EXAM PRESENTATION

Gavriliuc Tudor

01

02

03

04

INTRODUCTION

DATA ANALYSIS

DATA PREPROCESSING

DATA PROCESSING

Raw data

age	workclass	fnlwgt	education	education.	marital.sta	occupatior	relationshi	race	sex	capital.gai	capital.los	hours.per.	native.cou	income
90	?	77053	HS-grad	9	Widowed	?	Not-in-fan	White	Female	0	4356	40	United-Sta	<=50K
82	Private	132870	HS-grad	9	Widowed	Exec-mana	Not-in-fan	White	Female	0	4356	18	United-Sta	<=50K
66	?	186061	Some-colle	10	Widowed	?	Unmarried	Black	Female	0	4356	40	United-Sta	<=50K
54	Private	140359	7th-8th	4	Divorced	Machine-c	Unmarried	White	Female	0	3900	40	United-Sta	<=50K
41	Private	264663	Some-colle	10	Separated	Prof-speci	Own-child	White	Female	0	3900	40	United-Sta	<=50K
34	Private	216864	HS-grad	9	Divorced	Other-serv	Unmarried	White	Female	0	3770	45	United-Sta	<=50K
38	Private	150601	10th	6	Separated	Adm-cleric	Unmarried	White	Male	0	3770	40	United-Sta	<=50K
74	State-gov	88638	Doctorate	16	Never-mai	Prof-speci	Other-rela	White	Female	0	3683	20	United-Sta	>50K
68	Federal-go	422013	HS-grad	9	Divorced	Prof-speci	Not-in-fan	White	Female	0	3683	40	United-Sta	<=50K
41	Private	70037	Some-colle	10	Never-mai	Craft-repa	Unmarried	White	Male	0	3004	60	?	>50K
45	Private	172274	Doctorate	16	Divorced	Prof-speci	Unmarried	Black	Female	0	3004	35	United-Sta	>50K
38	Self-emp-r	164526	Prof-schoo	15	Never-mai	Prof-speci	Not-in-fan	White	Male	0	2824	45	United-Sta	>50K
52	Private	129177	Bachelors	13	Widowed	Other-serv	Not-in-fan	White	Female	0	2824	20	United-Sta	>50K
32	Private	136204	Masters	14	Separated	Exec-mana	Not-in-fan	White	Male	0	2824	55	United-Sta	>50K
51	?	172175	Doctorate	16	Never-mai	?	Not-in-fan	White	Male	0	2824	40	United-Sta	>50K
46	Private	45363	Prof-schoo	15	Divorced	Prof-speci	Not-in-fan	White	Male	0	2824	40	United-Sta	>50K
45	Private	172822	11th	7	Divorced	Transport-	Not-in-fan	White	Male	0	2824	76	United-Sta	>50K
57	Private	317847	Masters	14	Divorced	Exec-mana	Not-in-fan	White	Male	0	2824	50	United-Sta	>50K
22	Private	119592	Assoc-acdı	12	Never-mai	Handlers-c	Not-in-fan	Black	Male	0	2824	40	?	>50K
34	Private	203034	Bachelors	13	Separated	Sales	Not-in-fan	White	Male	0	2824	50	United-Sta	>50K
37	Private	188774	Bachelors	13	Never-mai	Exec-mana	Not-in-fan	White	Male	0	2824	40	United-Sta	>50K

Data analysis

General info

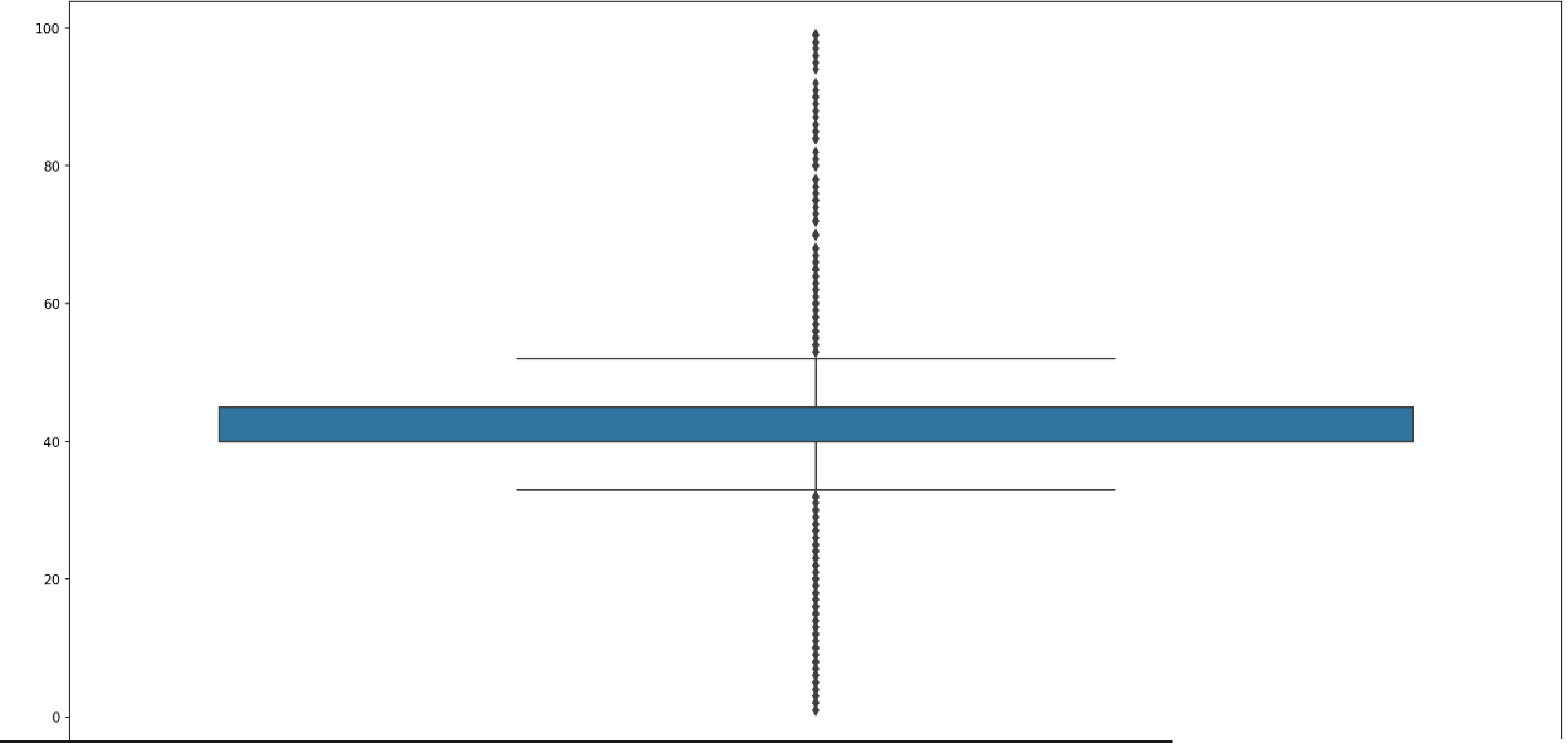
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 32561 entries, 0 to 32560
Data columns (total 15 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   age             32561 non-null  int64
 1   workclass       32561 non-null  object
 2   fnlwgt          32561 non-null  int64
 3   education       32561 non-null  object
 4   education.num   32561 non-null  int64
 5   marital.status  32561 non-null  object
 6   occupation      32561 non-null  object
 7   relationship    32561 non-null  object
 8   race            32561 non-null  object
 9   sex             32561 non-null  object
 10  capital.gain    32561 non-null  int64
 11  capital.loss    32561 non-null  int64
 12  hours.per.week  32561 non-null  int64
 13  native.country  32561 non-null  object
 14  income          32561 non-null  object
dtypes: int64(6), object(9)
memory usage: 3.7+ MB
```

