PREDICT WHETHER INCOME EXCEEDS $50K/YR BASED ON CENSUS DATA

Binary classification using 5 different machine learning algorithms.

Full codes in:

https://github.com/Thaleia18/Data-income-classificationproblem

The prediction task is to determine whether a person makes over $50K a year.

I used five different classification algorithms:

- **Decision Tree Classifier**
- **Random Forest Classifier**
- **Logistic classifier**
- **SVM classifier**
- **K Neighbors Classifier**

I evaluated my predictions using different metrics:

- **Accuracy**
- **Precision**
- **Recall**
- **F1**
- **Area under precision recall**

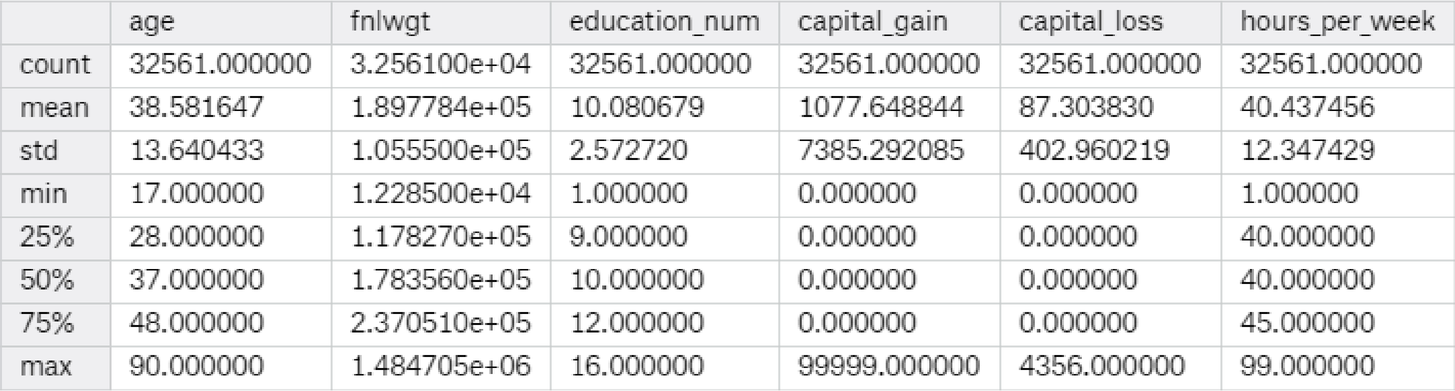
The data:

This data was extracted from the 1994 Census bureau database .

Attributes:

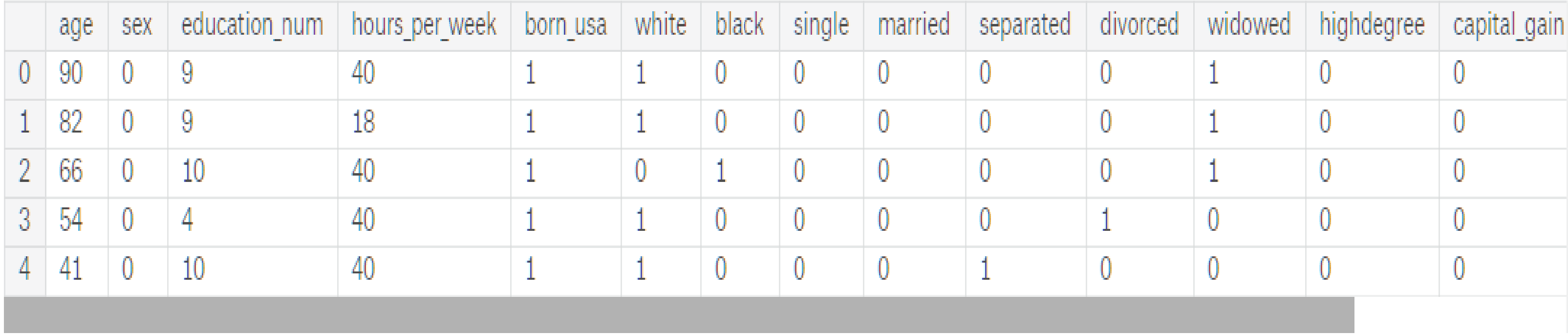
- >50K, <=50K
- age: continuous
- work class: Private, Self-emp-not-inc, Self-emp-inc, Federal-gov, …
- education: Bachelors, Some-college, Masters, Doctorate, 5th-6th, Preschool…
- education-num: continuous
- marital-status: Married-civ-spouse, Divorced, Never-married, Separated, Widowed, Married-spouse-absent, Married-AF-spouse
- occupation: Tech-support, Craft-repair, Machine-op-inspct, …
- relationship: Wife, Own-child, Husband, Not-in-family, Other-relative, ..
- race: White, Asian-Pac-Islander, Amer-Indian-Eskimo, Other, Black
- sex: Female, Male
- capital-gain: continuous
- capital-loss: continuous
- hours-per-week: continuous
- native-country: United-States, Cambodia, England, ..

- Sample of numerical data:



