Annual Income Prediction on Adult Data Set

Ilayda Celenk

Content

Exploration:

- Variable Overview
- Distributions
- Missing values
- Correlation
- Scaling

Modelling:

- Logistic Regression (random state = 0)
- Logistic Regression (random state = 60)
- Random Forest (Entropy)
- Random Forest (Gini)
- KNN (k=5)
- KNN (k=10)

Evaluation :

- Compare models
- Accuracy
- P-values
- Confusion Matrices
- Independent Variables
- Model Stability
- Result

Exploration - Variable Overview

```
In [360]: dataset.info()
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 48842 entries, 0 to 48841
Data columns (total 15 columns):
age                48842 non-null int64
workclass          48842 non-null object
fnlwgt             48842 non-null int64
education          48842 non-null object
educational-num    48842 non-null int64
marital-status     48842 non-null object
occupation         48842 non-null object
relationship       48842 non-null object
race               48842 non-null object
gender             48842 non-null object
capital-gain       48842 non-null int64
capital-loss       48842 non-null int64
hours-per-week     48842 non-null int64
native-country     48842 non-null object
income             48842 non-null object
dtypes: int64(6), object(9)
memory usage: 5.6+ MB
```

Is a person’s annual income above 50K?

Exploration - Variable Overview

```
In [364]: dataset.head()
Out[364]:
   age  workclass  fnlwgt     education  educational-num      marital-status  \
0   25    Private  226802          11th                7       Never-married
1   38    Private   89814       HS-grad                9  Married-civ-spouse
2   28  Local-gov  336951    Assoc-acdm               12  Married-civ-spouse
3   44    Private  160323  Some-college               10  Married-civ-spouse
4   18          ?  103497  Some-college               10       Never-married

          occupation relationship   race  gender  capital-gain  capital-loss  \
0  Machine-op-inspct    Own-child  Black    Male             0             0
1    Farming-fishing      Husband  White    Male             0             0
2    Protective-serv      Husband  White    Male             0             0
3  Machine-op-inspct      Husband  Black    Male          7688             0
4                  ?    Own-child  White  Female             0             0

   hours-per-week native-country income
0              40  United-States  <=50K
1              50  United-States  <=50K
2              40  United-States   >50K
3              40  United-States   >50K
4              30  United-States  <=50K
```

Exploration - Variable Overview

Independent Variables

Numerical (6-many)

- Age
- Fnlwgt
- Education-num (categorical ordinal)
- Capital-gain
- Capital-loss
- Hours-per-week

Categorical (8-many)

- Workclass
- Education
- Marital-status
- Occupation
- Relationship
- Race
- Gender
- Native-country

Dependent Variable

Categorical

- Income: >50K, <=50K

```
income
<=50K    37155
>50K     11687
Name: income, dtype: int64
Number of unique values:  2
```

Exploration - Variable Overview

- dataset.describe()

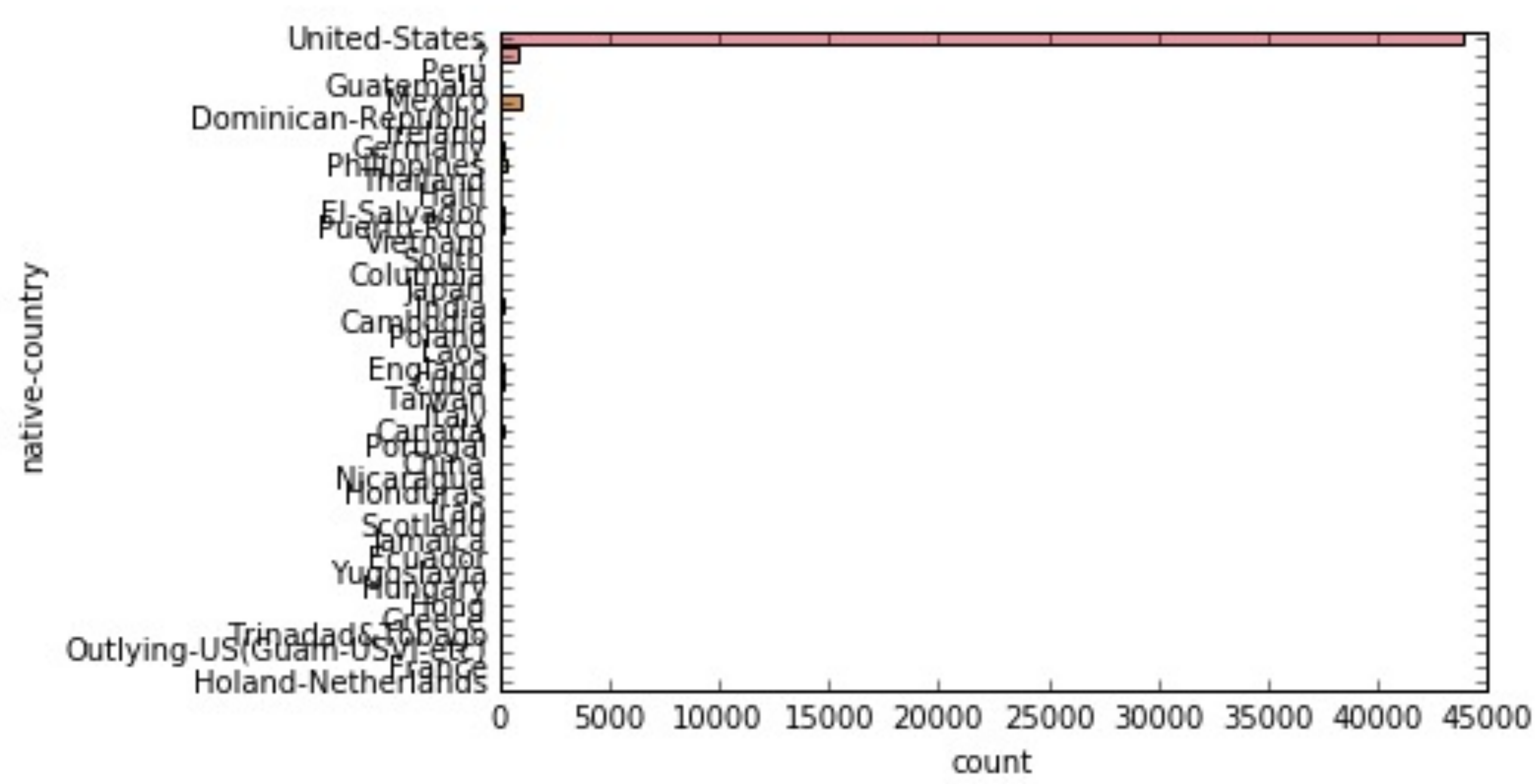
```
In [51]: dataset.describe()
Out[51]:
                age        fnlwgt  educational-num  capital-gain  \
count  48842.000000  4.884200e+04     48842.000000  48842.000000
mean      38.643585  1.896641e+05        10.078089   1079.067626
std       13.710510  1.056040e+05         2.570973   7452.019058
min       17.000000  1.228500e+04         1.000000      0.000000
25%       28.000000  1.175505e+05         9.000000      0.000000
50%       37.000000  1.781445e+05        10.000000      0.000000
75%       48.000000  2.376420e+05        12.000000      0.000000
max       90.000000  1.490400e+06        16.000000  99999.000000

       capital-loss  hours-per-week
count  48842.000000    48842.000000
mean      87.502314       40.422382
std      403.004552       12.391444
min        0.000000        1.000000
25%        0.000000       40.000000
50%        0.000000       40.000000
75%        0.000000       45.000000
max     4356.000000       99.000000
```

- age: range from 17 to 90
- fnlwgt: finalweight from 12285 to 1490400
- education-num: 1 to 16
- capital-gain: income from investment sources, apart from wages/salary, 0 to 99999
- capital-loss: losses from investment sources, apart from wages/salary, 0 to 4356
- hours-per-week: 1 to 99

Exploration - Missing Values

- Missing values in the dataset, denoted by “?”
- Approximately 3500 records with missing values:
  Workclass
  Occupation
  Native-country
- Solution:
  - Treat them as missing category(others)