```
df.describe()
```

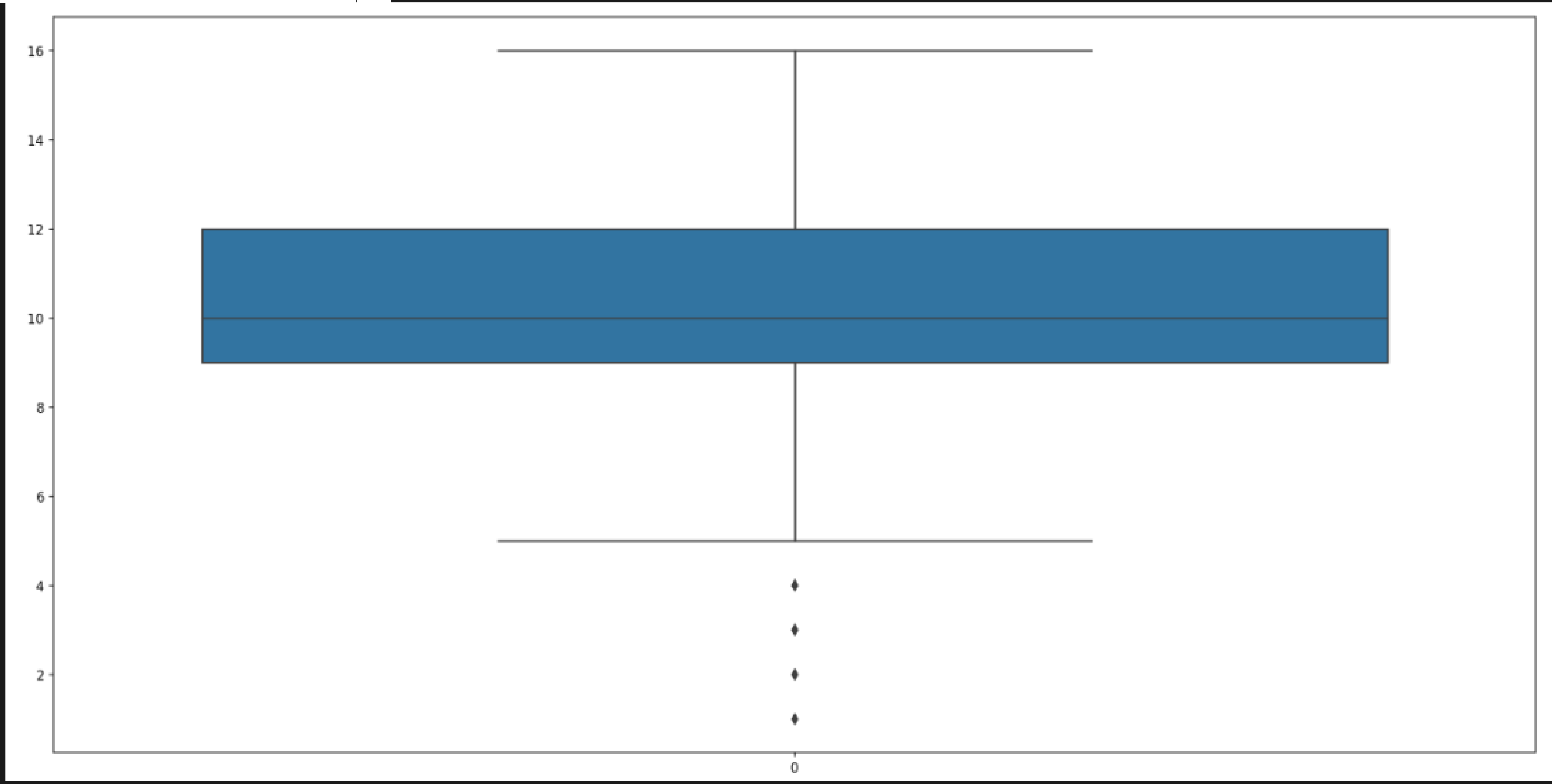
	age	fnlwgt	education.num	capital.gain	capital.loss	hours.per.week
count	32561.000000	3.256100e+04	32561.000000	32561.000000	32561.000000	32561.000000
mean	38.581647	1.897784e+05	10.080679	1077.648844	87.303830	40.437456
std	13.640433	1.055500e+05	2.572720	7385.292085	402.960219	12.347429
min	17.000000	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

Outliers Detection

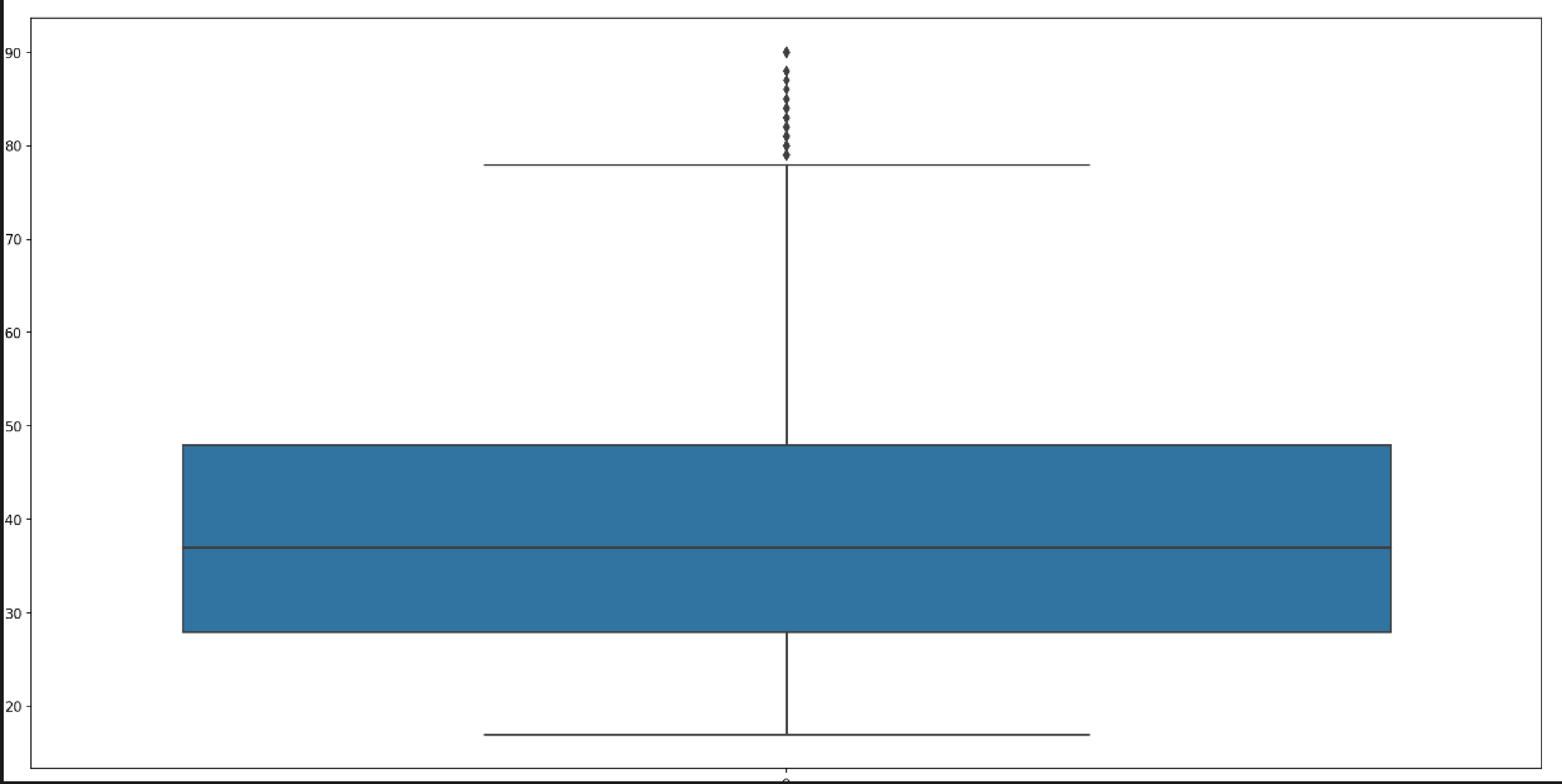


'hours.per.week'

'education.num'



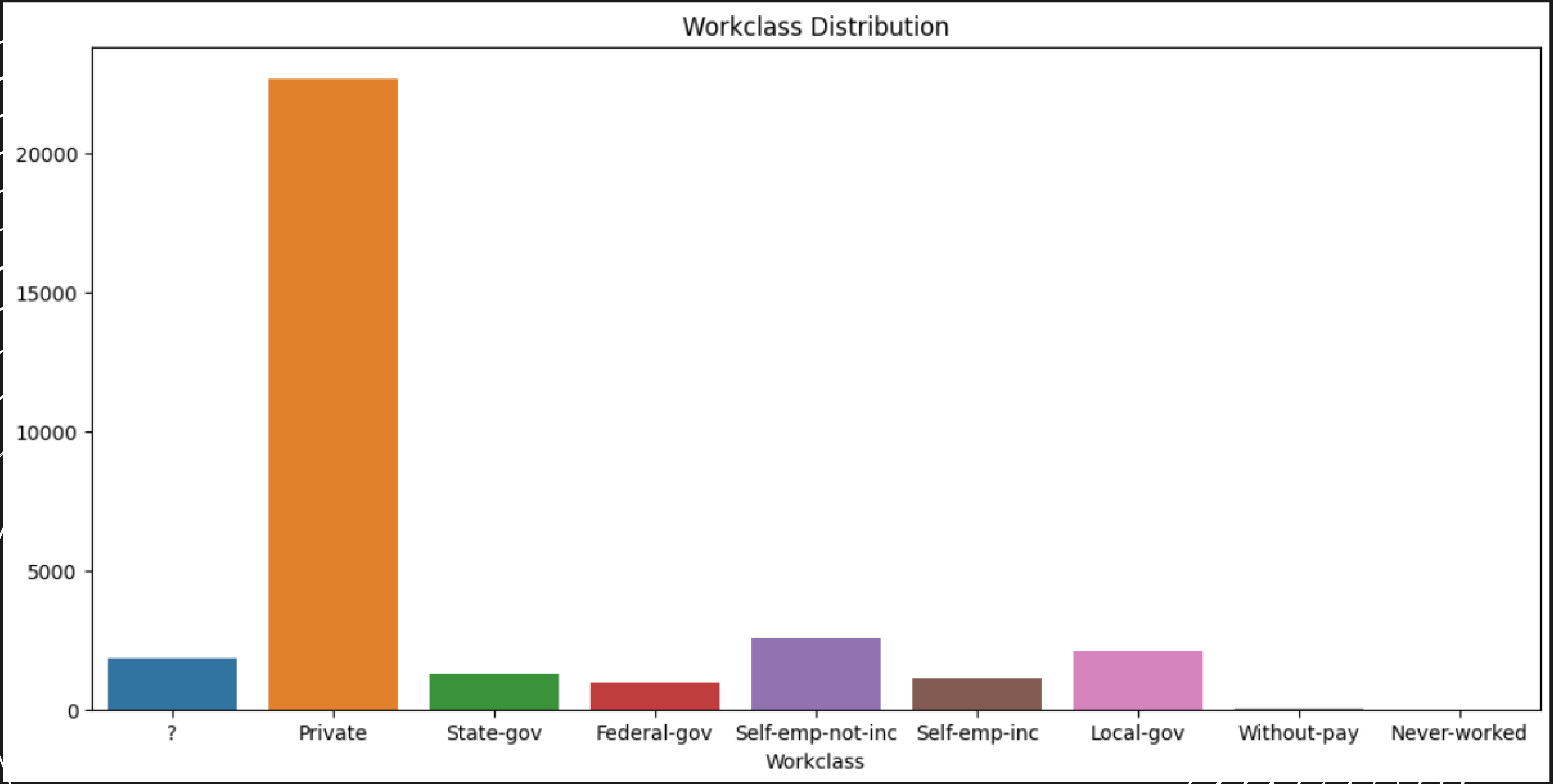
'age'