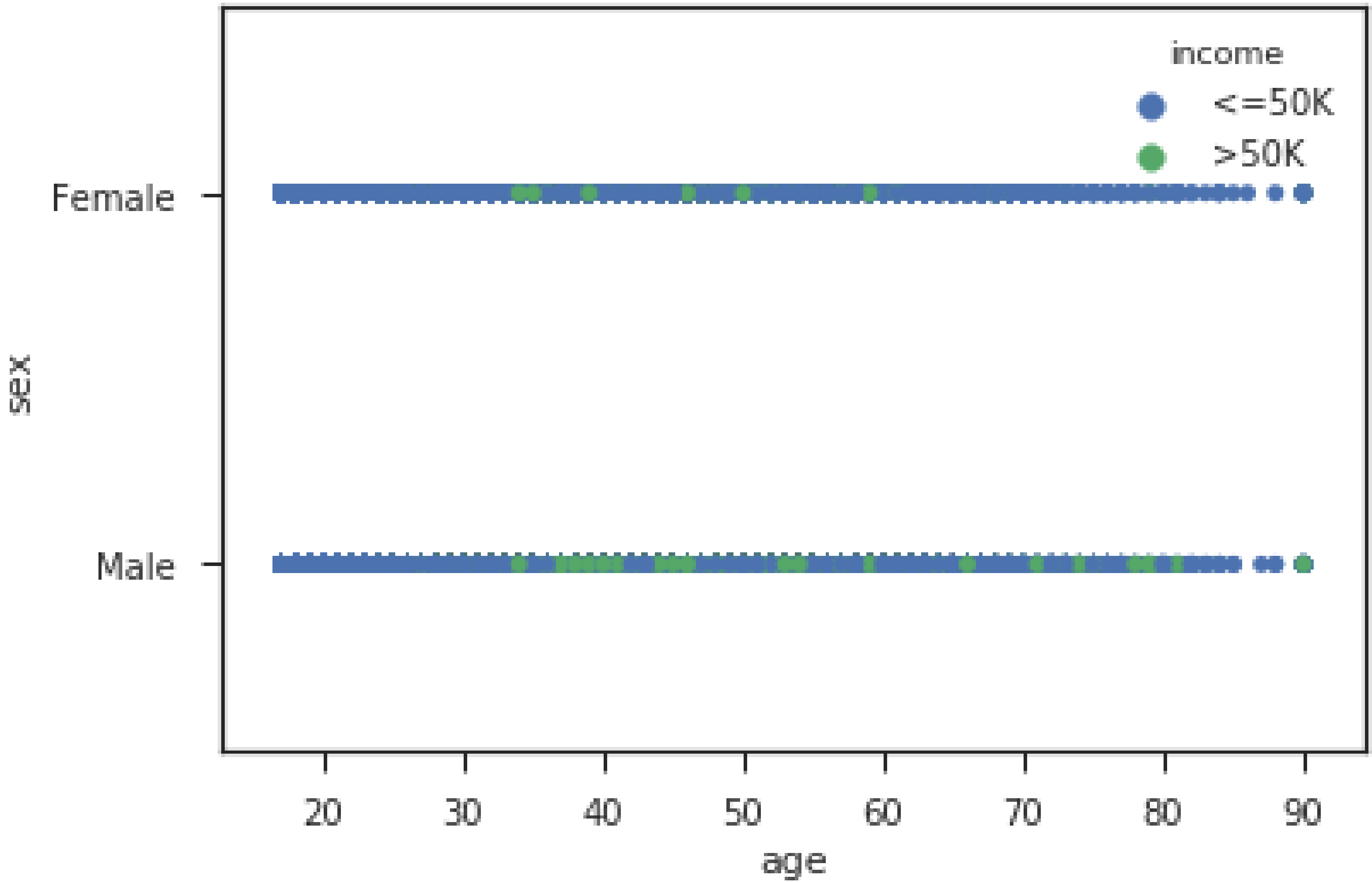
	age	fnlwgt	education_num	capital_gain	capital_loss	hours_per_week
count	32561.000000	3.256100e+04	32561.000000	32561.000000	32561.000000	32561.000000
mean	38.581647	1.897784e+05	10.080679	1077.648844	87.303830	40.437456
std	13.640433	1.055500e+05	2.572720	7385.292085	402.960219	12.347429
min	17.000000	1.228500e+04	1.000000	0.000000	0.000000	1.000000
25%	28.000000	1.178270e+05	9.000000	0.000000	0.000000	40.000000
50%	37.000000	1.783560e+05	10.000000	0.000000	0.000000	40.000000
75%	48.000000	2.370510e+05	12.000000	0.000000	0.000000	45.000000
max	90.000000	1.484705e+06	16.000000	99999.000000	4356.000000	99.000000

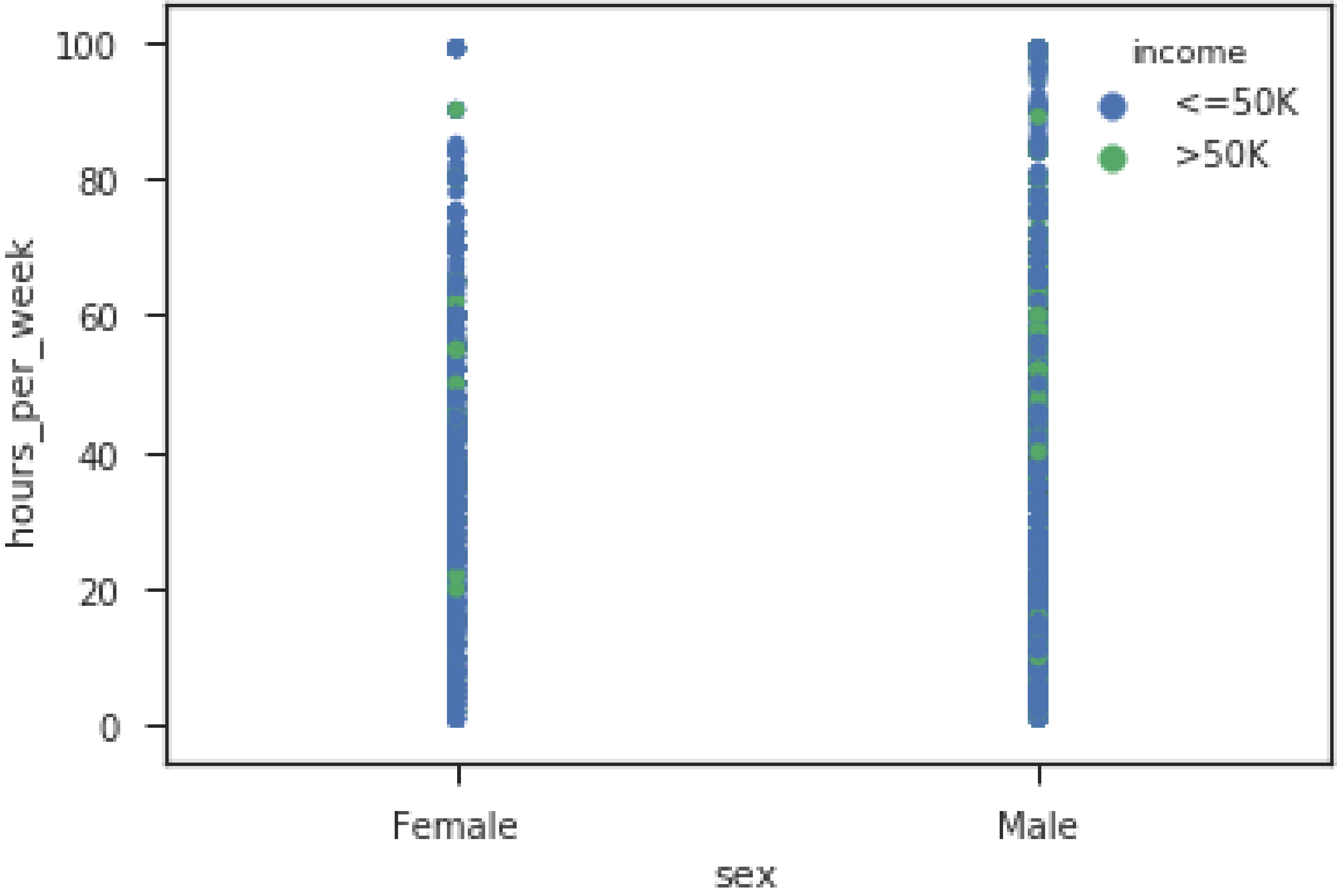
- After cleaning the data, remove null or repeated rows, and transform categorical attributes to numeric.