Exploration - Distributions

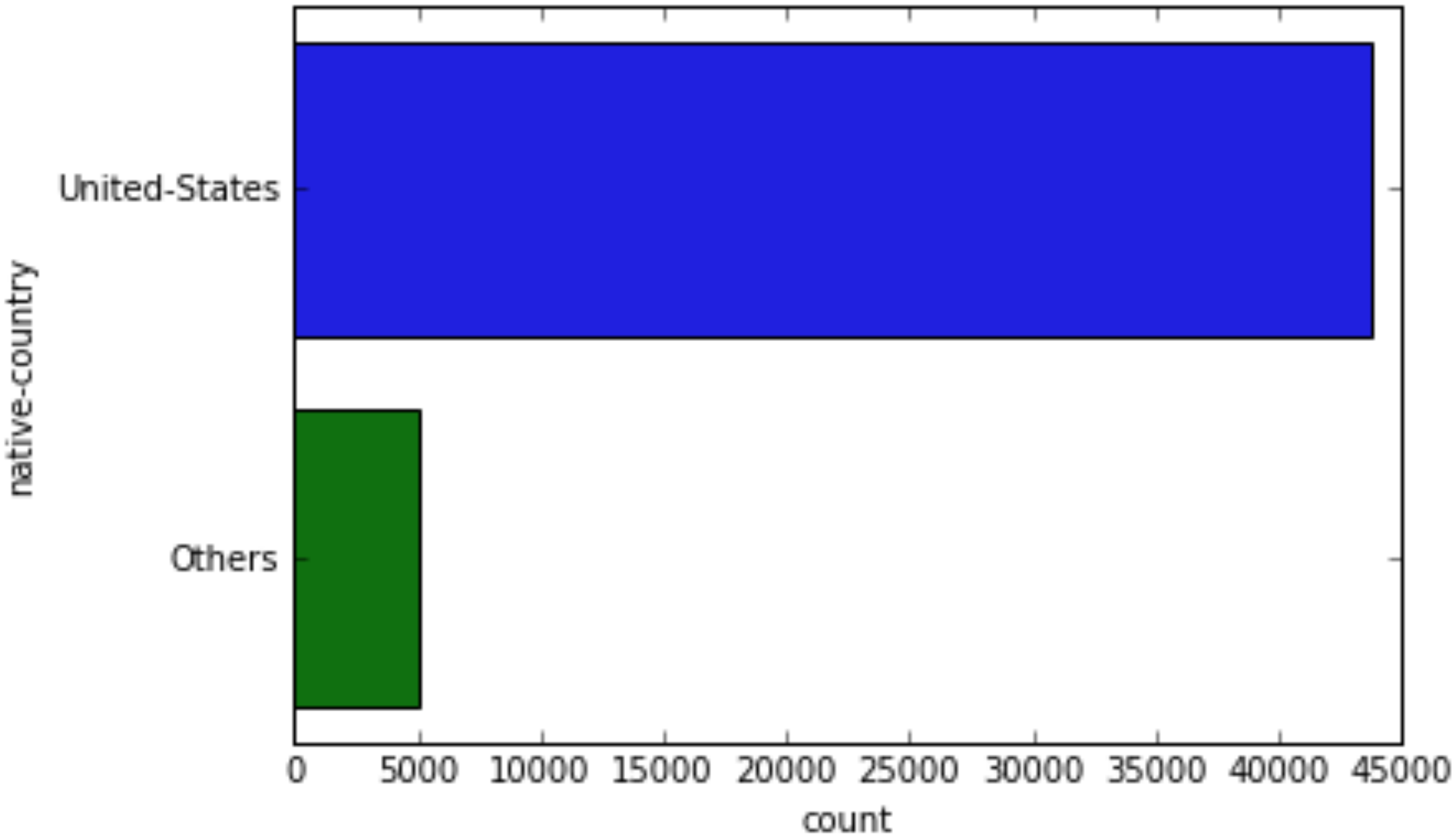


Highly skewed distribution

Solution: grouping some categories

```
native-country
United-States                 43832
Mexico                          951
?                               857
Philippines                     295
Germany                         206
Puerto-Rico                     184
Canada                          182
El-Salvador                     155
India                           151
Cuba                            138
England                         127
China                           122
South                           115
Jamaica                         106
Italy                           105
Dominican-Republic              103
Japan                            92
Guatemala                        88
Poland                           87
Vietnam                          86
Columbia                         85
Haiti                            75
Portugal                         67
Taiwan                           65
Iran                             59
Nicaragua                        49
Greece                           49
Peru                             46
Ecuador                          45
France                           38
Ireland                          37
Thailand                         30
Hong                             30
Cambodia                         28
Trinadad&Tobago                  27
Outlying-US(Guam-USVI-etc)       23
Laos                             23
Yugoslavia                       23
Scotland                         21
Honduras                         20
Hungary                          19
Holand-Netherlands                1
Name: native-country, dtype: int64
Number of unique values:  42
```

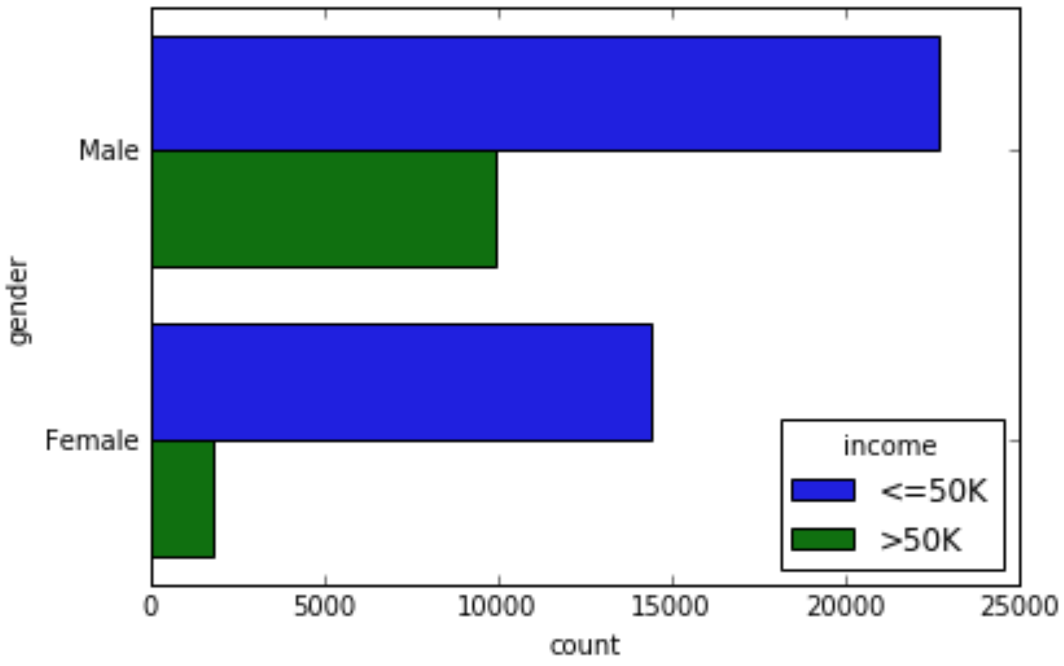
Exploration - Distributions




```
native-country
United-States    43832
Others            5010
Name: native-country, dtype: int64
Number of unique values:  2
```

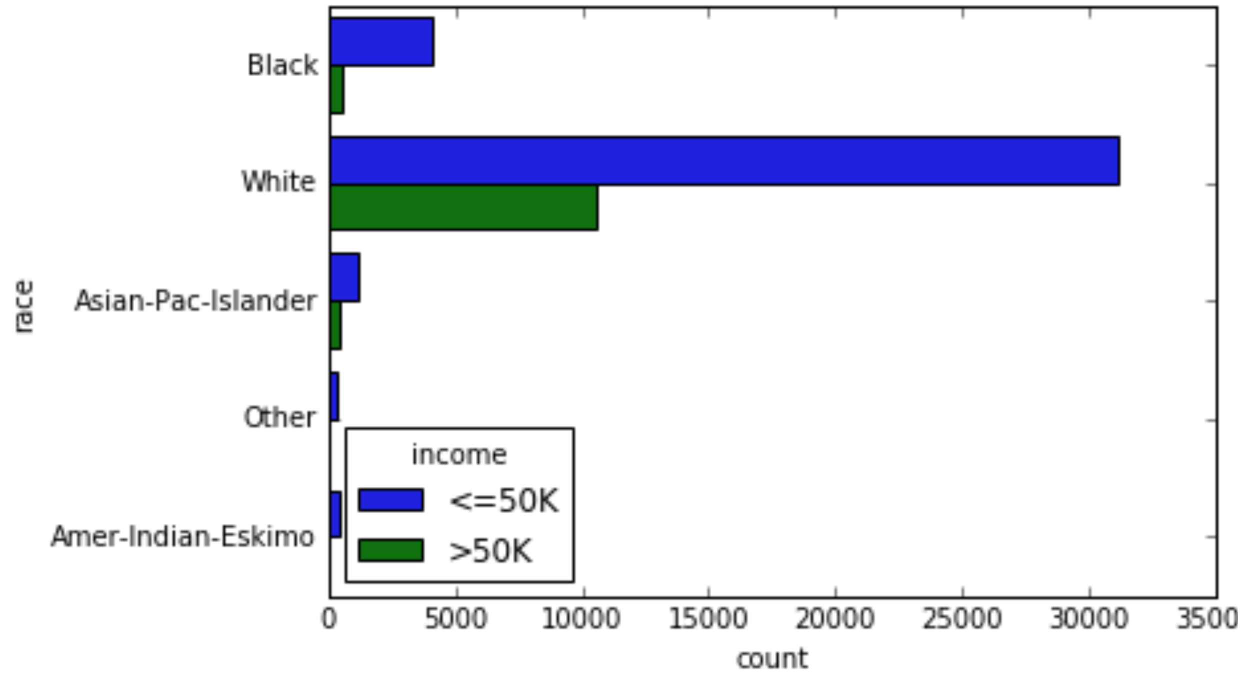
Still skewed but better

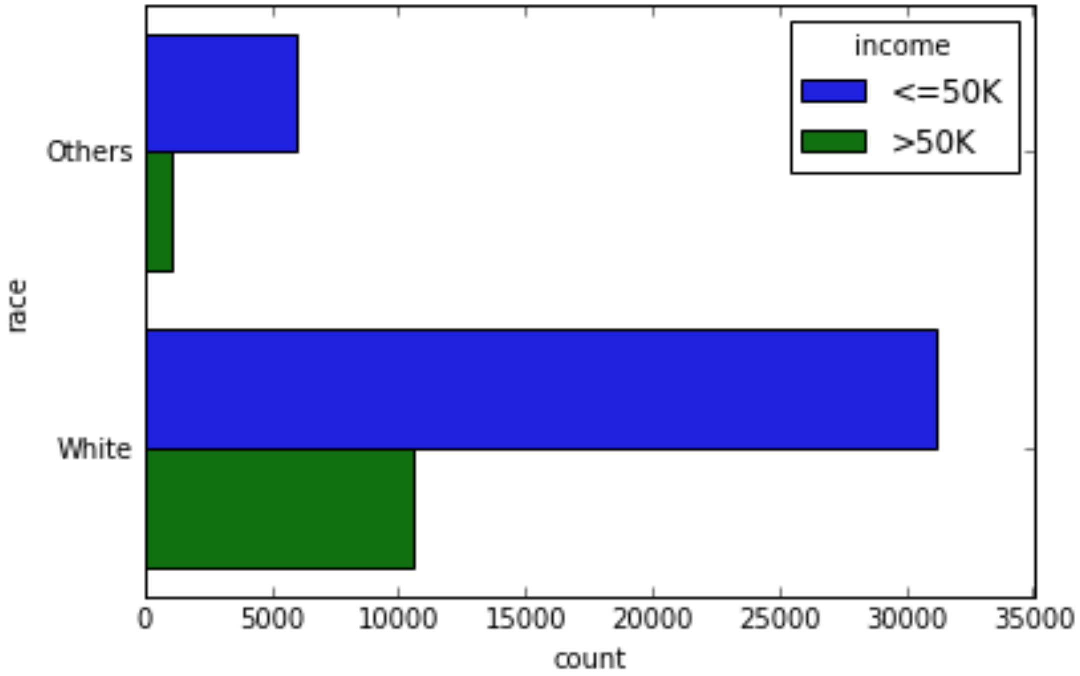
Exploration - Distributions




```
gender
Male      32650
Female    16192
Name: gender, dtype: int64
Number of unique values:  2
```