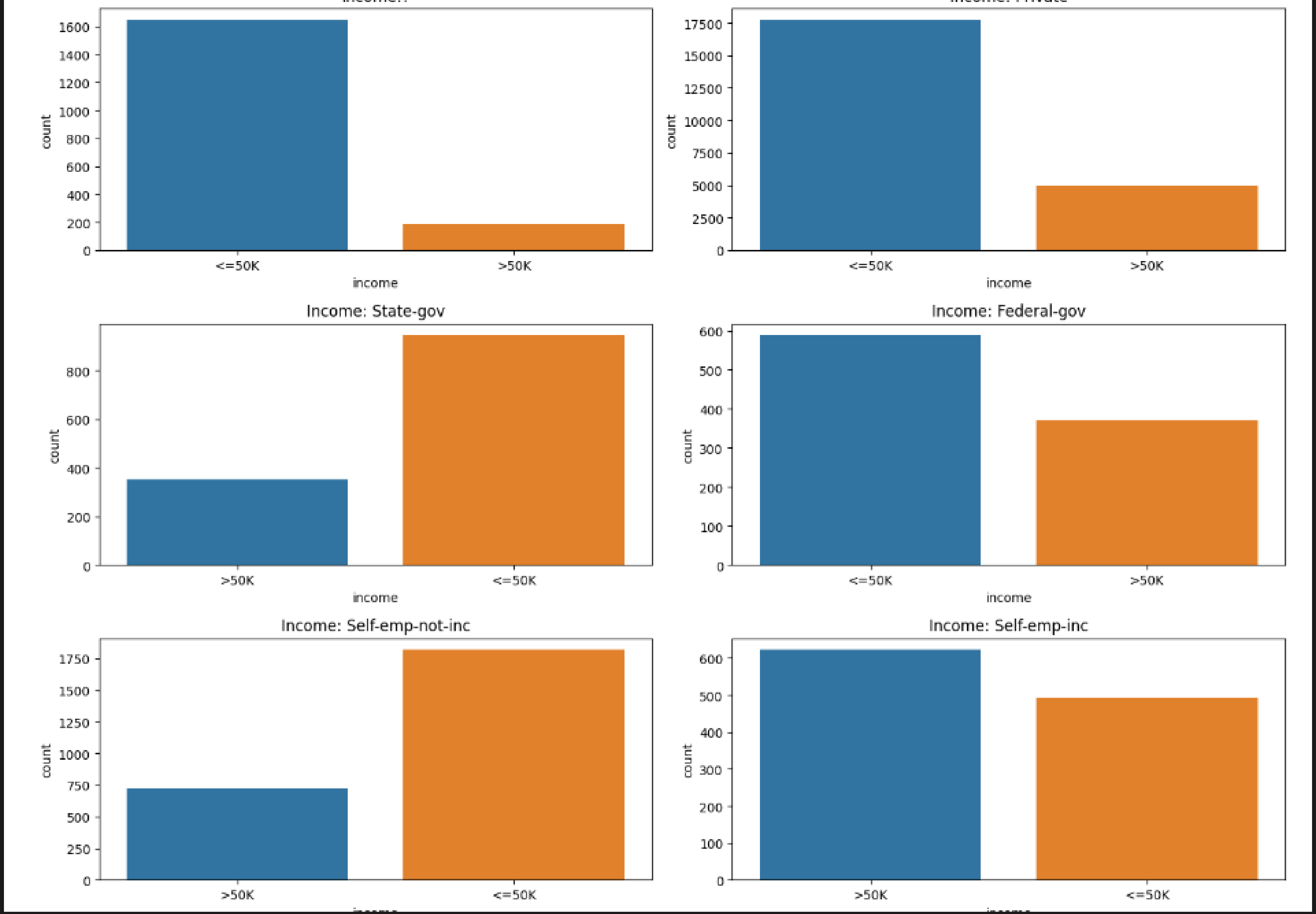


age - 0.43642622245443646
education.num - 3.6665949534376248
capital.gain - 8.335126164059378
capital.loss - 4.668531210621754
hours.per.week - 27.66696376432984

Interquartile range (IQR)

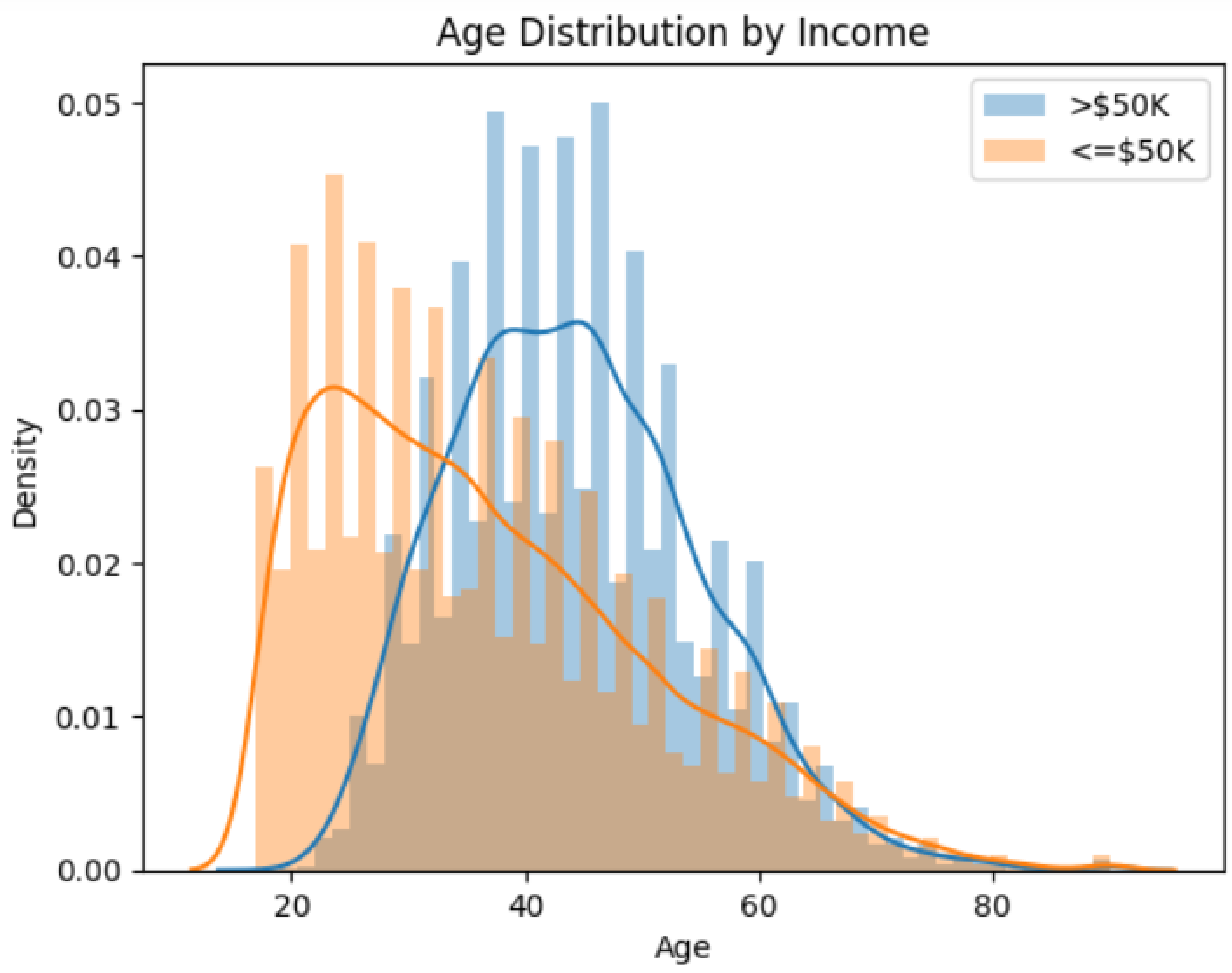
Workclass Distribution



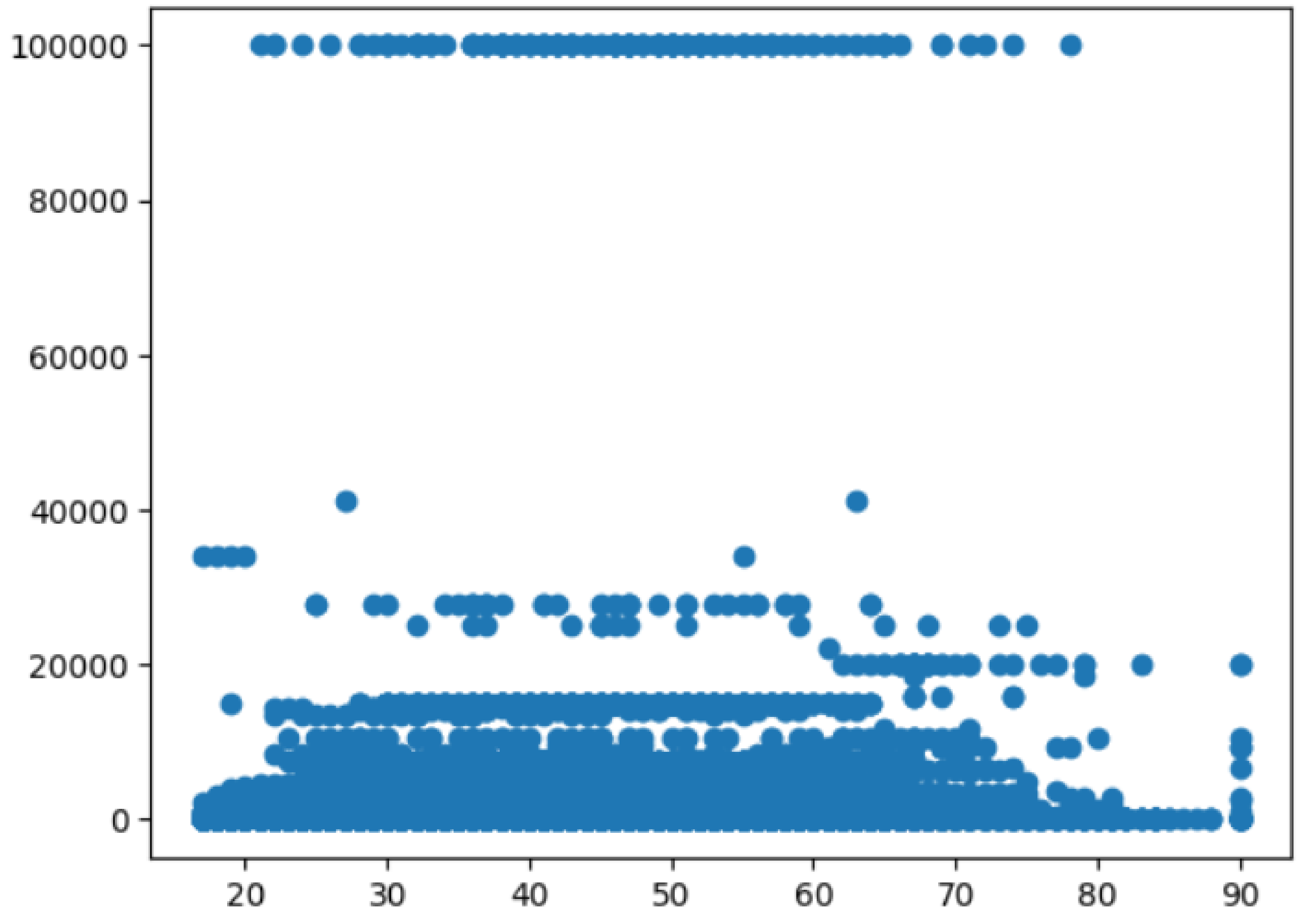


We can see that people working at Federal-gov and Self-emp-inc have a higher income comparative to other workclasses, and the private workclass is the most popular one.

How does age influence capital gain and income?

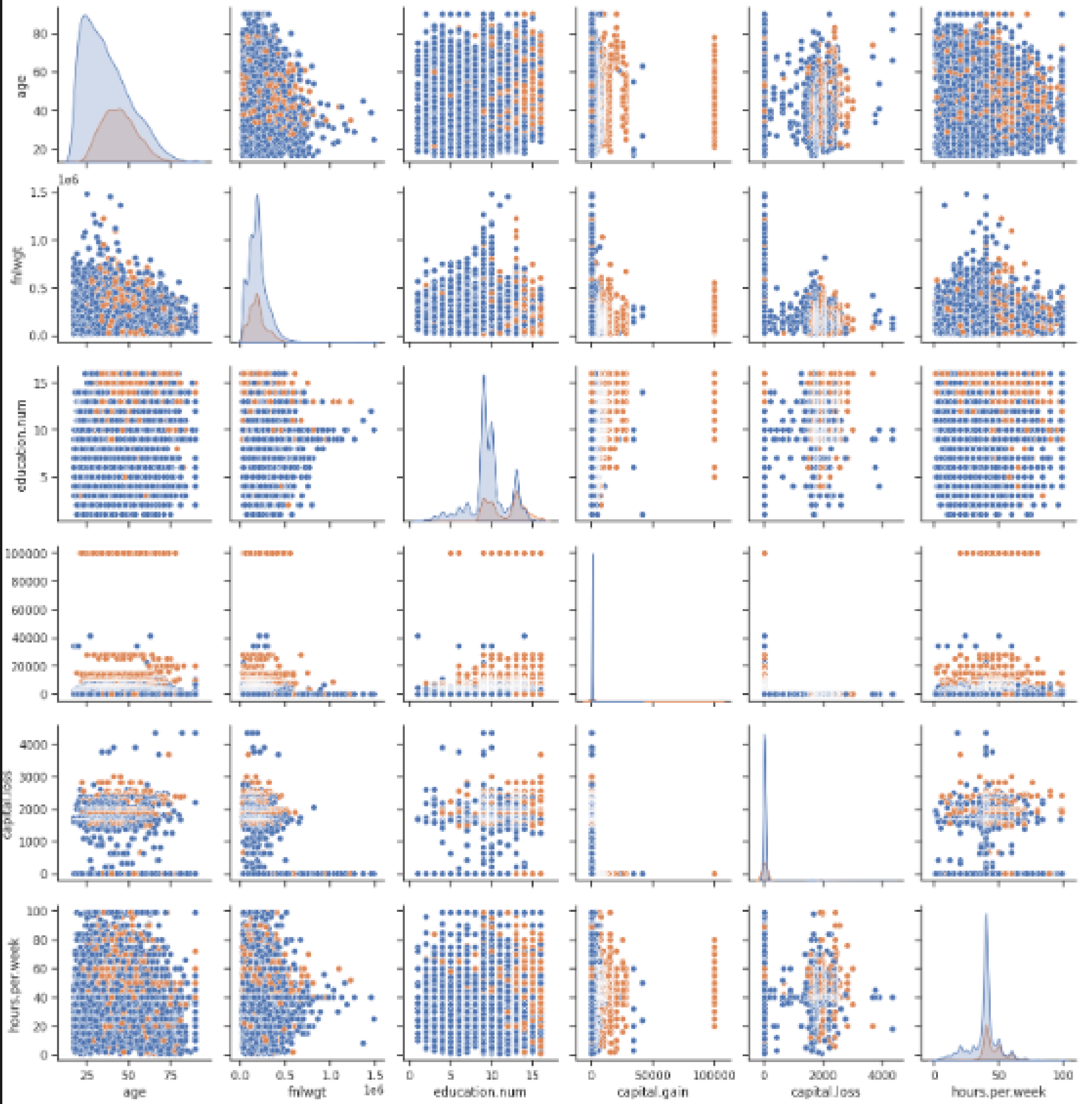


According to the plot above older people have a bigger income



Older people also reach a capital.gain but the values are not so high

We can notice that education.num has a linear relatioship with income, the same relation can be noticed with hours.per.week and age columns



Data Engineering

Analyzing the '?'

So far we can observe that the number of '?' in columns[occupation and workclass] is almost the same which means that we can not drop them since they have a valuable information [If X doesn't have a workclass results that it doesn't have an accupation]

Total number of '?' in df