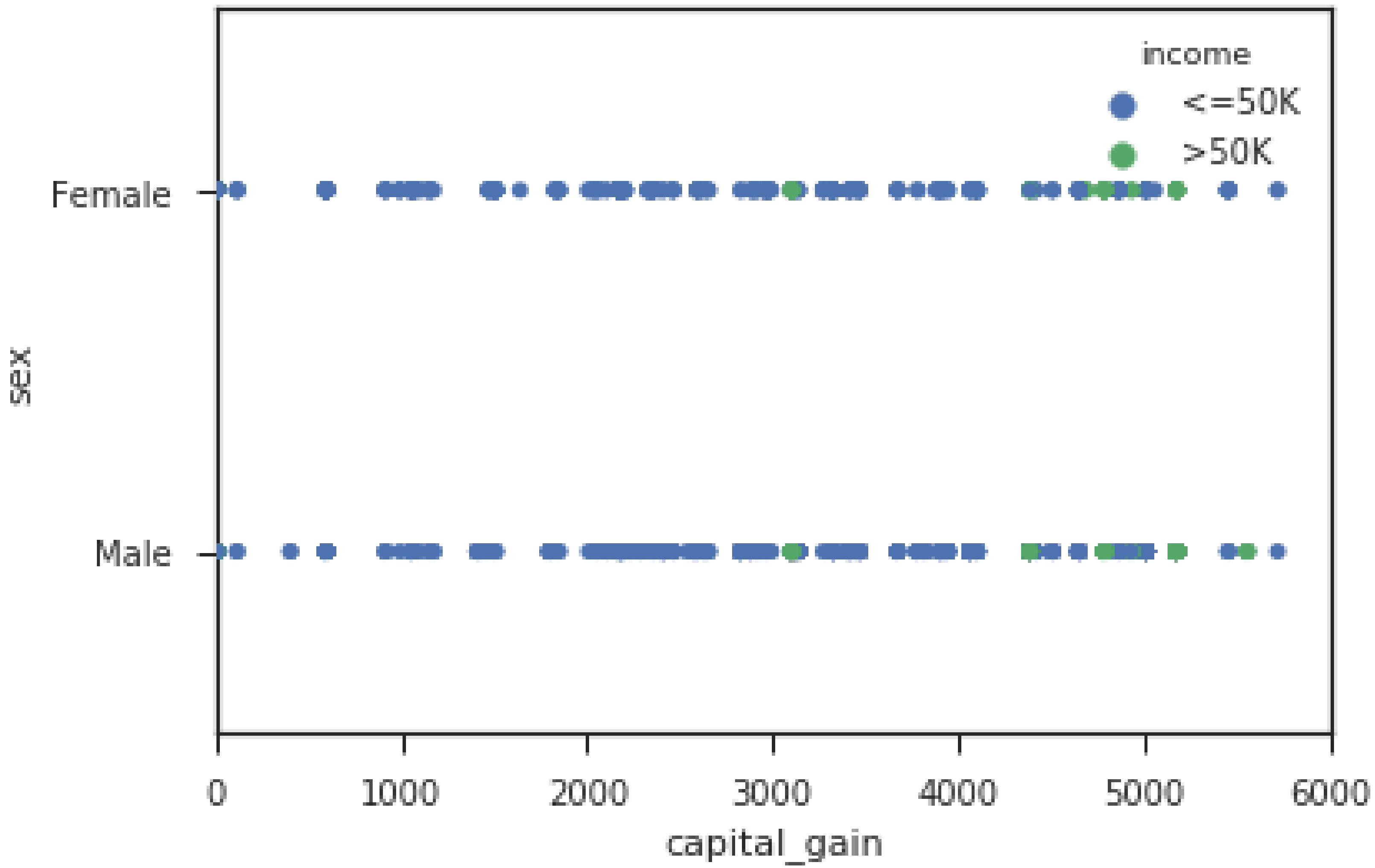


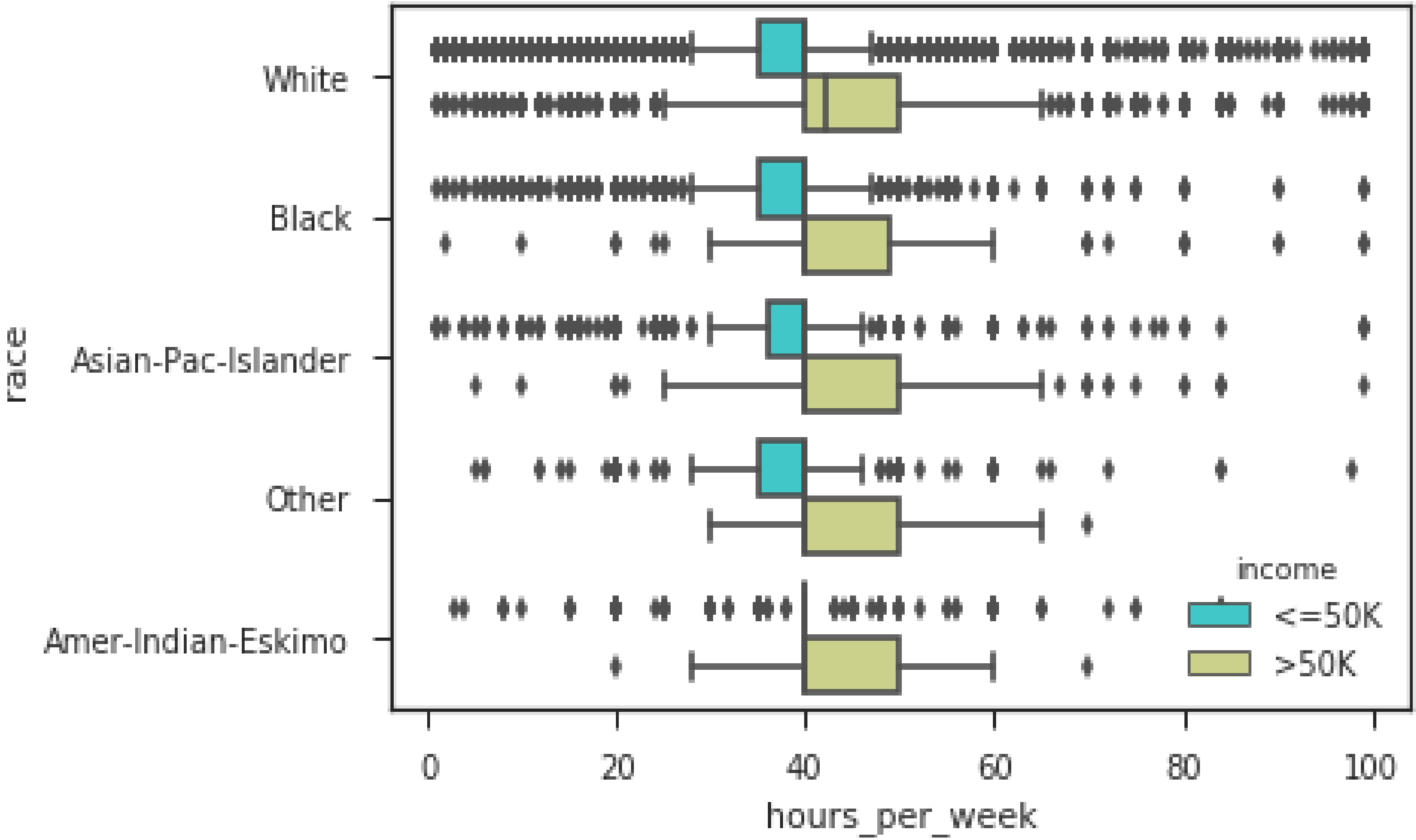
	age	sex	education_num	hours_per_week	born_usa	white	black	single	married	separated	divorced	widowed	highdegree	capital_gain
0	90	0	9	40	1	1	0	0	0	0	0	1	0	0
1	82	0	9	18	1	1	0	0	0	0	0	1	0	0
2	66	0	10	40	1	0	1	0	0	0	0	1	0	0
3	54	0	4	40	1	1	0	0	0	0	1	0	0	0
4	41	0	10	40	1	1	0	0	0	1	0	0	0	0

SOME DATA VISUALIZATIONS:

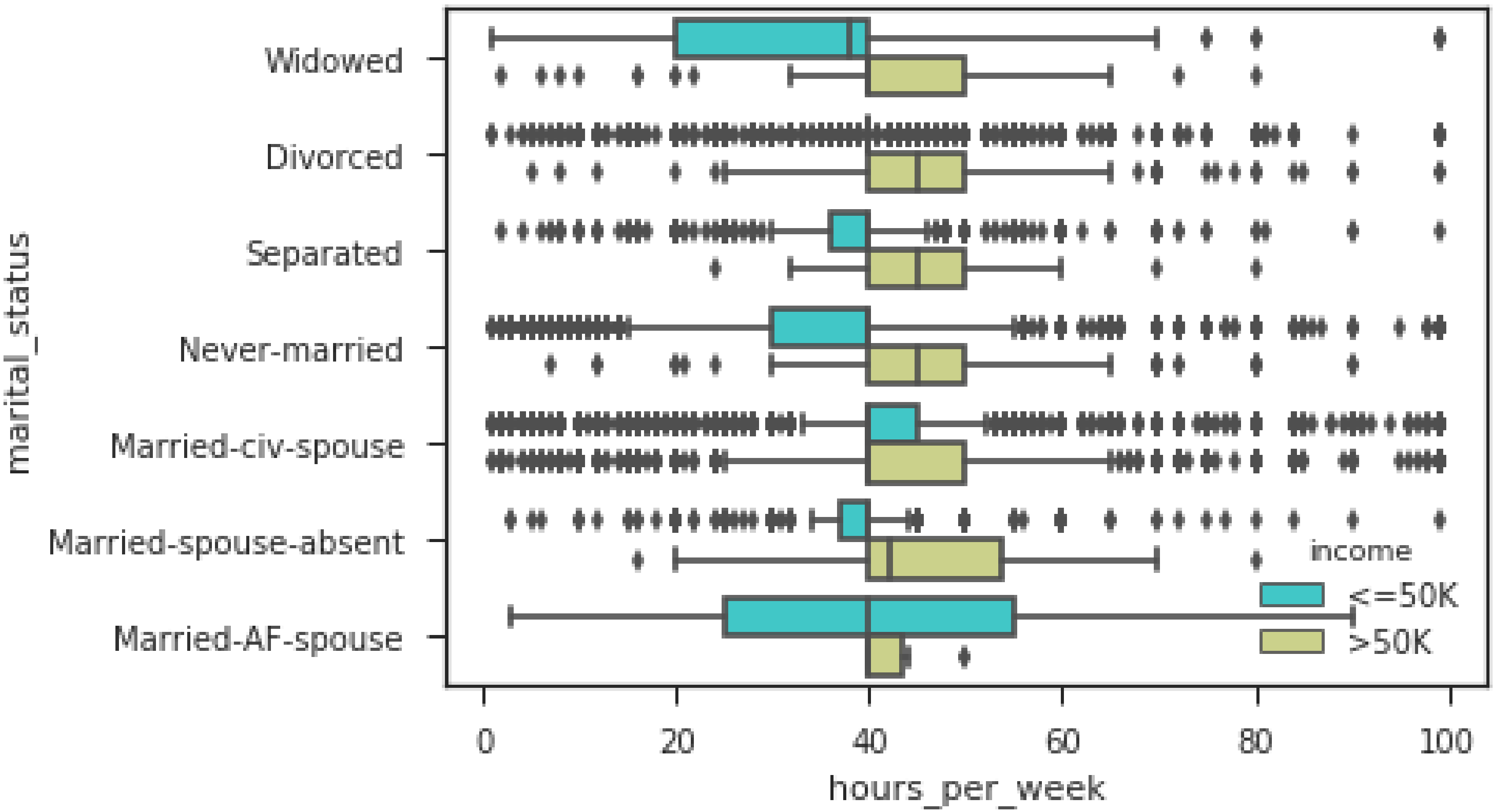


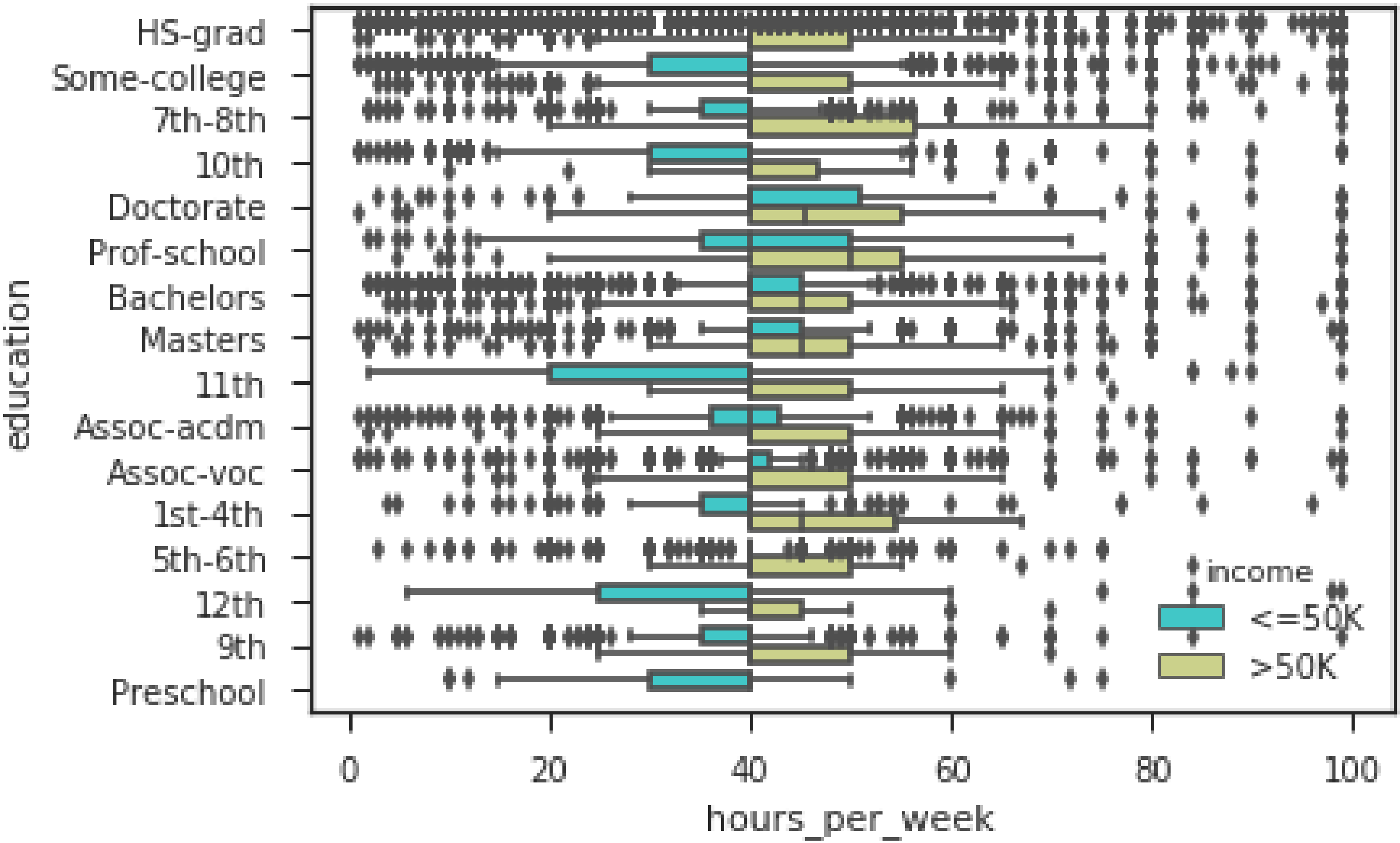

100
80
60
40
20
0
hours_per_week
Female
Male
sex
income
<=50K
>50K


income
<=50K
>50K
Female
Male
sex
0
1000
2000
3000
4000
5000
6000
capital_gain

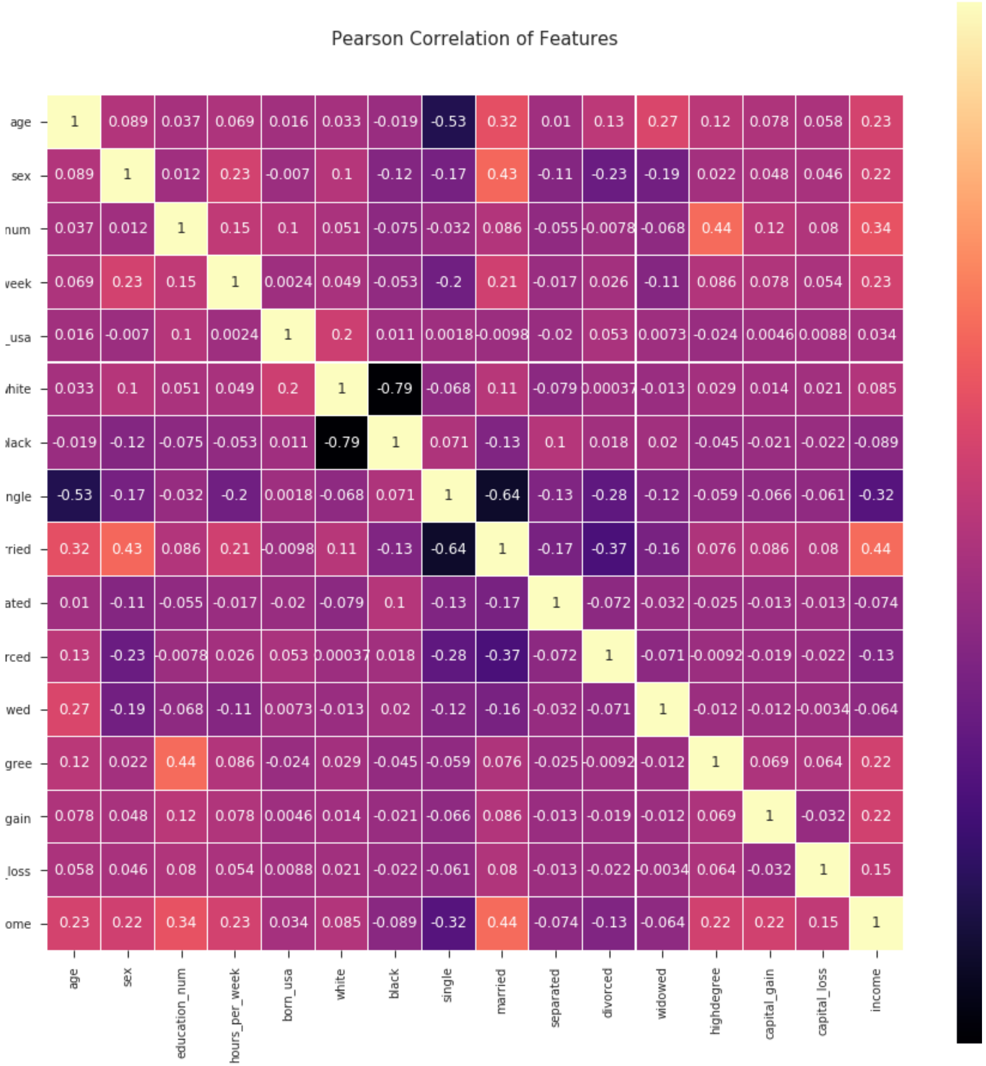


race
White
Black
Asian-Pac-Islander
Other
Amer-Indian-Eskimo
0
20
40
60
80
100
hours_per_week
income
<=50K