Exploration - Distributions

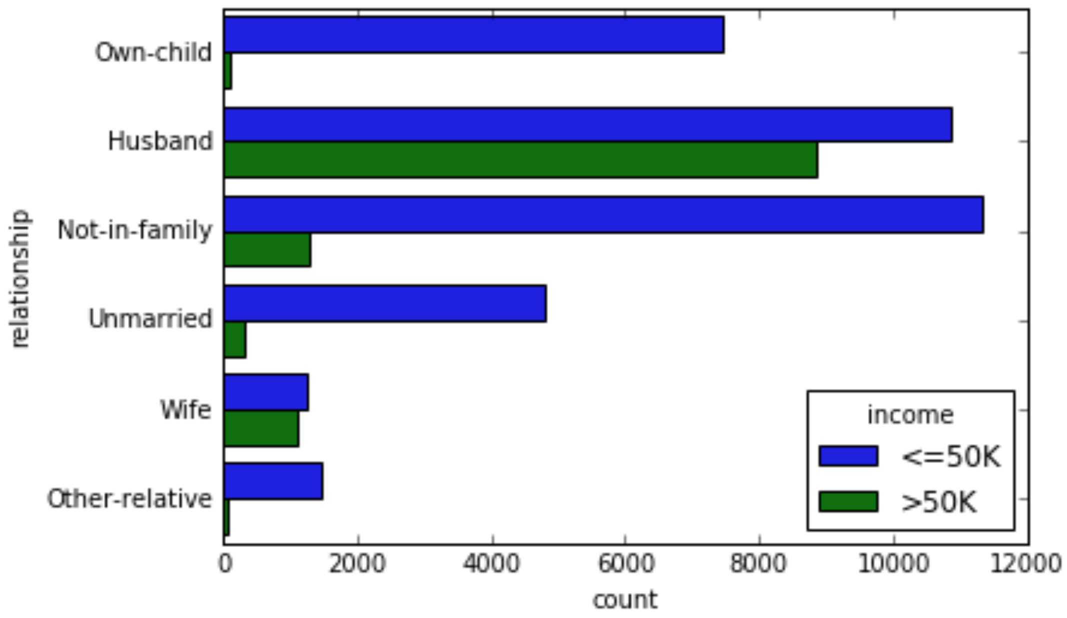





```
race
White                 41762
Black                  4685
Asian-Pac-Islander     1519
Amer-Indian-Eskimo      470
Other                   406
Name: race, dtype: int64
Number of unique values:  5
```

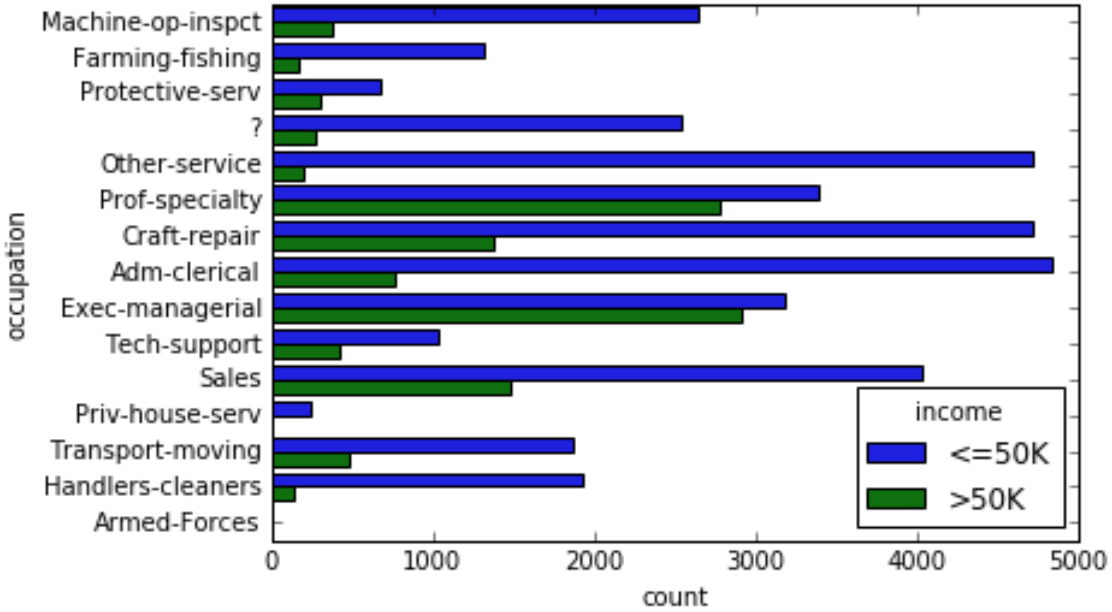
```
race
White     41762
Others     7080
Name: race, dtype: int64
Number of unique values:  2
```

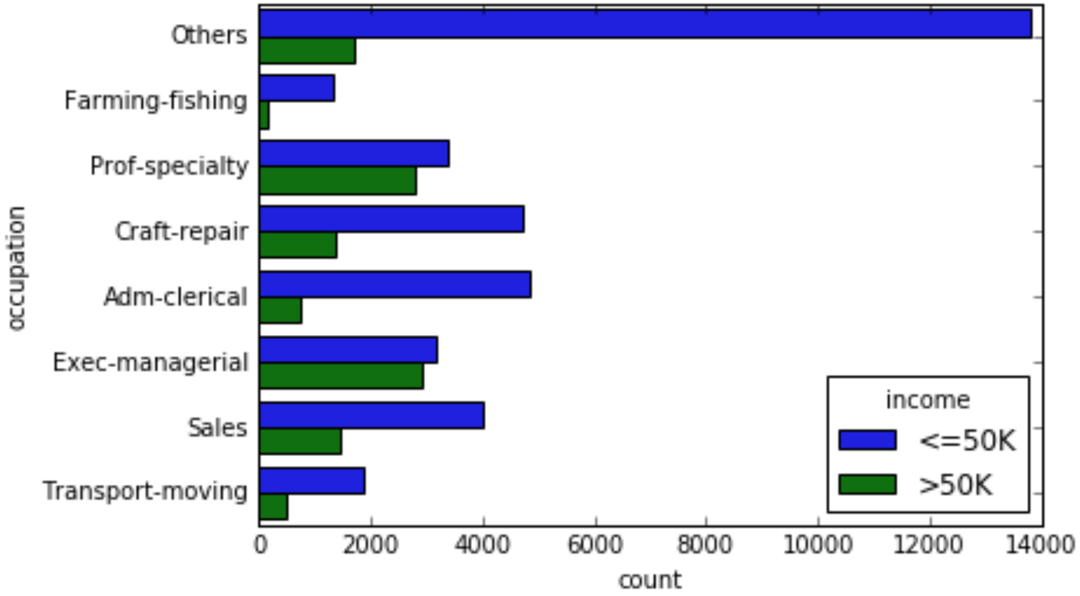
Exploration - Distributions




```
relationship
Husband           19716
Not-in-family     12583
Own-child          7581
Unmarried          5125
Wife               2331
Other-relative     1506
Name: relationship, dtype: int64
Number of unique values:  6
```