```
k = 0
lista = len(df.columns)
for i in range(lista):
    for j in df[df.columns[i]]:
        if j == '?':
            k = k + 1
print(k)
```

```
4261
```

check the case when '?' is present in workclass and occupation

```
k = 0
for i in range(len(df)):
    if df.iloc[i,1]==df.iloc[i,6]=='?':
        k = k + 1
print(k)
```

```
1836
```

check the case when '?' is present in workclass and native.country

```
k = 0
for i in range(len(df)):
    if df.iloc[i,1]==df.iloc[i,14]=='?':
        k = k + 1
print(k)
```

```
0
```

check the case when '?' is present in native.country and occupation

```
k = 0
for i in range(len(df)):
    if df.iloc[i,6]==df.iloc[i,14]=='?':
        k = k + 1
print(k)
```

```
0
```

check the case when '?' is present in workclass , occupation and native.country

```
k = 0
for i in range(len(df)):
    if df.iloc[i,1]==df.iloc[i,14]==df.iloc[i,6]=='?':
        k = k + 1
print(k)
```

```
0
```

```
def replace_inter(df):
    df['workclass'] = df['workclass'].str.replace('?','unemployed')
    df['occupation'] = df['occupation'].str.replace('?','no_occupation')
    df['native.country'] = df['native.country'].str.replace('?','unknown')
    return df
df = replace_inter(df)
```

```
k = 0
for i in df.workclass:
    if i == '?':
        k = k + 1
print(k)
```

```
1836
```

Check the number of '?' in occupation

```
k = 0
for i in df.occupation:
    if i == '?':
        k = k + 1
print(k)
```

```
1843
```

Check the number of '?' in native.country

```
k = 0
for i in df['native.country']:
    if i == '?':
        k = k + 1
print(k)
```

```
582
```

Categorical features

Binary_encoder

```
def binary_encoder(df,column_list):
    for i in column_list:
        rep = len(df[i].unique())
        if rep > 3:
            li1.append(i)
    encoder = ce.BinaryEncoder(cols=li1, return_df = True)
    df = encoder.fit_transform(df)
    return df

df = binary_encoder(df,lista)
```

Map function

```
def map_binar(df):
    df['income'] = df['income'].map({'<=50K' : 0, '>50K' : 1})
    df['sex'] = df['sex'].map({'Female' : 0, 'Male' : 1})
    return df
df = map_binar(df)
```

Class balancing

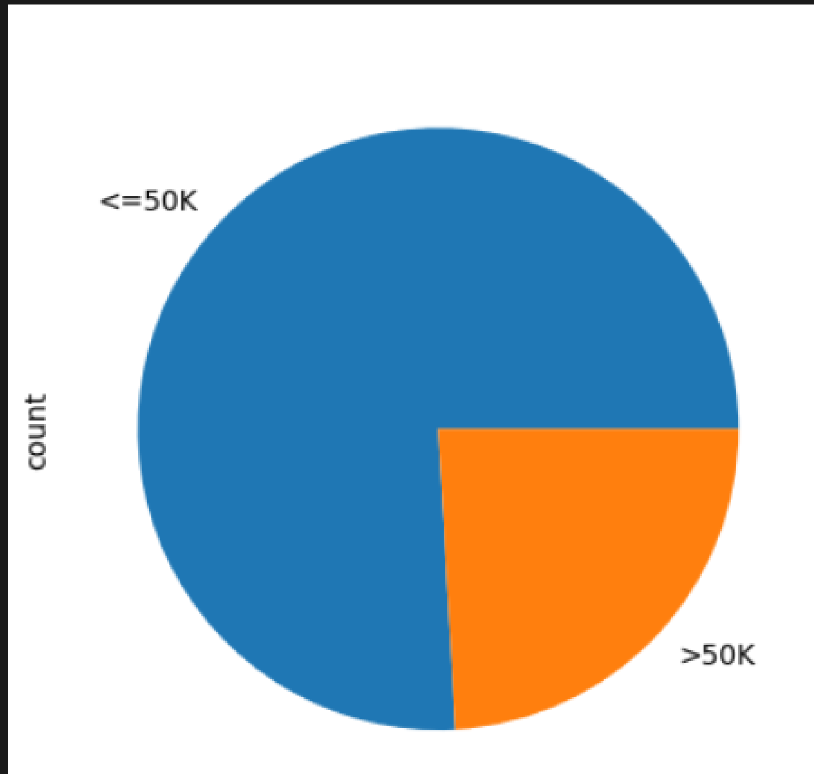
SMOTENC

```
from imblearn.over_sampling import SMOTENC
from sklearn.model_selection import train_test_split
```

```
def get_smotenc(df,target,num):
    X = df.drop(target, axis=1)
    Y = df[target]
    X_train, X_test, y_train, y_test = train_test_split(X,Y, random_state = 12, test_size=0.25)
    sm = SMOTENC(random_state=42, categorical_features=[num])
    X_res, y_ress = sm.fit_resample(X_train,y_train)
    y_ress.value_counts().plot.pie()
    df_balanced = pd.concat([X_res, y_ress], axis = 1)
    return df_balanced
df = get_smotenc(df,'income',34)
df
```

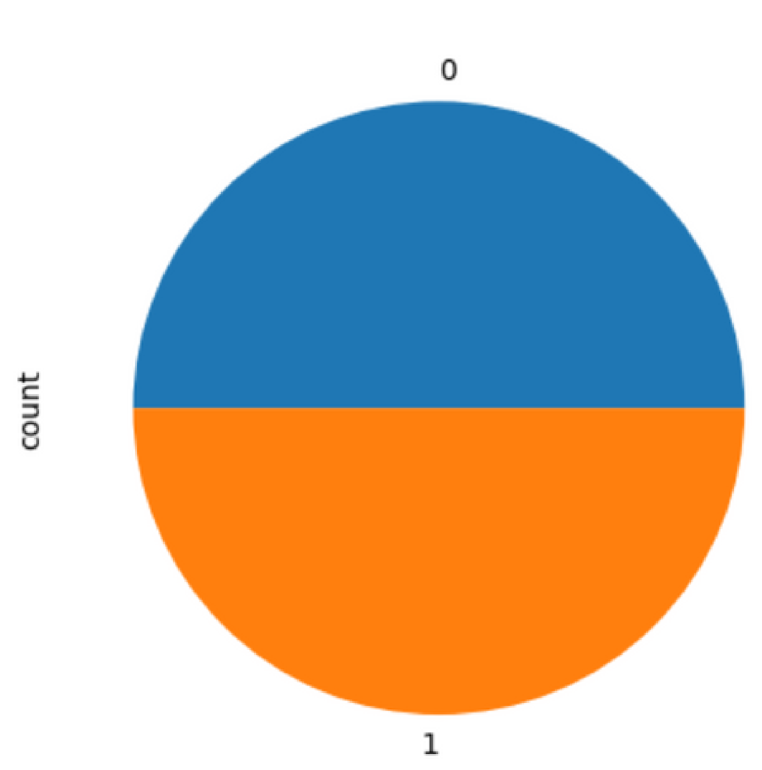
Before

32537



After

37186



Outliers

IsolationForest
EllipticEnvelope
LocalOutlierFactor