>50K


marital_status
Widowed
Divorced
Separated
Never-married
Married-civ-spouse
Married-spouse-absent
Married-AF-spouse
0
20
40
60
80
100
hours_per_week
income
<=50K
>50K


education
HS-grad
Some-college
7th-8th
10th
Doctorate
Prof-school
Bachelors
Masters
11th
Assoc-acdm
Assoc-voc
1st-4th
5th-6th
12th
9th
Preschool
0
20
40
60
80
100
hours_per_week
income
<=50K
>50K

I removed attributes that contain just minor categories. I kept attributes that have larrge categories, for example for race white and black are large categories and for native-country United States is the main caegory.

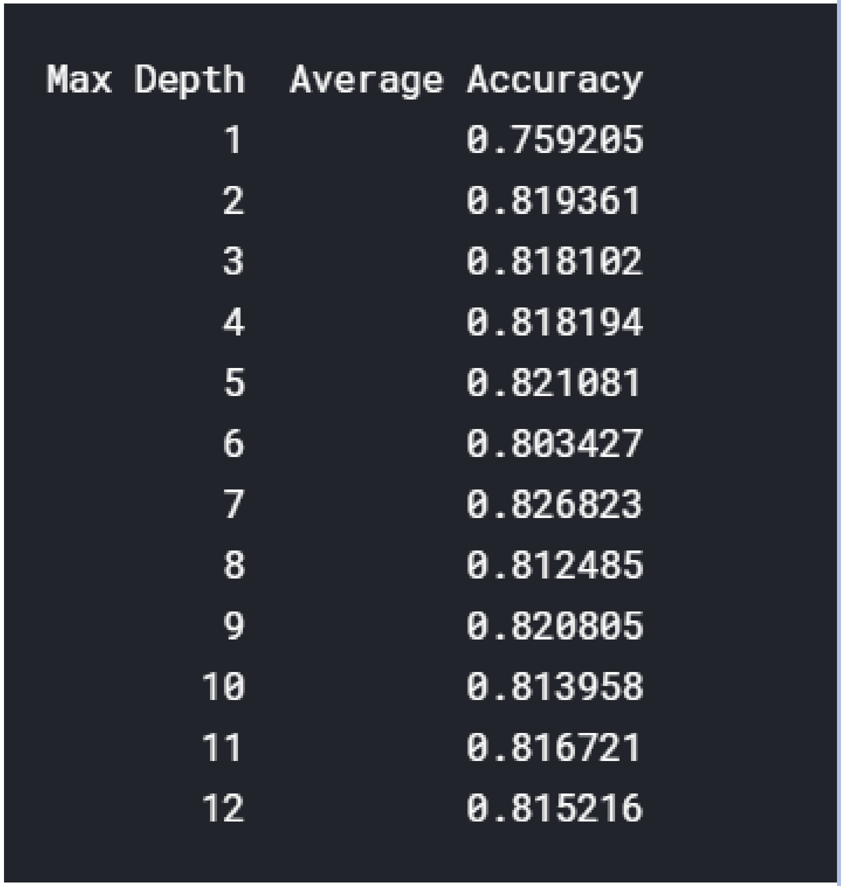


So my final attributes are: income, age, education-num, marital-status, sex, capital-gain, capital-loss, hours per week, native country.