Exploration - Distributions

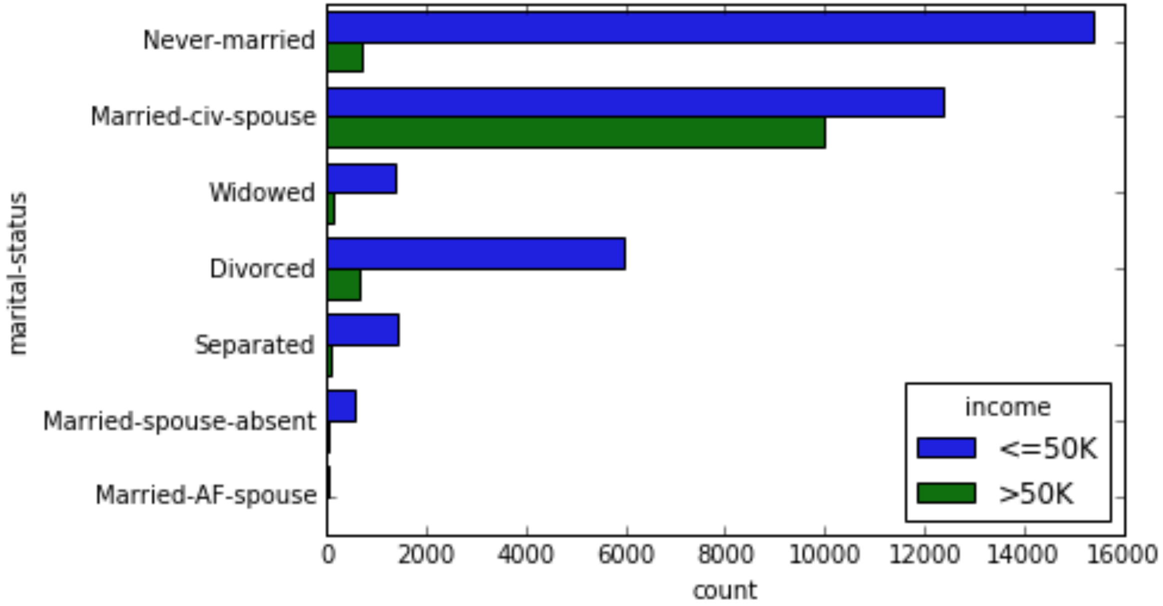




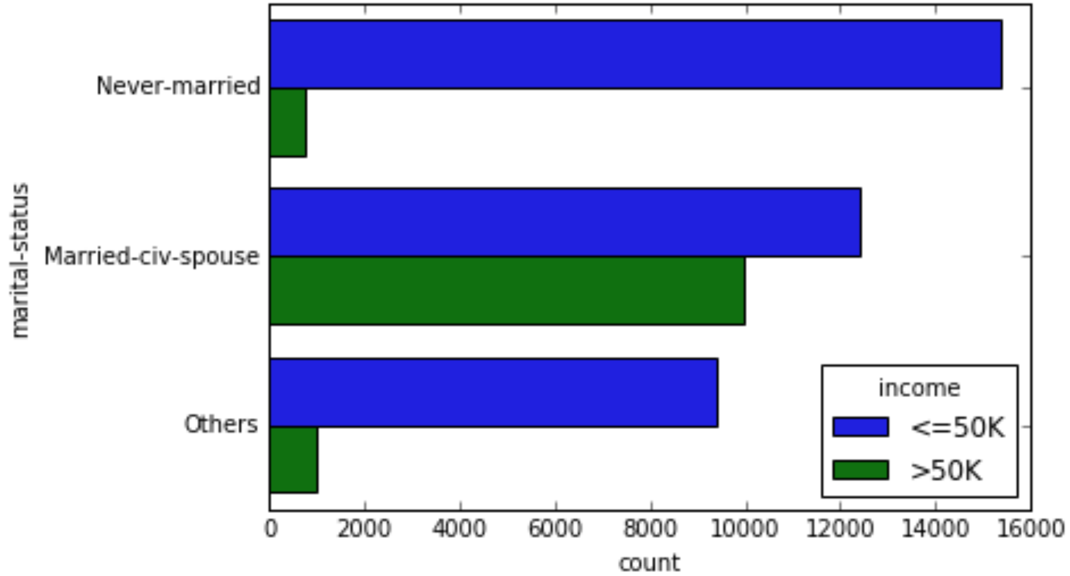

```
occupation
Prof-specialty       6172
Craft-repair         6112
Exec-managerial      6086
Adm-clerical         5611
Sales                5504
Other-service        4923
Machine-op-inspct    3022
?                    2809
Transport-moving     2355
Handlers-cleaners    2072
Farming-fishing      1490
Tech-support         1446
Protective-serv       983
Priv-house-serv       242
Armed-Forces           15
Name: occupation, dtype: int64
Number of unique values:  15
```

```
occupation
Others              15512
Prof-specialty       6172
Craft-repair         6112
Exec-managerial      6086
Adm-clerical         5611
Sales                5504
Transport-moving     2355
Farming-fishing      1490
Name: occupation, dtype: int64
Number of unique values:  8
```

Exploration - Distributions

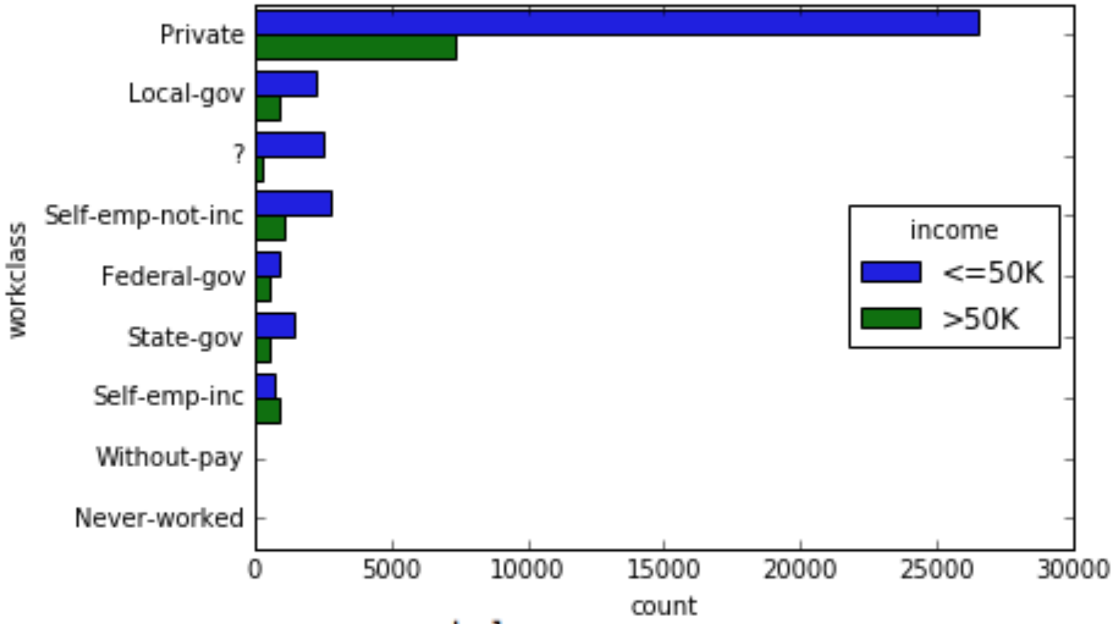



```
marital-status
Married-civ-spouse       22379
Never-married            16117
Divorced                  6633
Separated                 1530
Widowed                   1518
Married-spouse-absent      628
Married-AF-spouse           37
Name: marital-status, dtype: int64
Number of unique values:  7
```

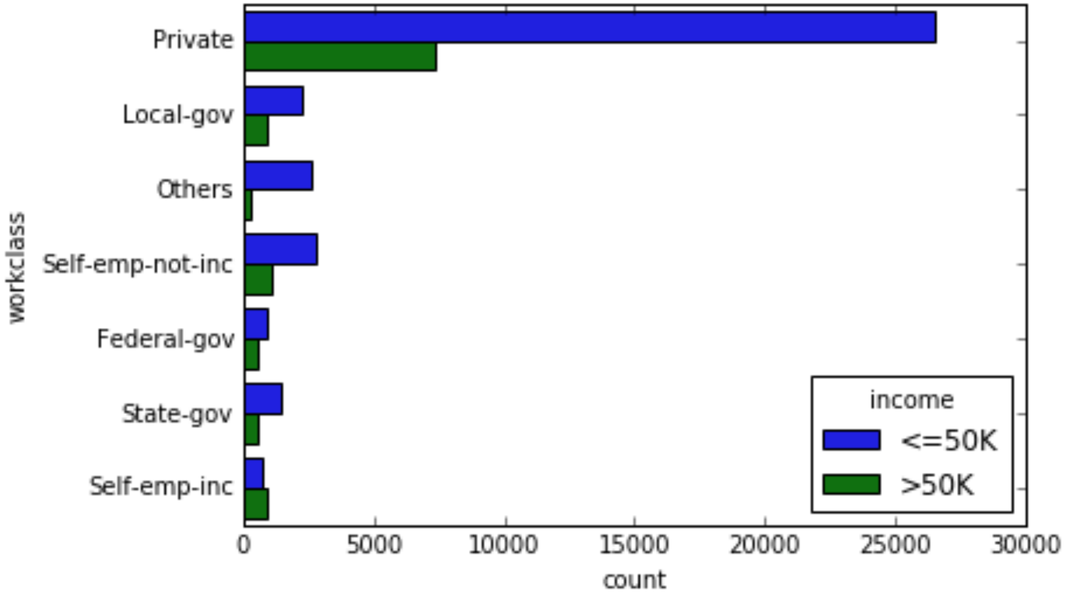



```
marital-status
Married-civ-spouse    22379
Never-married         16117
Others                10346
Name: marital-status, dtype: int64
Number of unique values:  3
```

Exploration - Distributions

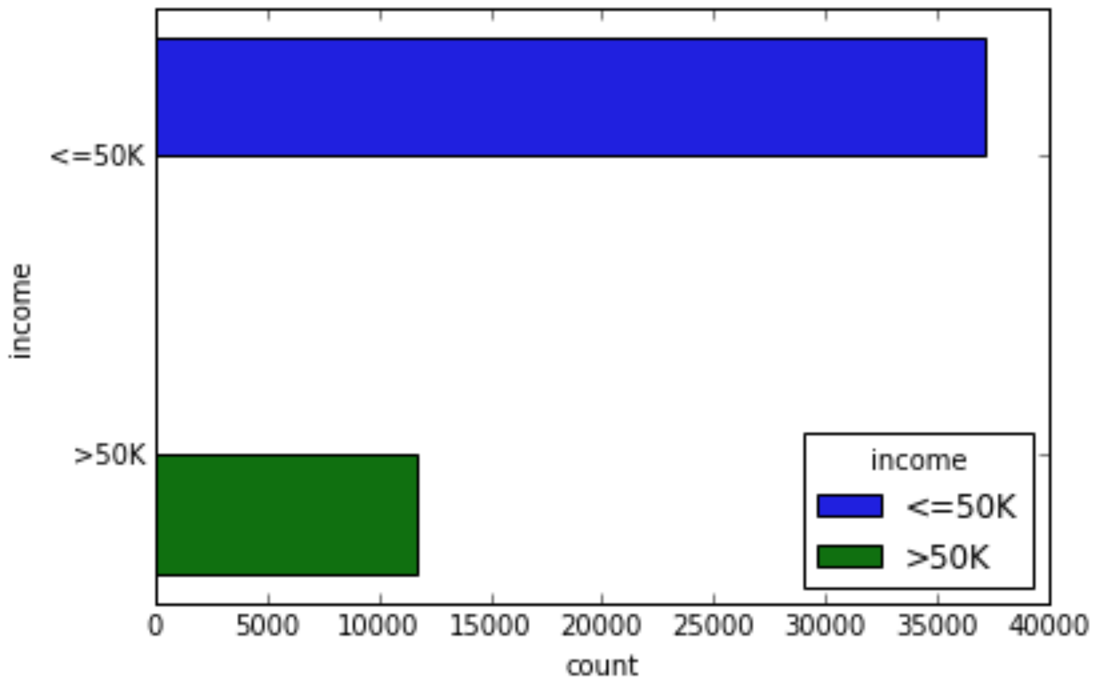



```
workclass
Private             33906
Self-emp-not-inc     3862
Local-gov            3136
?                    2799
State-gov            1981
Self-emp-inc         1695
Federal-gov          1432
Without-pay            21
Never-worked           10
Name: workclass, dtype: int64
Number of unique values:  9
```



