-per-week	salary
count	32561.000000	3.256100e+04	32561.000000	32561.000000	32561.000000	32561.000000	32561.000000	32561.000000
mean	38.581647	1.897784e+05	10.080679	0.330795	1077.648844	87.303830	40.437456	0.240810
std	13.640433	1.055500e+05	2.572720	0.470506	7385.292085	402.960219	12.347429	0.427581
min	17.000000	1.228500e+04	1.000000	0.000000	0.000000	0.000000	1.000000	0.000000
25%	28.000000	1.178270e+05	9.000000	0.000000	0.000000	0.000000	40.000000	0.000000
50%	37.000000	1.783560e+05	10.000000	0.000000	0.000000	0.000000	40.000000	0.000000
75%	48.000000	2.370510e+05	12.000000	1.000000	0.000000	0.000000	45.000000	0.000000
max	90.000000	1.484705e+06	16.000000	1.000000	99999.000000	4356.000000	99.000000	1.000000

2.3 Removing Outliers

From the bar plot on the right where the blue box represents the first quartile and third quartile; red line represents the median. We find that people's work hour per week follows a good distribution with high density in the middle and thin tail; while there are only a few (99) people older than 80 and a small number (219) people get education less than 3 years. Thus, we remove those people to seek for better performance on the data analysis. The remaining training data number is 32225 out of 32561

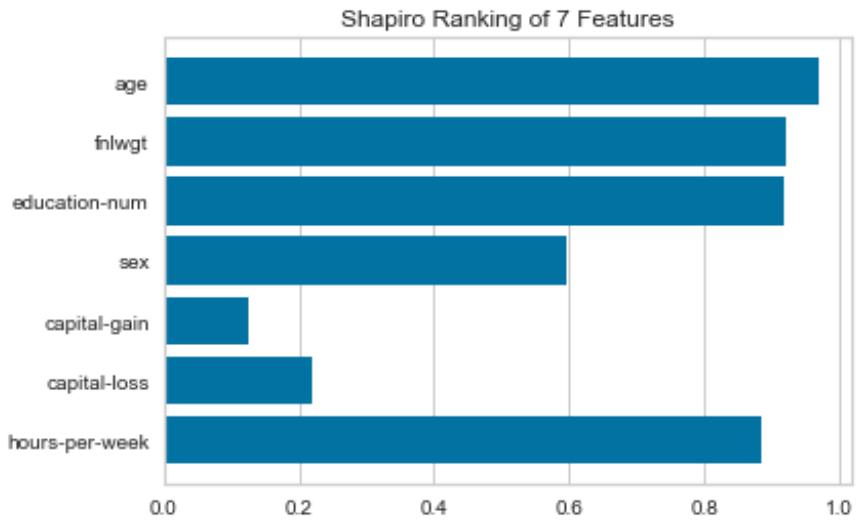


2.4 Feature Interactions

To visualize how important the features affect the prediction for a person's salary. We both capitalize on 1D shapiro algorithm and 2D covariance algorithm via the Yellow Brick library.

2.4.1 1D Shapiro Ranking algorithm

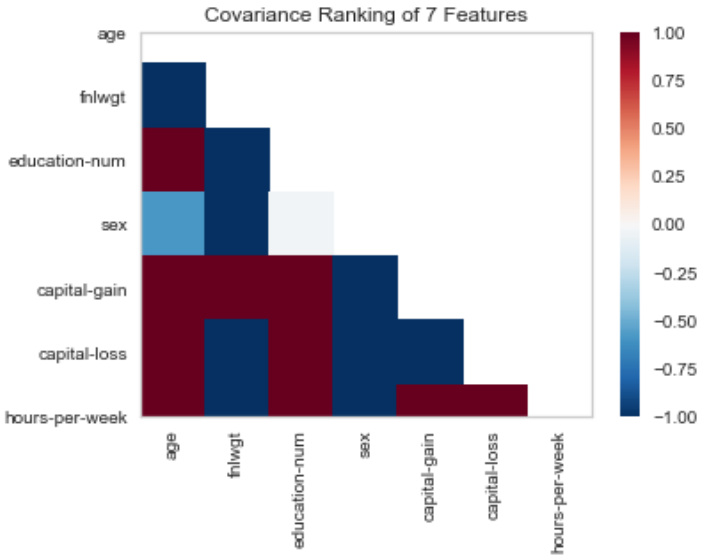
Shapiro algorithm has been used to determine the various aberrant splicing mechanisms in genes due to deleterious mutations in the splice sites, which cause numerous diseases. We regard each feature as a gene and whether salary is greater than 50k corresponds to whether a person has disease. From the figure on the left, we can see that age plays the most important role in prediction; while fnlwgt, time for getting education and work hours per week is also strongly related to our prediction. Sex has mild influence and investment gain/loss has very small impact on prediction. These future led us to try to predict without sex, capital gain/loss features. Besides, we can also try add nonlinear features such as age*age in our model.