```
age - 0.47047492035267097
education.num - 0.048158850114840335
capital.gain - 13.2807290509002
capital.loss - 6.271764095724976
hours.per.week - 9.36504408387049
```

```
age - 0.7111482953357038
education.num - 0.40039441838228707
capital.gain - 12.872381749185765
capital.loss - 0.020916126333403055
hours.per.week - 16.89425404129441
```

```
age - 0.6837002328117172
education.num - 0.4449945480800401
capital.gain - 10.897945952317803
capital.loss - 5.160168567471193
hours.per.week - 12.775174608787907
```

According to the result above IsolationForest algorithm has the best result since it is capable to identify outliers in non-linear data distributions.

Feature Selection

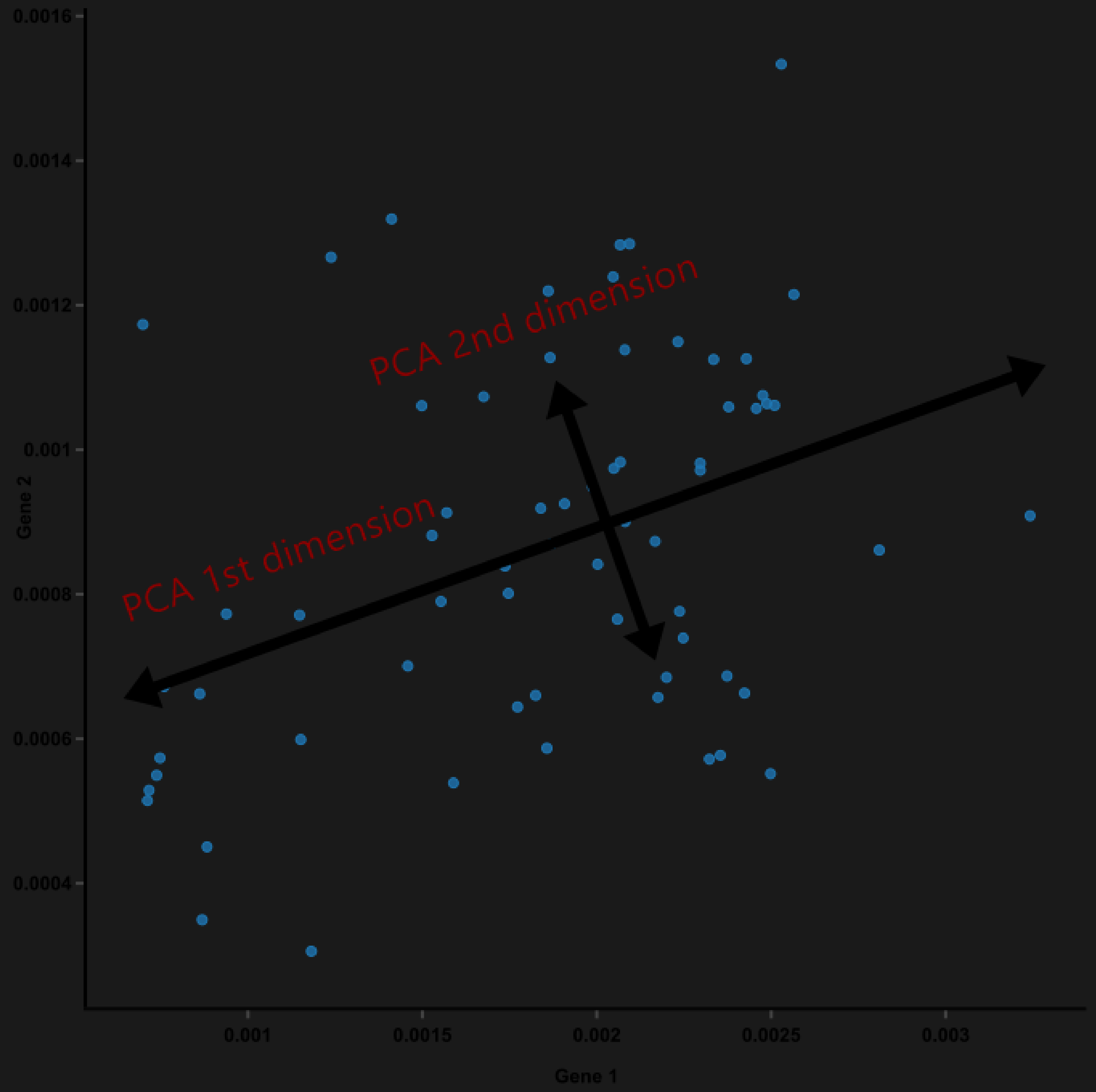
```python
from kydavra import PValueSelector
from kydavra import BregmanDivergenceSelector
from kydavra import JensenShannonSelector
```

As we can see the output of first two columns is very similar.

	PValueSelector	JensenShannonSelector	BregmanDivergenceSelector
0	age	age	workclass_0
1	workclass_2	workclass_2	workclass_1
2	workclass_3	fnlwgt	workclass_2
3	fnlwgt	education_2	workclass_3
4	education_1	education_3	education_0
5	education_2	education_4	education_1
6	education_3	education.num	education_2
7	education_4	marital.status_0	education_3
8	education.num	marital.status_2	education_4
9	marital.status_0	occupation_1	marital.status_0
10	marital.status_1	occupation_2	marital.status_1
11	marital.status_2	relationship_0	marital.status_2
12	occupation_0	relationship_2	occupation_0
13	occupation_1	race_2	occupation_1
14	occupation_2	sex	occupation_2
15	occupation_3	capital.gain	occupation_3
16	relationship_0	hours.per.week	relationship_0

Data Correlation

Principal component analysis & ICAReducer



PCA from 36 to 30 columns

ICAReducer from 36 to 36

Feature Scaling - Standartization

	0	1	2	3	4	5	6	7	8	9	...	26	27	28	29
0	0.872712	-0.019407	-0.442500	0.510810	-0.462514	0.092040	-0.032815	-0.424906	-0.547292	1.048787	...	-0.196578	-0.255178	0.267353	-0.08584
1	-1.470061	-0.019407	-0.442500	0.510810	-0.462514	-0.795088	-0.032815	-0.424906	1.827179	-0.953482	...	-0.196578	-0.255178	-1.042844	-0.08584
2	1.195853	-0.019407	-0.442500	0.510810	-0.462514	2.608657	-0.032815	-0.424906	1.827179	1.048787	...	-0.196578	-0.255178	1.577549	-0.08584
3	-0.419853	-0.019407	-0.442500	0.510810	-0.462514	-0.450815	-0.032815	2.353460	-0.547292	1.048787	...	-0.196578	-0.255178	-0.518765	-0.08584
4	-0.339067	-0.019407	-0.442500	0.510810	-0.462514	0.391721	-0.032815	-0.424906	-0.547292	-0.953482	...	-0.196578	-0.255178	-0.606112	-0.08584
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
37181	-0.500638	-0.019407	-0.442500	0.510810	-0.462514	-1.066124	-0.032815	-0.424906	-0.547292	-0.953482	...	-0.196578	-0.255178	1.053471	-0.08584
37182	0.791926	-0.019407	-0.442500	0.510810	-0.462514	-0.295012	-0.032815	-0.424906	-0.547292	-0.953482	...	-0.196578	4.373423	-0.169380	-0.08584
37183	0.388000	-0.019407	-0.442500	0.510810	-0.462514	1.079886	-0.032815	-0.424906	1.827179	1.048787	...	-0.196578	-0.255178	0.704085	-0.08584
37184	0.064859	-0.019407	-0.442500	0.510810	-0.462514	-0.750734	-0.032815	-0.424906	1.827179	1.048787	...	-0.196578	-0.255178	0.704085	-0.08584
37185	0.630356	-0.019407	2.259885	-1.957674	2.162095	-0.719079	-0.032815	-0.424906	-0.547292	1.048787	...	-0.196578	-0.255178	-0.169380	-0.08584

37186 rows × 36 columns

Standardization is less sensitive to outliers compared to normalization (Min-Max scaling).

Model Training

-SVC

-RandomForestClassifier

-MLPClassifier

-KNeighborsClassifier

Model: SVC

```
elif isinstance(model, SVC):
    clf = SVC()
    clf.fit(X_train, y_train)
    output69 = clf.predict(X_test)
    f1 = accuracy_score(y_test,output69)
    lista4.append(f1)
```

```
fun1(SVC())
```

```
Simple df
0.8911108669376991
PCA dataframe
0.891000988902318 4
Isolation Forest dataframe
0.892922513727882 9
EllipticEnvelope dataframe
0.886094493956781 9
LocalOutlierFactor dataframe
0.888142385667138 1
```

Model: RandomForestClassifier