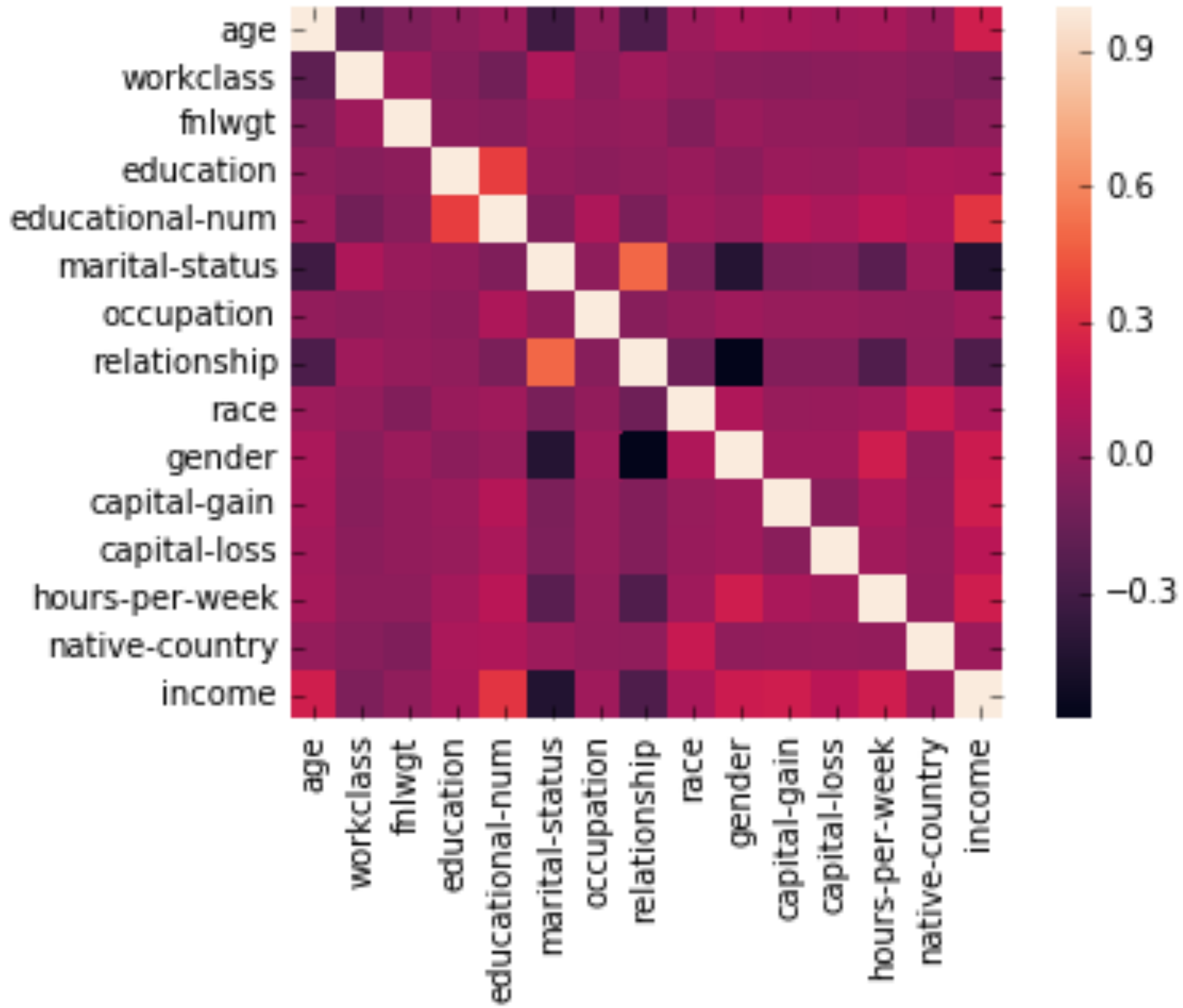

```
workclass
Private             33906
Self-emp-not-inc     3862
Local-gov            3136
Others               2830
State-gov            1981
Self-emp-inc         1695
Federal-gov          1432
Name: workclass, dtype: int64
Number of unique values:  7
```

Exploration - Distributions

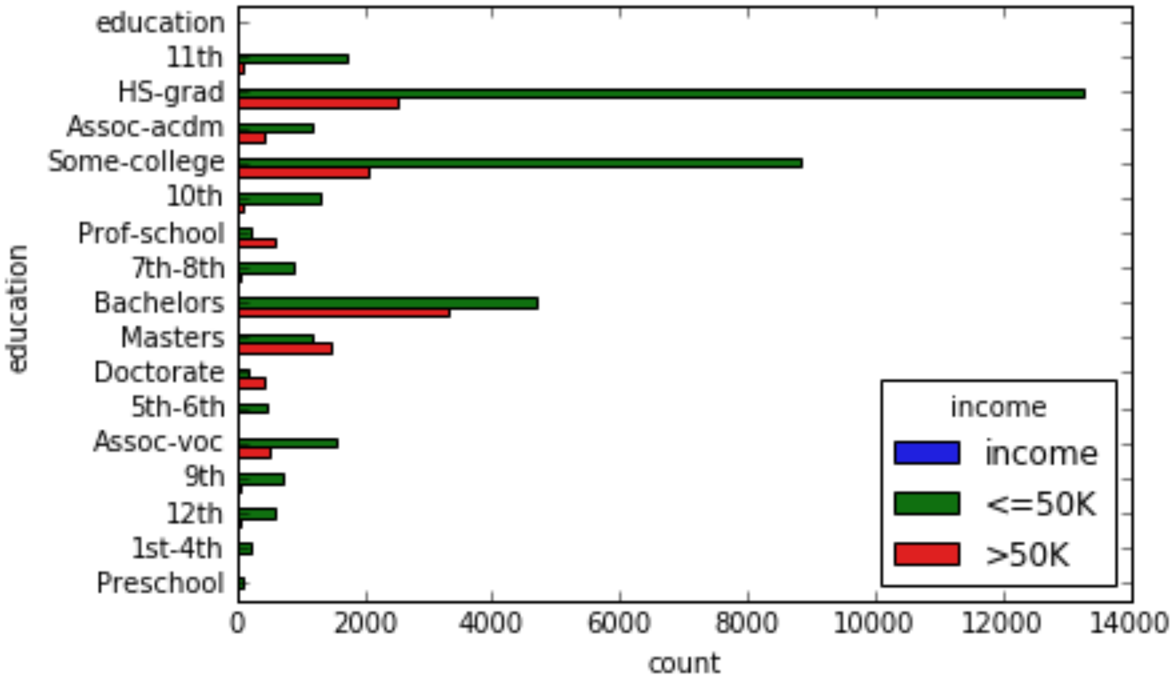



```
income
<=50K    37155
>50K     11687
Name: income, dtype: int64
Number of unique values:  2
```