2.4.2 2D Covariance Ranking algorithm

Apart from visualizing how each feature affects our prediction alone. We also want to understand how pairs of them work together on the prediction. The figure below confirms us that fnlwgt is strongly negative relates to all

the other features except capital gain; Year of education is positively related to investment and hours to work per week; Thus, we also want to test whether to treat interaction between those features as brand new features will help our prediction performance. For instance, add a new feature which multiplies people's education time and hours work per week.



2.5 Transform Categorical NLP Feature into Binary Type

Features including "Workclass", "Education", "Marital-status", "Occupation", "Relationship", "Race" and "Native-country" were represented as categorical NLP language. To convert them into numerical features such that regression could be applied to the data, we utilized one-hot-encoding, which expands the one categorical feature into multiple ones, with each new feature accounts for whether this instance belongs to a certain category in a certain parent feature, the contents of which are binary values. The figure on the left shows how many child features a parent feature split into. For instance, the previous feature name race will change into five new features which respectively are race_white, race_Assian, race_Amer, race_Black and race_other.

```
number of added columns (education): 16
number of added columns (workclass): 8
number of added columns (marital-status): 7
number of added columns (occupation): 14
number of added columns (relationship): 6
number of added columns (race): 5
number of added columns (native-country): 41
```

2.6 Feature Standardize

To prevent the results from potential collinearity, the last step we use the sklearn's preprocessing function to standardize features which are not binary or sparse. We standardize "Age", "Fnlwgt ", "Education-num" and "Hours-per-week" to improve the model's performance.

3. Model Selection

We use the following algorithms for this dataset: Random Forest, Multi-layer Perceptrons/ Neural Network, and AdaBoost. We vary the hyper-parameters of each model and evaluate the models based on two metrics: prediction accuracy, confusion matrix, and area-under-the-curve (AUC) of the receiver-operating-characteristic (ROC) curve. For some model, we use a five-fold cross-validation on the training split of the data to estimate the performance on the test split, and use the best performing hyper-parameter combination of each model to make predictions on the test data.

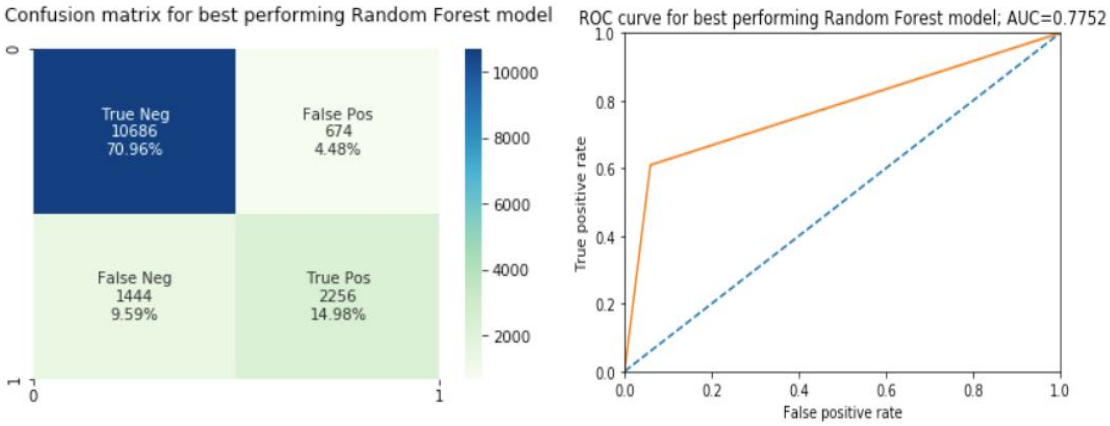
3.1 Random Forest Classifier

For the Random Forest algorithm, we search the parameter space through the following hyper-parameters: n_estimators (number of decision trees in the model), max_features (maximum number of features considered for a