MACHINE LEARNING MODELS:

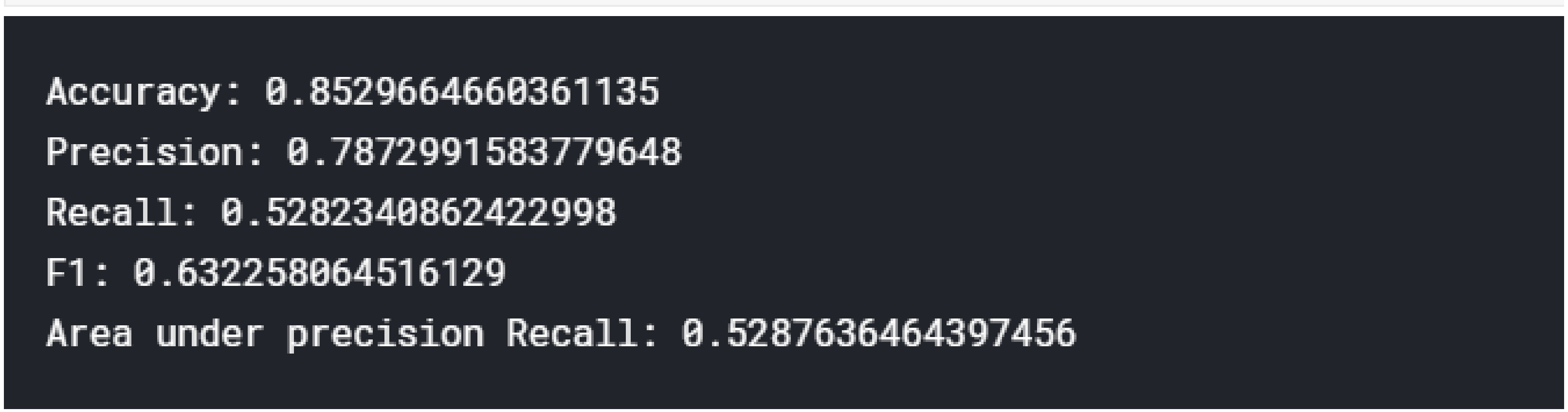
DECISION TREE:

First I looked for the depth that gives the best accuracy:



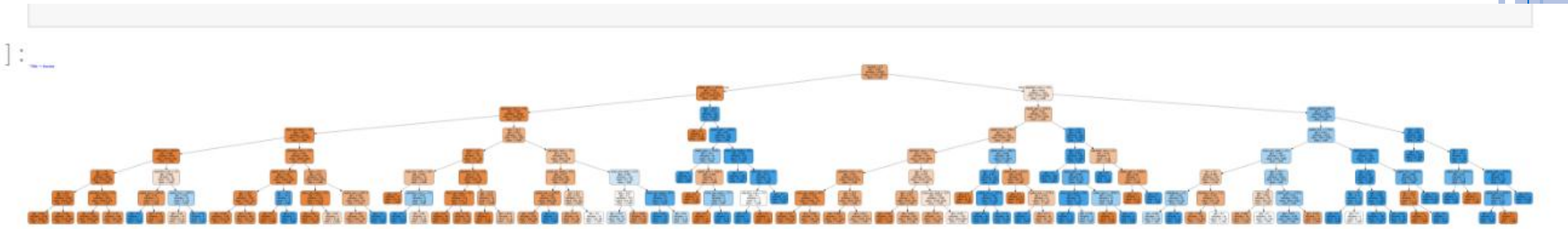
```
Max Depth  Average Accuracy
        1          0.759205
        2          0.819361
        3          0.818102
        4          0.818194
        5          0.821081
        6          0.803427
        7          0.826823
        8          0.812485
        9          0.820805
       10          0.813958
       11          0.816721
       12          0.815216
```

- The best depth was 7. The full metrics results for this depth is:

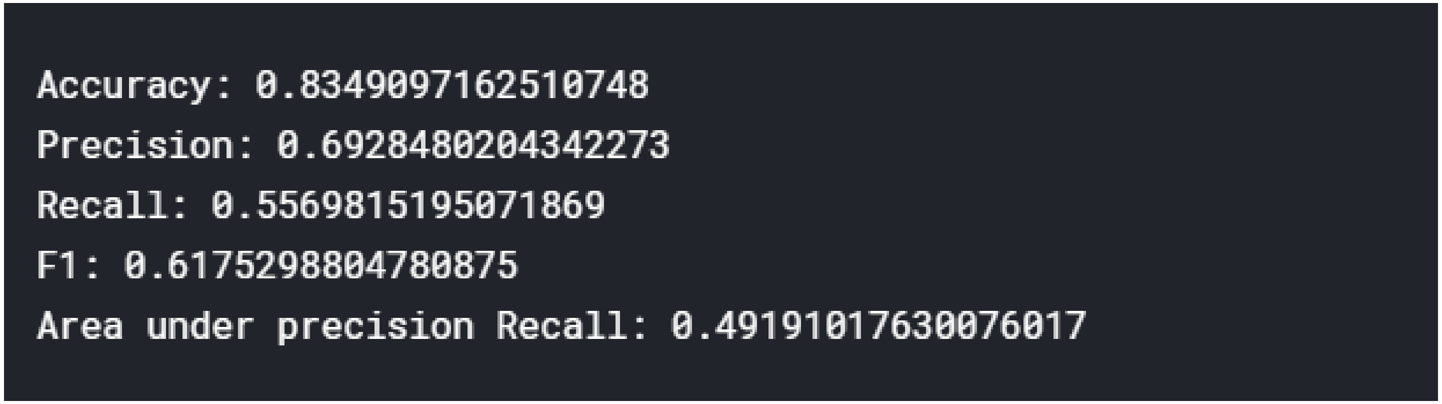


```
Accuracy: 0.8529664660361135
Precision: 0.7872991583779648
Recall: 0.5282340862422998
F1: 0.632258064516129
Area under precision Recall: 0.5287636464397456
```

- We can visualize our decision tree:

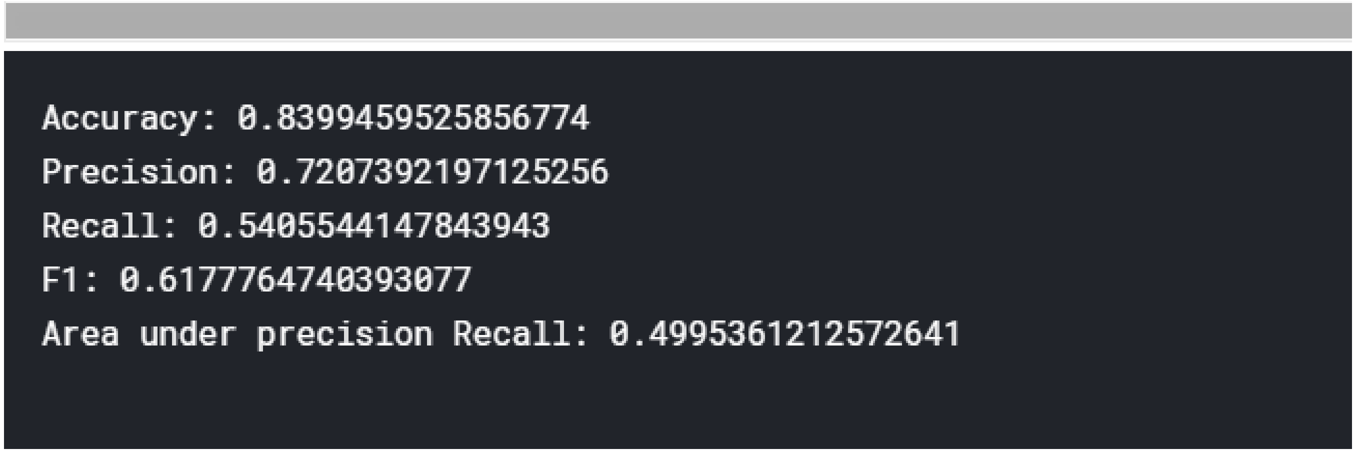


RANDOM FOREST:



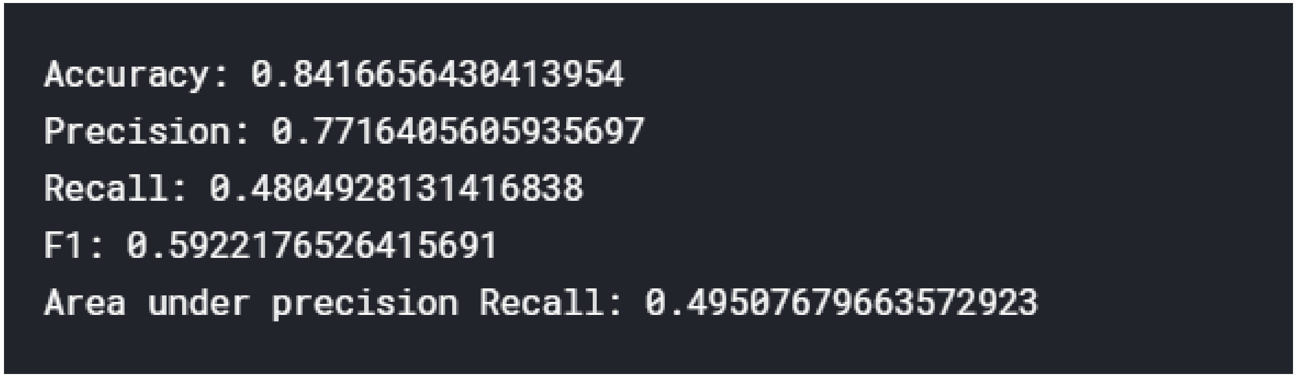
```
Accuracy: 0.8349097162510748
Precision: 0.6928480204342273
Recall: 0.5569815195071869
F1: 0.6175298804780875
Area under precision Recall: 0.49191017630076017
```

LOGISTIC REGRESSION:



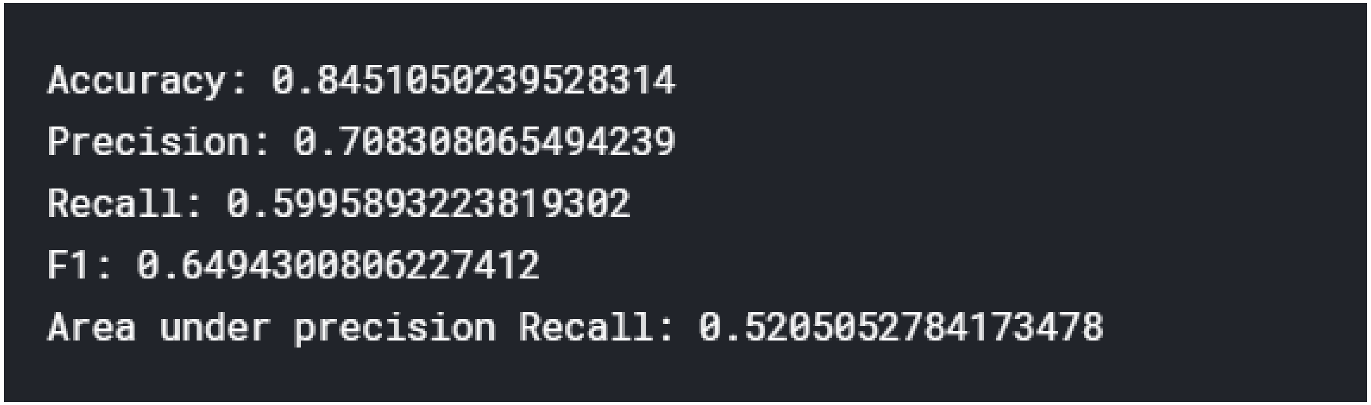
```
Accuracy: 0.8399459525856774
Precision: 0.7207392197125256
Recall: 0.5405544147843943
F1: 0.6177764740393077
Area under precision Recall: 0.4995361212572641
```

SVM CLASSIFIER:



```
Accuracy: 0.8416656430413954
Precision: 0.7716405605935697
Recall: 0.4804928131416838
F1: 0.5922176526415691
Area under precision Recall: 0.49507679663572923
```

K NEIGBORS CLASSIFIER:



```
Accuracy: 0.8451050239528314
Precision: 0.708308065494239
Recall: 0.5995893223819302
F1: 0.6494300806227412
Area under precision Recall: 0.5205052784173478
```

METRICS:

- **Accuracy** Fraction of predictions our model got right

  The best was the classification tree.
- **Precision** Proportion of those predicted positive, how many of them are actual positive.

  The best was the classification tree.
- **Recall** Proportion of the actual positive that were predicted correctly

  The best was the Kneighbors.
- **F1** Conveys the balance between the precision and the recall

  The best was the Kneighbors
- **Area under precision recall** The precision-recall curve shows the tradeoff between precision and recall for different threshold.

  The best was the classification tree.