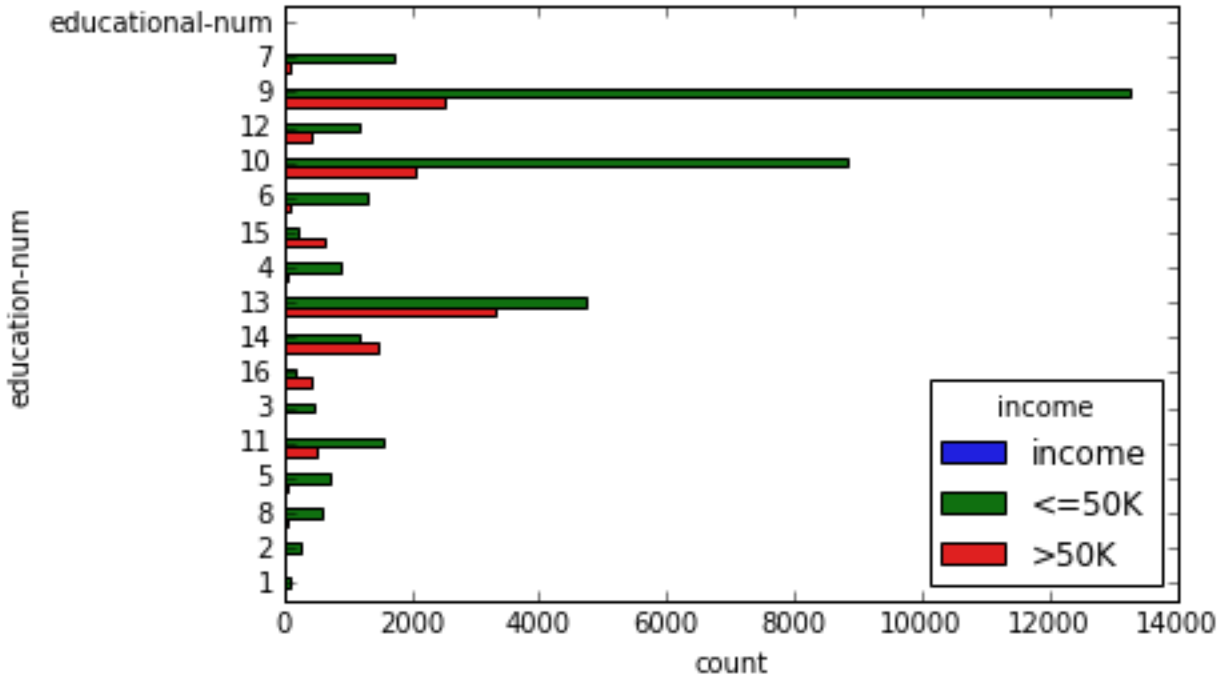
Exploration - Correlation



Exploration - Correlation



Education vs Income

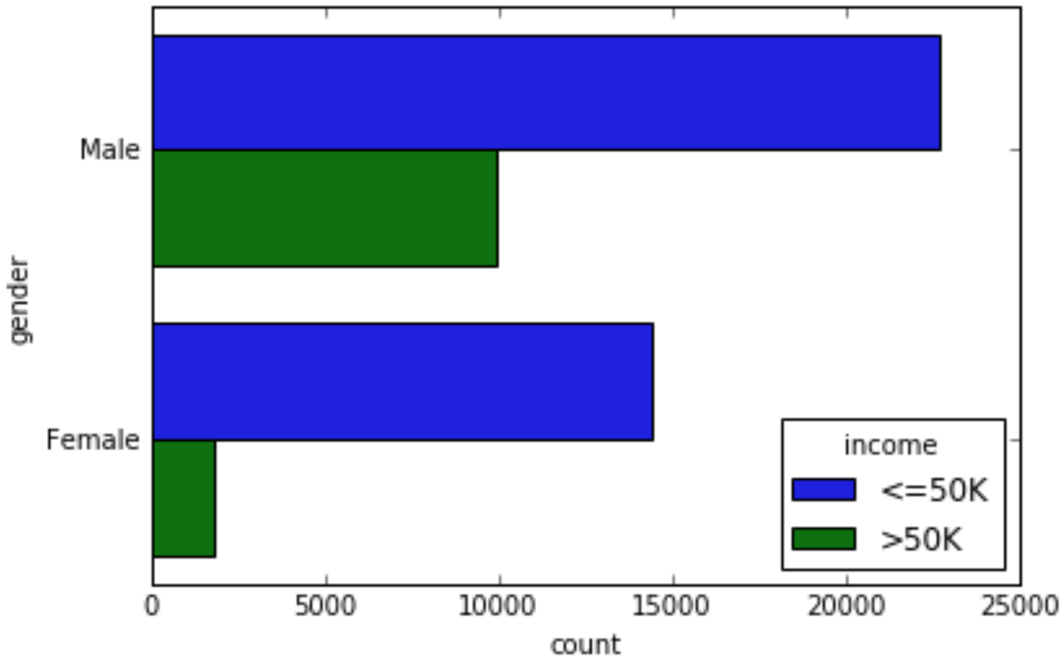


Education-num vs Income

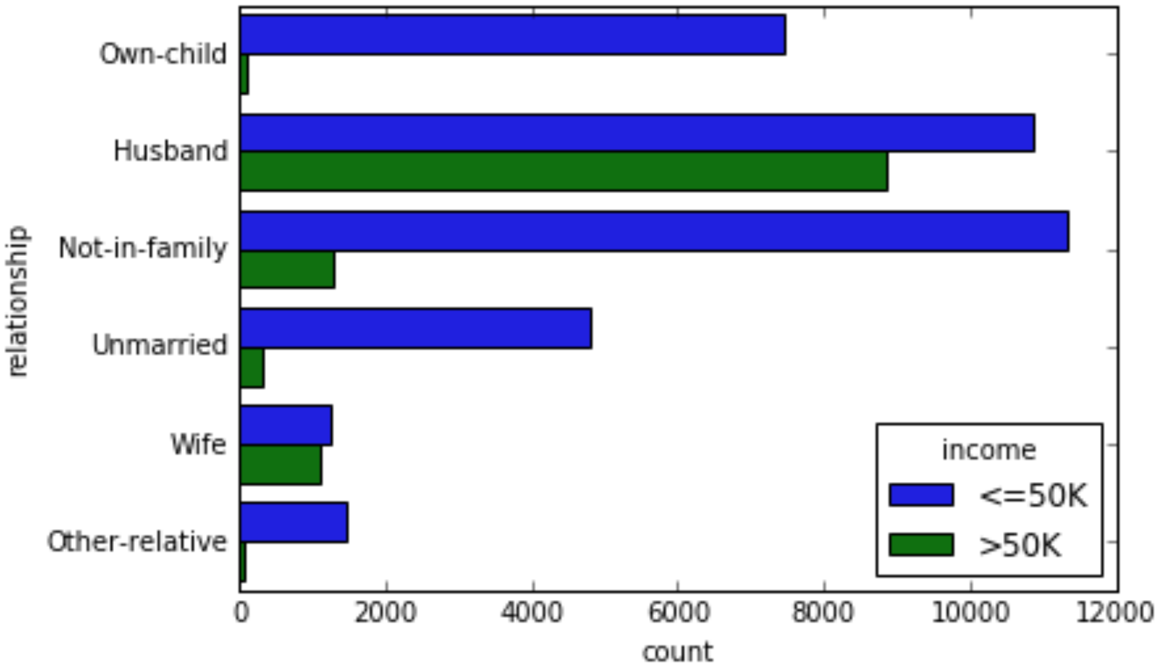
They are strictly corralated.

Solution: omit education

Exploration - Correlation



Gender vs Income



Relationship vs Income

Males are likely to be husbands, and Females are likely to be wives.

Exploration - Scaling

Ranges:

age: 17 – 90
workclass: 0 – 1
fnlwgt: 12285 – 1490400
education-num: 0 – 16
marital-status: 0 – 1
occupation: 0 – 13
relationship: 0 – 5
race: 0 – 1
gender: 0 – 1
capital-gain: 0 – 99999
capital-loss: 0 – 4356
hours-per-week: 1 – 99
native-country: 0 – 1

Scaling is definetely needed

Modelling

- Encode using get_dummies
  get_dummies: encodes the object variables and creates dummy variables with appropriate column names
  label_encoder and one_hot_encoder: we need to encode each column and create their dummy variables one variable at a time.
- Dummy trap
- Test_size = 0.2
- Scale

Modelling - Logistic Regression

TEST

	0	1
0	6903	517
1	984	1365

Index	<= 50K	> 50K
<= 50K	0.93	0.0697
> 50K	0.419	0.581

TRAIN

	0	1
0	27812	1923
1	3904	5434

Index	<= 50K	> 50K
<= 50K	0.935	0.0647
> 50K	0.418	0.582

```
In [26]: print(classification_report(y_test, y_pred_logreg1))
    ...:
             precision    recall  f1-score   support

        0.0       0.88      0.93      0.90      7420
        1.0       0.73      0.58      0.65      2349

avg / total       0.84      0.85      0.84      9769

In [27]: print(classification_report(y_train, y_train_logreg1))
    ...:
             precision    recall  f1-score   support

        0.0       0.88      0.94      0.91     29735
        1.0       0.74      0.58      0.65      9338

avg / total       0.84      0.85      0.84     39073
```

acc_test_logreg2	...	1	0.85157129695977074
acc_train_logreg2	...	1	0.84956363729429529

rec_test_logreg1	...	1	0.58109833971902936
rec_train_logreg1	...	1	0.58192332405225955

Evaluation - P-values

	coef	std err	t	P>|t|	[95.0% Conf.	Int.]
const	-0.5562	0.014	-40.236	0.000	-0.583	-0.529
x1	0.0025	0.000	18.177	0.000	0.002	0.003
x2	7.883e-08	1.48e-08	5.324	0.000	4.98e-08	1.08e-07
x3	0.0326	0.001	44.815	0.000	0.031	0.034
x4	8.12e-06	2.12e-07	38.290	0.000	7.7e-06	8.54e-06
x5	9.36e-05	3.88e-06	24.126	0.000	8.6e-05	0.000
x6	0.0026	0.000	19.395	0.000	0.002	0.003
x7	0.0987	0.012	8.493	0.000	0.076	0.122
x8	0.0071	0.009	0.756	0.449	-0.011	0.025
x9	0.0172	0.007	2.381	0.017	0.003	0.031
x10	0.0769	0.011	6.804	0.000	0.055	0.099
x11	-0.0496	0.009	-5.354	0.000	-0.068	-0.031
x12	-0.0130	0.010	-1.236	0.217	-0.034	0.008
x13	0.2912	0.005	64.058	0.000	0.282	0.300
x14	-0.0044	0.005	-0.884	0.377	-0.014	0.005
x15	0.0113	0.006	1.974	0.048	7.99e-05	0.022
x16	-0.0026	0.006	-0.478	0.633	-0.013	0.008
x17	0.1484	0.006	25.688	0.000	0.137	0.160
x18	-0.0763	0.010	-7.769	0.000	-0.095	-0.057
x19	0.1145	0.006	18.755	0.000	0.103	0.126
x20	0.0571	0.006	10.012	0.000	0.046	0.068
x21	-0.0247	0.008	-3.151	0.002	-0.040	-0.009
x22	0.0219	0.005	4.793	0.000	0.013	0.031
x23	0.0318	0.004	8.032	0.000	0.024	0.040
x24	0.0107	0.005	2.030	0.042	0.000	0.021