```
elif isinstance(model, RandomForestClassifier):
    tuned_parameters = {
    'criterion': ['gini', 'entropy', 'log_loss'],
    'max_features': ['log2', 'sqrt',None],
    'class_weight' :['balanced', 'balanced_subsample']
    }
    clf = GridSearchCV(RandomForestClassifier(), tuned_parameters,scoring='recall')
    clf.fit(X_train, y_train)
    output69 = clf.predict(X_test)
    f1 = accuracy_score(y_test,output69)
    lista2.append(f1)
```

```
fun1(RandomForestClassifier())
```

```
Simple df
0.8978134270959235
PCA dataframe
0.8905614767607956
Isolation Forest dataframe
0.8942953020134228
EllipticEnvelope dataframe
0.8902453912831156
LocalOutlierFactor dataframe
0.8892032060348892
```

Model: KNeighborsClassifier

```
elif isinstance(model, KNeighborsClassifier):
    tuned_parameters = {
    'weights': ['uniform', 'distance'],
    'algorithm': ['auto', 'ball_tree', 'kd_tree', 'brute']
    }
    clf = GridSearchCV(KNeighborsClassifier(), tuned_parameters,scoring='recall')
    clf.fit(X_train, y_train)
    output69 = clf.predict(X_test)
    f1 = accuracy_score(y_test,output69)
    lista3.append(f1)
```

```
fun1(KNeighborsClassifier())
```

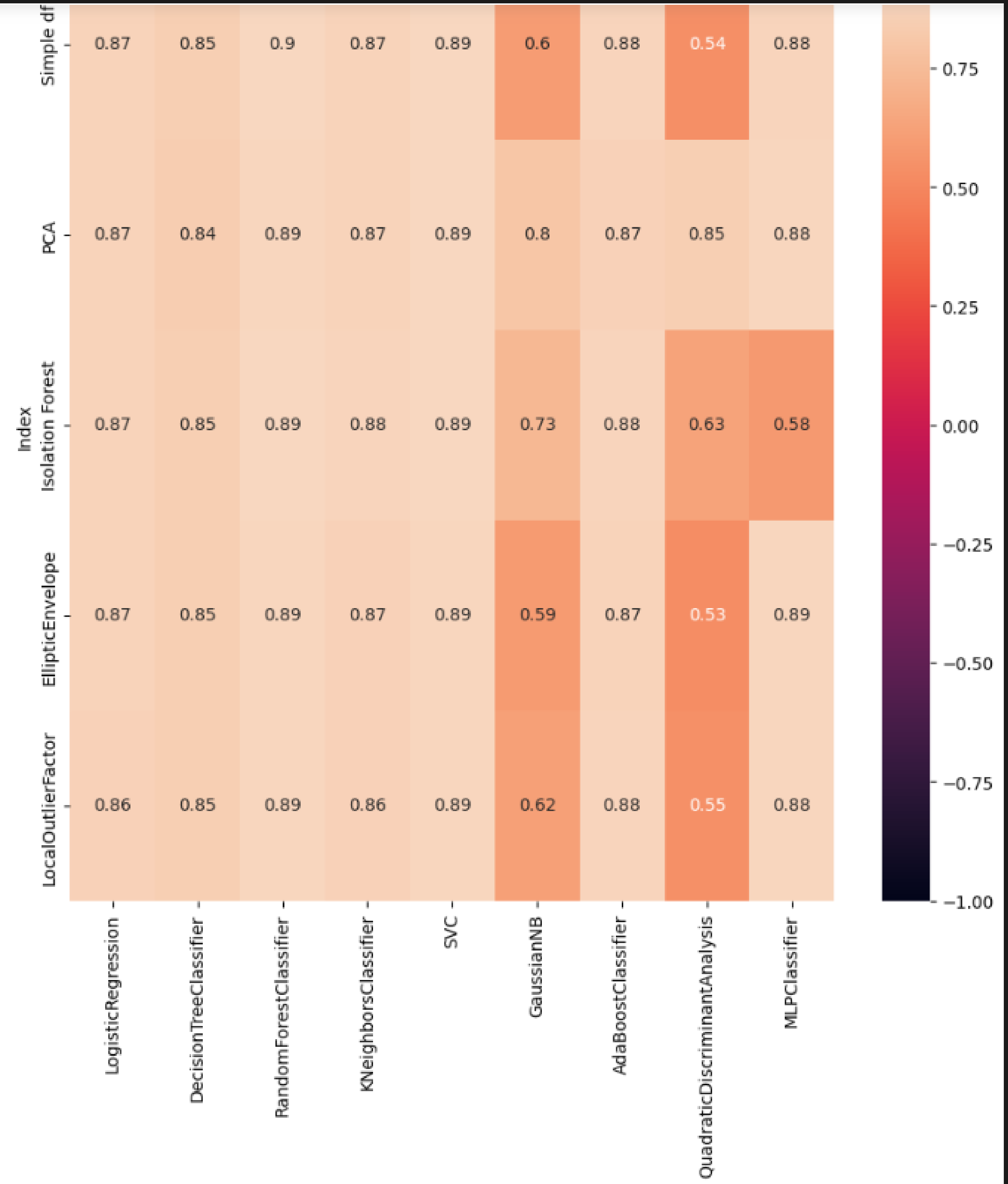
```
Simple df
0.8674870893308427
PCA dataframe
0.869245137896934
Isolation Forest dataframe
0.87614399023795
EllipticEnvelope dataframe
0.867049200341838б
LocalOutlierFactor dataframe
0.864097123998114
```

Model: MLPClassifier

```
elif isinstance(model, MLPClassifier):
    tuned_parameters = {
    'activation':['identity', 'logistic', 'tanh', 'relu'],
    'solver': ['lbfgs', 'sgd', 'adam'],
    'learning_rate':['constant', 'invscaling', 'adaptive']
    }
    clf = GridSearchCV(MLPClassifier(), tuned_parameters,scoring='recall')
    clf.fit(X_train, y_train)
    output69 = clf.predict(X_test)
    f1 = accuracy_score(y_test,output69)
    lista8.append(f1)
```