splitting node), max_depth (maximum number of levels in each decision tree), and min_samples_leaf (minimum number of data points allowed in a leaf node). By assigning a range of random values to each hyper-parameter, we create a parameter grid which contains 660 combinations of different hyper-parameter values. The parameter values we search through are summarized in the table below:

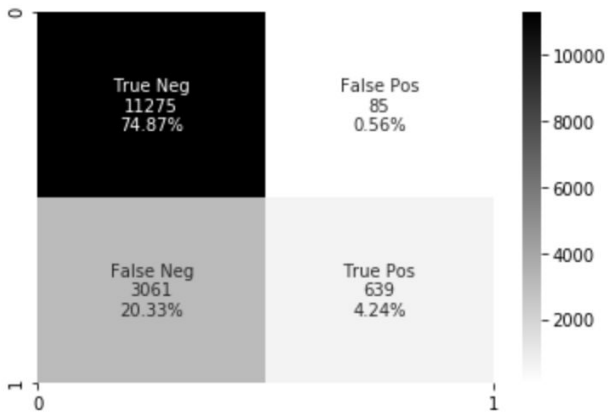
Parameter	Values
max_depth	[10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 110, None]
max_features	[‘auto’, ‘sqrt]
min_samples_leaf	[1, 2, 4]
n_estimators	[10, 31, 52, 73, 94, 115, 136, 157, 178, 200]

Through a grid search training, we identify that the best combination of hyper-parameters is {'n_estimators': 157, 'min_samples_leaf': 1, 'max_features': 'sqrt', 'max_depth': 20}. With the best hyper-parameters, we obtain an accuracy score of **0.9194** on the training set, and an accuracy score of **0.8594** on the test set. Since accuracy is sometimes misleading on the performance of a classifier, we also evaluate the model based on the confusion matrix and the ROC curve. From the following plots, we can see that the model makes pretty accurate predictions on the test data and produces relatively low Type I and Type II errors. The AUC-ROC value of **0.7752** also shows that the model is capable of distinguishing between the two classes in the data.