for x16 we have 0.633
so we'll remove 16

	coef	std err	t	P>|t|	[95.0% Conf.	Int.]
const	-0.5559	0.014	-40.268	0.000	-0.583	-0.529
x1	0.0025	0.000	18.179	0.000	0.002	0.003
x2	7.881e-08	1.48e-08	5.323	0.000	4.98e-08	1.08e-07
x3	0.0326	0.001	44.822	0.000	0.031	0.034
x4	8.121e-06	2.12e-07	38.295	0.000	7.71e-06	8.54e-06
x5	9.36e-05	3.88e-06	24.127	0.000	8.6e-05	0.000
x6	0.0026	0.000	19.397	0.000	0.002	0.003
x7	0.0981	0.012	8.498	0.000	0.075	0.121
x8	0.0064	0.009	0.695	0.487	-0.012	0.025
x9	0.0165	0.007	2.332	0.020	0.003	0.030
x10	0.0761	0.011	6.809	0.000	0.054	0.098
x11	-0.0506	0.009	-5.605	0.000	-0.068	-0.033
x12	-0.0136	0.010	-1.308	0.191	-0.034	0.007
x13	0.2911	0.005	64.072	0.000	0.282	0.300
x14	-0.0043	0.005	-0.866	0.386	-0.014	0.005
x15	0.0120	0.006	2.168	0.030	0.001	0.023
x16	0.1493	0.005	27.340	0.000	0.139	0.160
x17	-0.0752	0.010	-7.863	0.000	-0.094	-0.056
x18	0.1153	0.006	19.695	0.000	0.104	0.127
x19	0.0580	0.005	10.718	0.000	0.047	0.069
x20	-0.0237	0.008	-3.133	0.002	-0.039	-0.009
x21	0.0218	0.005	4.775	0.000	0.013	0.031
x22	0.0316	0.004	8.047	0.000	0.024	0.039
x23	0.0106	0.005	2.017	0.044	0.000	0.021

for x8 we have 0.487
so we'll remove 8

Evaluation - P-values

	coef	std err	t	P>|t|	[95.0% Conf.	Int.]
const	-0.5529	0.013	-42.151	0.000	-0.579	-0.527
x1	0.0025	0.000	18.166	0.000	0.002	0.003
x2	7.888e-08	1.48e-08	5.328	0.000	4.99e-08	1.08e-07
x3	0.0326	0.001	44.823	0.000	0.031	0.034
x4	8.118e-06	2.12e-07	38.289	0.000	7.7e-06	8.53e-06
x5	9.359e-05	3.88e-06	24.124	0.000	8.6e-05	0.000
x6	0.0026	0.000	19.534	0.000	0.002	0.003
x7	0.0944	0.010	9.195	0.000	0.074	0.115
x8	0.0130	0.005	2.608	0.009	0.003	0.023
x9	0.0725	0.010	7.351	0.000	0.053	0.092
x10	-0.0542	0.007	-7.293	0.000	-0.069	-0.040
x11	-0.0172	0.009	-1.920	0.055	-0.035	0.000
x12	0.2911	0.005	64.069	0.000	0.282	0.300
x13	-0.0045	0.005	-0.901	0.368	-0.014	0.005
x14	0.0125	0.005	2.298	0.022	0.002	0.023
x15	0.1498	0.005	27.603	0.000	0.139	0.160
x16	-0.0749	0.010	-7.838	0.000	-0.094	-0.056
x17	0.1161	0.006	20.250	0.000	0.105	0.127
x18	0.0583	0.005	10.813	0.000	0.048	0.069
x19	-0.0233	0.008	-3.088	0.002	-0.038	-0.009
x20	0.0217	0.005	4.759	0.000	0.013	0.031
x21	0.0316	0.004	8.071	0.000	0.024	0.039
x22	0.0107	0.005	2.028	0.043	0.000	0.021

for x13 we have 0.368
so we'll remove 14

	coef	std err	t	P>|t|	[95.0% Conf.	Int.]
const	-0.5578	0.012	-46.650	0.000	-0.581	-0.534
x1	0.0026	0.000	20.889	0.000	0.002	0.003
x2	7.902e-08	1.48e-08	5.338	0.000	5e-08	1.08e-07
x3	0.0326	0.001	44.821	0.000	0.031	0.034
x4	8.117e-06	2.12e-07	38.284	0.000	7.7e-06	8.53e-06
x5	9.358e-05	3.88e-06	24.122	0.000	8.6e-05	0.000
x6	0.0026	0.000	19.797	0.000	0.002	0.003
x7	0.0945	0.010	9.207	0.000	0.074	0.115
x8	0.0131	0.005	2.632	0.008	0.003	0.023
x9	0.0724	0.010	7.348	0.000	0.053	0.092
x10	-0.0541	0.007	-7.283	0.000	-0.069	-0.040
x11	-0.0171	0.009	-1.904	0.057	-0.035	0.001
x12	0.2935	0.004	79.699	0.000	0.286	0.301
x13	0.0125	0.005	2.284	0.022	0.002	0.023
x14	0.1497	0.005	27.598	0.000	0.139	0.160
x15	-0.0752	0.010	-7.875	0.000	-0.094	-0.056
x16	0.1160	0.006	20.234	0.000	0.105	0.127
x17	0.0582	0.005	10.792	0.000	0.048	0.069
x18	-0.0233	0.008	-3.086	0.002	-0.038	-0.008
x19	0.0217	0.005	4.772	0.000	0.013	0.031
x20	0.0311	0.004	8.030	0.000	0.023	0.039
x21	0.0107	0.005	2.029	0.042	0.000	0.021

for x11 we have 0.057
so we'll remove 12

Evaluation - P-values

	coef	std err	t	P>|t|	[95.0% Conf. Int.]	
const	-0.5615	0.012	-47.602	0.000	-0.585	-0.538
x1	0.0026	0.000	20.953	0.000	0.002	0.003
x2	7.938e-08	1.48e-08	5.362	0.000	5.04e-08	1.08e-07
x3	0.0326	0.001	44.788	0.000	0.031	0.034
x4	8.121e-06	2.12e-07	38.306	0.000	7.71e-06	8.54e-06
x5	9.366e-05	3.88e-06	24.143	0.000	8.61e-05	0.000
x6	0.0026	0.000	19.773	0.000	0.002	0.003
x7	0.0991	0.010	9.939	0.000	0.080	0.119
x8	0.0174	0.004	3.933	0.000	0.009	0.026
x9	0.0770	0.010	8.049	0.000	0.058	0.096
x10	-0.0497	0.007	-7.039	0.000	-0.064	-0.036
x11	0.2935	0.004	79.692	0.000	0.286	0.301
x12	0.0114	0.005	2.102	0.036	0.001	0.022
x13	0.1490	0.005	27.532	0.000	0.138	0.160
x14	-0.0757	0.010	-7.929	0.000	-0.094	-0.057
x15	0.1151	0.006	20.145	0.000	0.104	0.126
x16	0.0577	0.005	10.717	0.000	0.047	0.068
x17	-0.0237	0.008	-3.144	0.002	-0.038	-0.009
x18	0.0219	0.005	4.797	0.000	0.013	0.031
x19	0.0308	0.004	7.969	0.000	0.023	0.038
x20	0.0107	0.005	2.033	0.042	0.000	0.021

all below 0.05, stop the
elimination process

Modelling - Logistic Regression

TEST

	0	1
0	6968	510
1	940	1351

Index	<= 50K	> 50K
<= 50K	0.932	0.0682
> 50K	0.41	0.59

TRAIN

	0	1
0	27727	1950
1	3928	5468

Index	<= 50K	> 50K
<= 50K	0.934	0.0657
> 50K	0.418	0.582

```
In [82]: print(classification_report(y_test2, y_pred_logreg2))
    ...:
             precision    recall  f1-score   support

        0.0       0.88      0.93      0.91      7478
        1.0       0.73      0.59      0.65      2291

avg / total       0.84      0.85      0.85      9769
```

```
In [84]: print(classification_report(y_train2, y_train_logreg2))
    ...:
             precision    recall  f1-score   support

        0.0       0.88      0.93      0.90     29677
        1.0       0.74      0.58      0.65      9396

avg / total       0.84      0.85      0.84     39073
```

acc_test_logreg2	...	1	0.8515712969597074
acc_train_logreg2	...	1	0.8495636372942952

rec_test_logreg2	...	1	0.5896988214753825
rec_train_logreg2	...	1	0.5819497658578118

Modelling - Random Forest (entropy)