```
fun1(MLPClassifier())
```

```
Simple df
0.878694648939677
PCA dataframe
0.884518184814856
Isolation Forest dataframe
0.584655276082977
EllipticEnvelope dataframe
0.887925772188979
LocalOutlierFactor dataframe
0.884370579915134
```


Index
Simple df
0.87 0.85 0.9 0.87 0.89 0.6 0.88 0.54 0.88
PCA
0.87 0.84 0.89 0.87 0.89 0.8 0.87 0.85 0.88
Isolation Forest
0.87 0.85 0.89 0.88 0.89 0.73 0.88 0.63 0.58
EllipticEnvelope
0.87 0.85 0.89 0.87 0.89 0.59 0.87 0.53 0.89
LocalOutlierFactor
0.86 0.85 0.89 0.86 0.89 0.62 0.88 0.55 0.88
LogisticRegression
DecisionTreeClassifier
RandomForestClassifier
KNeighborsClassifier
SVC
GaussianNB
AdaBoostClassifier
QuadraticDiscriminantAnalysis
MLPClassifier
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00

Model: Hyperparameter tuning

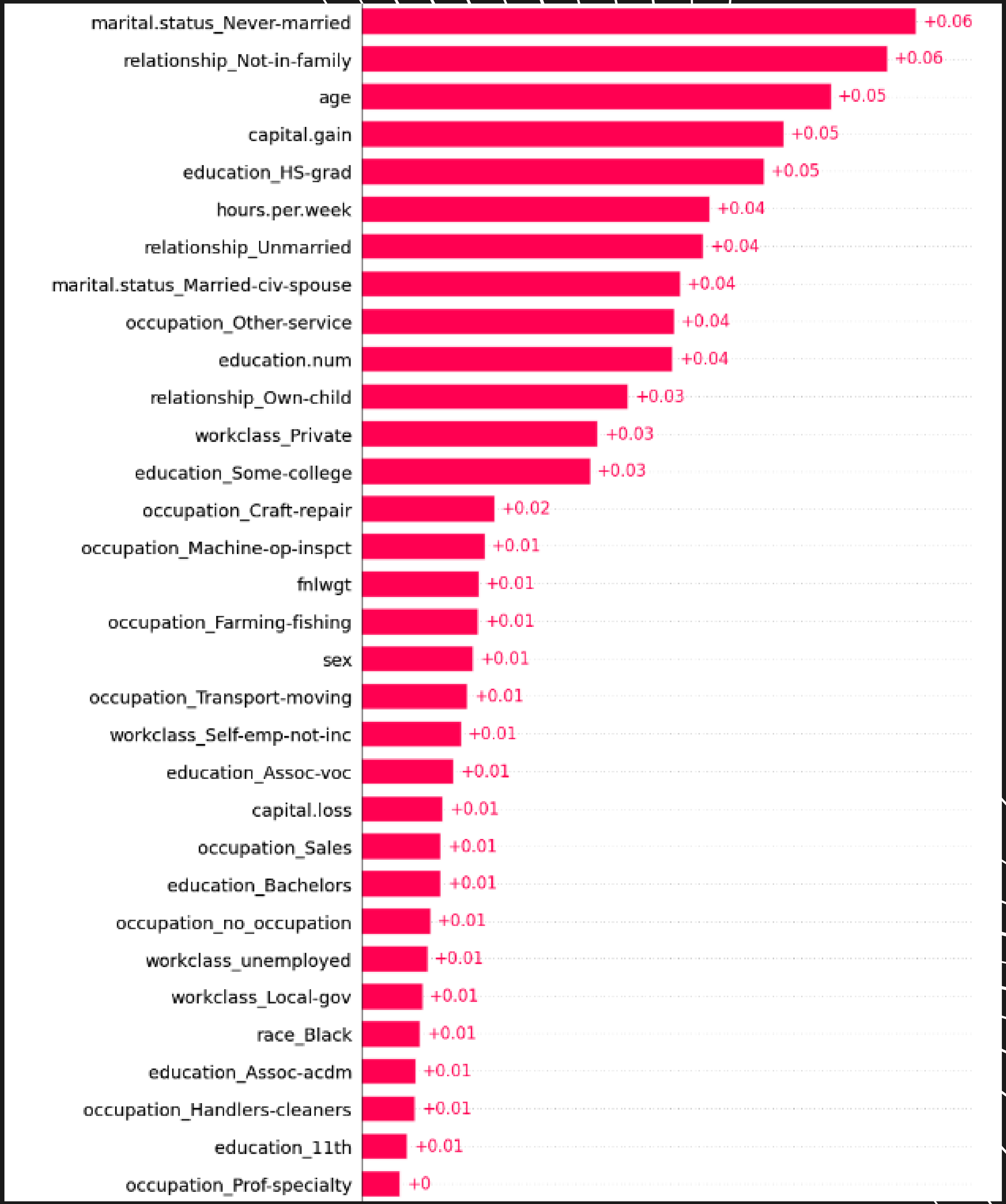
```
from sklearn.model_selection import GridSearchCV
tuned_parameters = {
        'criterion': ['gini', 'entropy', 'log_loss'],
        'max_features': ['log2', 'sqrt',None],
        'class_weight' :['balanced', 'balanced_subsample']
        }
clf = GridSearchCV(RandomForestClassifier(), tuned_parameters,scoring='recall')
clf.fit(X_train, y_train)
output69 = clf.predict(X_test)
f1 = accuracy_score(y_test,output69)
f1
```

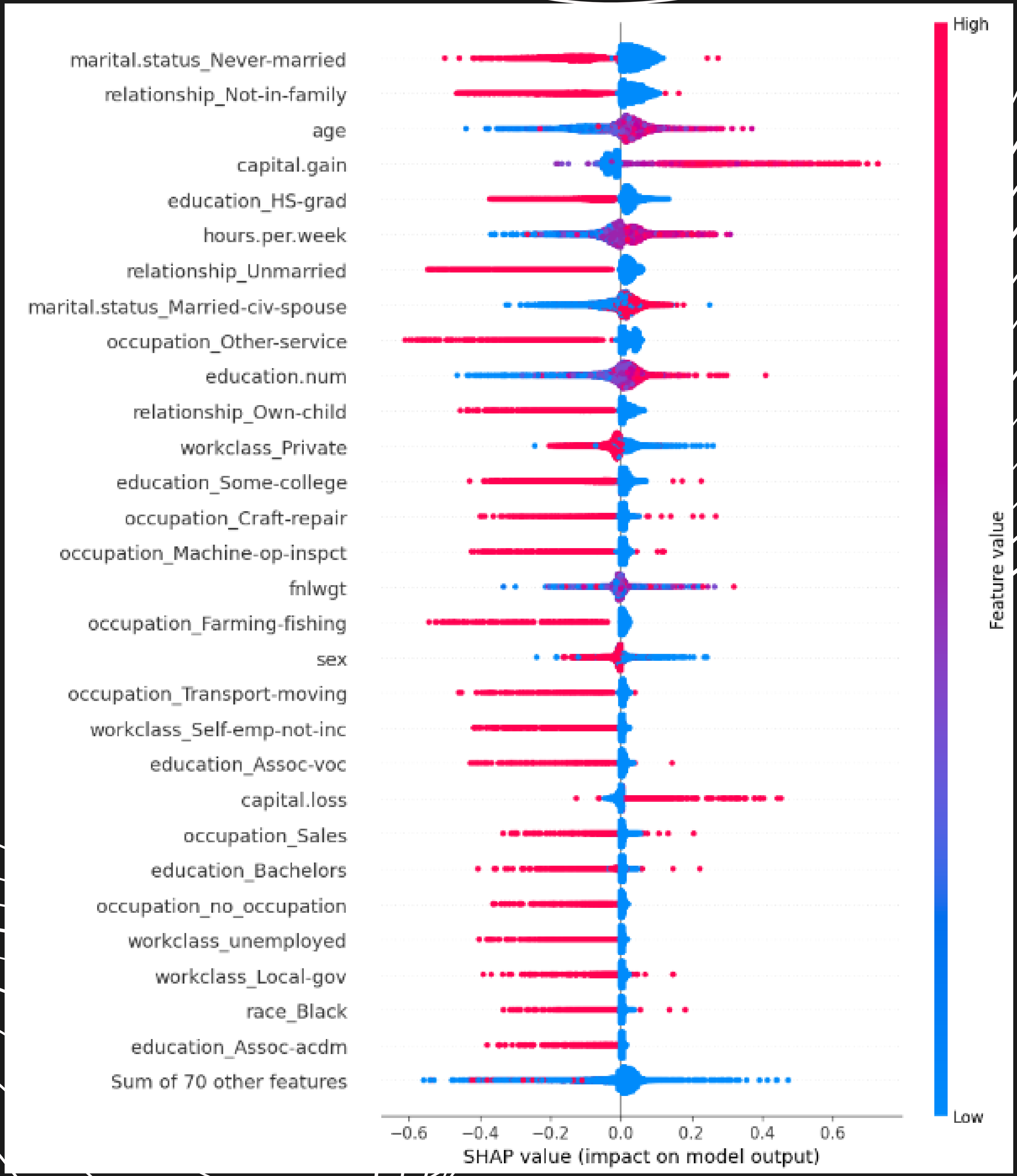
```
RandomForestClassifier

RandomForestClassifier(class_weight='balanced_subsample', criterion='entropy',
                       max_features=None)
```

Model Interpretation

RandomForestClassifier


marital.status_Never-married +0.06
relationship_Not-in-family +0.06
age +0.05
capital.gain +0.05
education_HS-grad +0.05
hours.per.week +0.04
relationship_Unmarried +0.04
marital.status_Married-civ-spouse +0.04
occupation_Other-service +0.04
education.num +0.04
relationship_Own-child +0.03
workclass_Private +0.03
education_Some-college +0.03
occupation_Craft-repair +0.02
occupation_Machine-op-inspct +0.01
fnlwgt +0.01
occupation_Farming-fishing +0.01
sex +0.01
occupation_Transport-moving +0.01
workclass_Self-emp-not-inc +0.01
education_Assoc-voc +0.01
capital.loss +0.01
occupation_Sales +0.01
education_Bachelors +0.01
occupation_no_occupation +0.01
workclass_unemployed +0.01
workclass_Local-gov +0.01
race_Black +0.01
education_Assoc-acdm +0.01
occupation_Handlers-cleaners +0.01
education_11th +0.01
occupation_Prof-specialty +0


marital.status_Never-married
relationship_Not-in-family
age
capital.gain
education_HS-grad
hours.per.week
relationship_Unmarried
marital.status_Married-civ-spouse
occupation_Other-service
education.num
relationship_Own-child
workclass_Private
education_Some-college
occupation_Craft-repair
occupation_Machine-op-inspct
fnlwgt
occupation_Farming-fishing
sex
occupation_Transport-moving
workclass_Self-emp-not-inc
education_Assoc-voc
capital.loss
occupation_Sales
education_Bachelors
occupation_no_occupation
workclass_unemployed
workclass_Local-gov
race_Black
education_Assoc-acdm
Sum of 70 other features
−0.6 −0.4 −0.2 0.0 0.2 0.4 0.6
SHAP value (impact on model output)
High
Low
Feature value

Multumesc pentru Atentie!