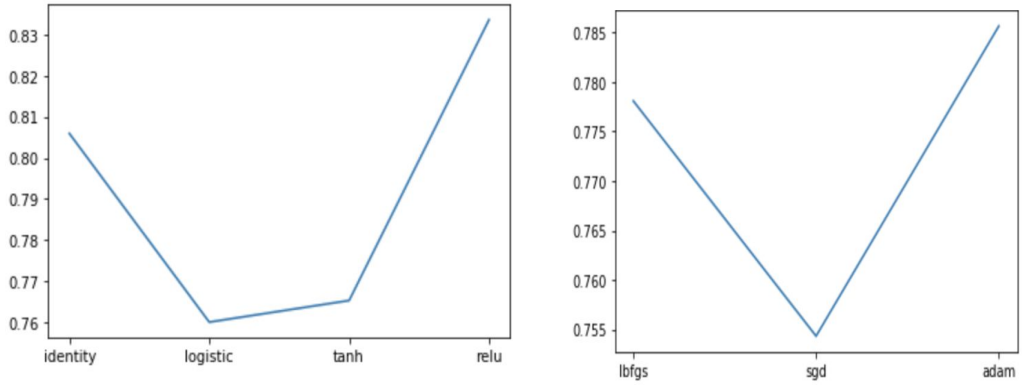


3.2 Neural Network

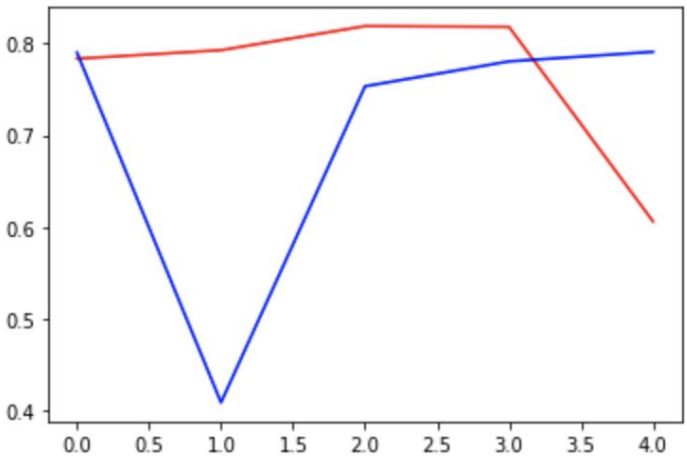
For Neural Network, we use the Multi-layer Perceptron classifier. To explore the effectiveness of this model, we first use two layers with 100 and 200 hidden units respectively, and keep the other parameters as default values given in the scikit-learn package. With the default parameters, we use Rectified Linear unit function as the activation function and the stochastic gradient-based optimizer. We also use L2 regularization with alpha = 0.0001. Here is the confusion matrix of the network model using the above parameters:



We also want to explore on different activation functions and different weight optimization methods. From the figure below we can see that Rectified Linear unit function and gradient-based optimizer have the best score for test data set.



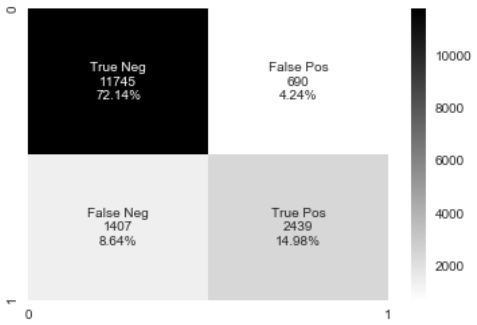
The next experiment involves changing the number of layers and hidden units per layer to measure the performance. We use five different number of nodes for single and double layers: (150,),(175,),(200,),(225,),(250,); and (150,150), (175,175), (200,200), (225,225), (250,250). As shown below, the red line represents single layer settings and the blue line represents double layers settings. Single layer with 225 hidden units has the best accuracy.

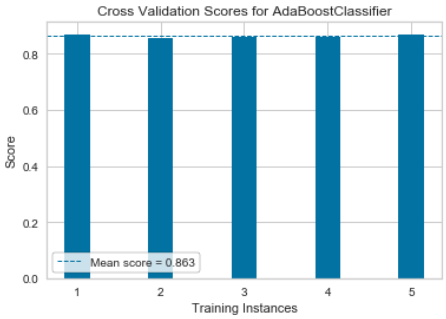


3.3. AdaBoost

Model Feature Selection	Best Accuracy	Number of Estimators
Linear: All Feature	**0.8706**	**701**
Linear: All Feature without Capital gain \ Capital lose	**0.8404**	**100**
Linear: All Feature without Work Class and Marital-status	**0.8686**	**667**
Linear: All Feature without Native country,race and hour per work	**0.8692**	**647**
Linear: All Feature Without Native country, race and hour per work **Nonlinear Interaction: Add feature: interaction hours per week *age**	**0.8712**	**730**

In this part we use adaptive boost methods in several combinations of models both linear and nonlinear; then, find the number of estimators for the largest the accuracy rate to avoid under/overfitting The best performance is the model with all linear features except Native country and race; besides, add a nonlinear feature hours per work multiplied by age. It gets the accuracy rate 0.8712 with total 730 estimators. The confusion matrix shows on the upright and the cross validation plot in downright.





4. Conclusion

In this project, we use the given adult dataset to predict whether a person can earn 50k dollars per year. Start with optimizing data processing by imputing missing data, analyzing feature value distribution and removing outliers. Then, we turn to visualizing the interaction of data via both 1D Shapiro ranking method and 2D covariance ranking method. By doing this, we design some linear and nonlinear combination of features. After transferring category features into numerical type and standardizing non-sparse features, we train our models by three kinds of learners which respectively are Random Forest with best test accuracy 0.8594, Neural Network with best accuracy 0.8214 as well as a nonlinear adaBoost model with test accuracy 0.8712. For each model we do cross-validation to assert the performance as well as tuning parameters to minimize under/overfitting as well as control the complexity (only some representative plots attached since the page limited).