TEST

	0	1
0	7176	244
1	1202	1147

Index	<= 50K	> 50K
<= 50K	0.967	0.0329
> 50K	0.512	0.488

TRAIN

	0	1
0	28821	914
1	4724	4614

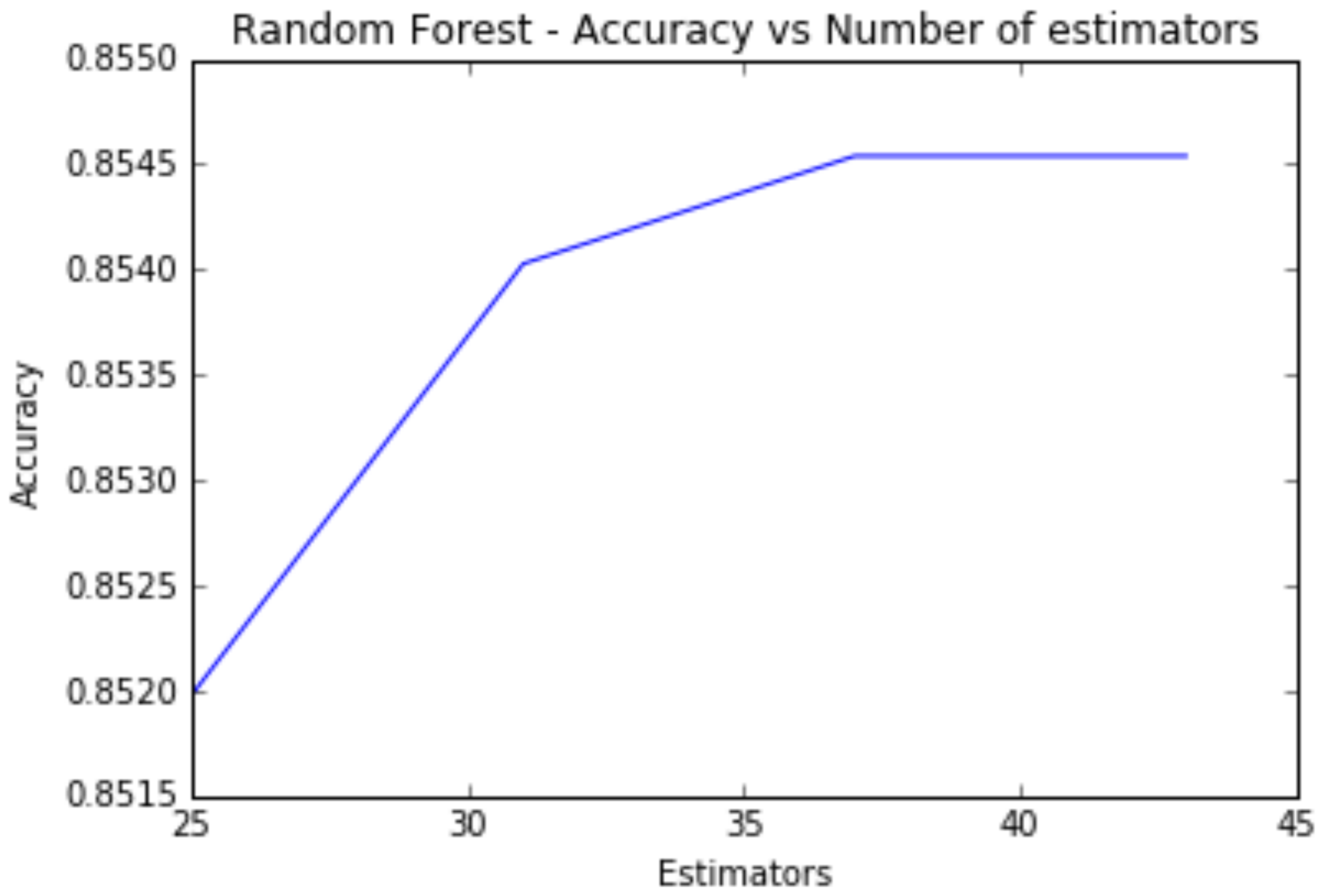
Index	<= 50K	> 50K
<= 50K	0.969	0.0307
> 50K	0.506	0.494

```
In [410]: print(classification_report(y_test, y_pred_ranfor_ent))
             precision    recall  f1-score   support

        0.0       0.86      0.97      0.91      7420
        1.0       0.82      0.49      0.61      2349

avg / total       0.85      0.85      0.84      9769

In [411]: print(classification_report(y_train, y_train_ranfor_ent))
     ...:
             precision    recall  f1-score   support

        0.0       0.86      0.97      0.91     29735
        1.0       0.83      0.49      0.62      9338

avg / total       0.85      0.86      0.84     39073
```

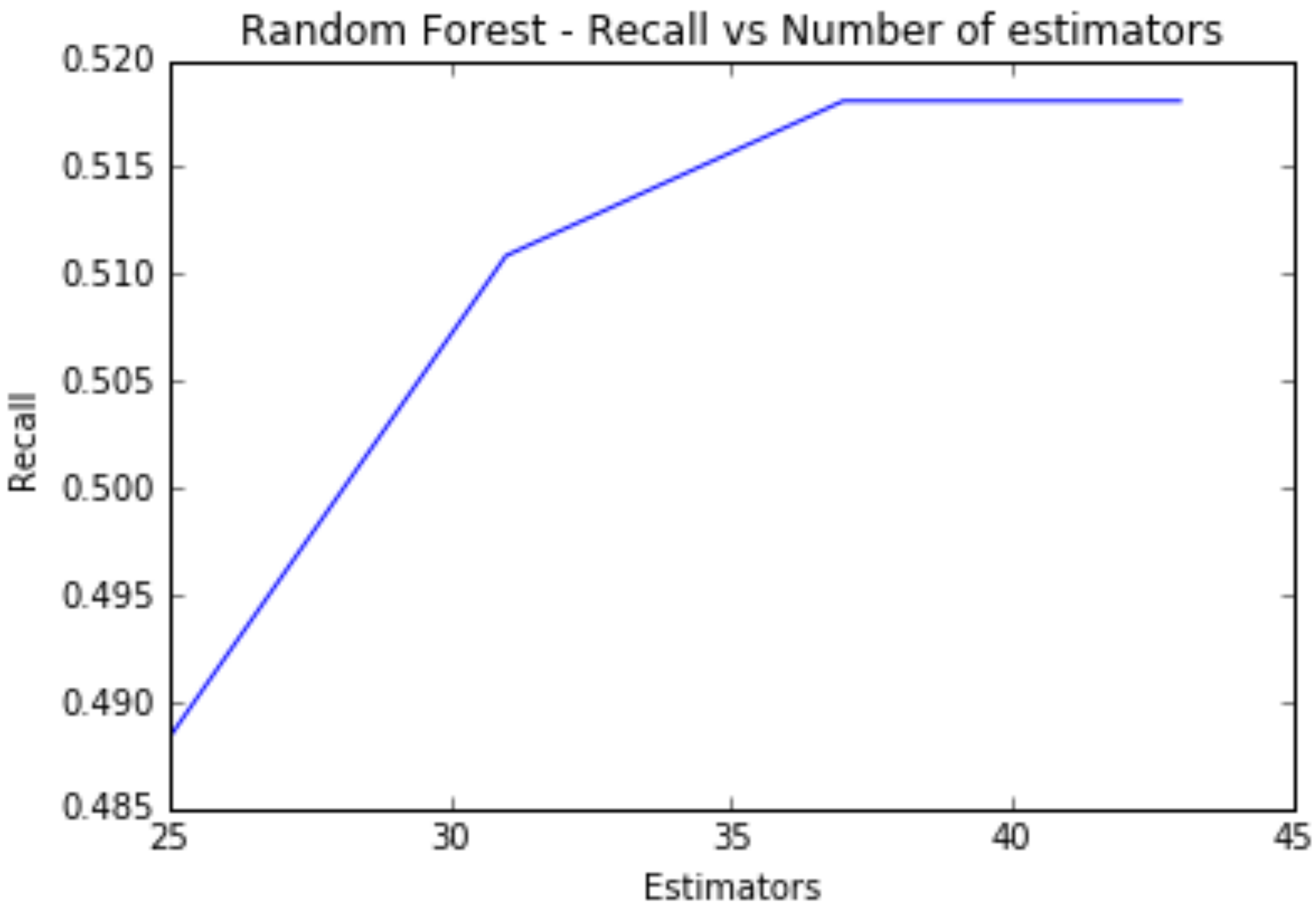
acc_test_ranfor_ent	...	1	0.8519807545091612
acc_train_ranfor_ent	...	1	0.8557059862309011

rec_test_ranfor_ent	...	1	0.4882928905917419
rec_train_ranfor_ent	...	1	0.4941100878132362

Random Forest Entropy Accuracy



Random Forest Entropy Recall



Modelling - Random Forest (gini)

TEST

	0	1
0	7190	230
1	1193	1156

Index	<= 50K	> 50K
<= 50K	0.969	0.031
> 50K	0.508	0.492

TRAIN

	0	1
0	28893	842
1	4705	4633

Index	<= 50K	> 50K
<= 50K	0.972	0.0283
> 50K	0.504	0.496

```
In [434]: print(classification_report(y_test, y_pred_ranfor_gini))
     ...:
             precision    recall  f1-score   support

        0.0       0.86      0.97      0.91      7420
        1.0       0.83      0.49      0.62      2349

avg / total       0.85      0.85      0.84      9769

In [435]: print(classification_report(y_train, y_train_ranfor_gini))
     ...:
             precision    recall  f1-score   support

        0.0       0.86      0.97      0.91     29735
        1.0       0.85      0.50      0.63      9338

avg / total       0.86      0.86      0.84     39073
```

acc_test_ranfor_gini	...	1	0.8543351417750258
acc_train_ranfor_gini	...	1	0.85803496020269754

rec_test_ranfor_gini	...	1	0.49212430821626224
rec_train_ranfor_gini	...	1	0.4961447847504819

Modelling - Knn (5nb)

TEST

	0	1
0	6719	701
1	998	1351

Index	<= 50K	> 50K
<= 50K	0.906	0.0945
> 50K	0.425	0.575

TRAIN

	0	1
0	28017	1718
1	3001	6337

Index	<= 50K	> 50K
<= 50K	0.942	0.0578
> 50K	0.321	0.679

```
In [112]: print(classification_report(y_test, y_pred_knn_5))
     ...:
             precision    recall  f1-score   support

        0.0       0.87      0.91      0.89      7420
        1.0       0.66      0.58      0.61      2349

avg / total       0.82      0.83      0.82      9769

In [113]: print(classification_report(y_train, y_train_knn_5))
             precision    recall  f1-score   support

        0.0       0.90      0.94      0.92     29735
        1.0       0.79      0.68      0.73      9338

avg / total       0.88      0.88      0.88     39073
```

acc_test_knn_5	...	1	0.8260825058859657
acc_train_knn_5	...	1	0.8792260640339876

rec_test_knn_5	...	1	0.575138356747552
rec_train_knn_5	...	1	0.678624973227672

Modelling - Knn (15nb)

TEST

	0	1
0	6821	599
1	1021	1328

Index	<= 50K	> 50K
<= 50K	0.919	0.0807
> 50K	0.435	0.565

TRAIN

	0	1
0	27755	1980
1	3671	5667

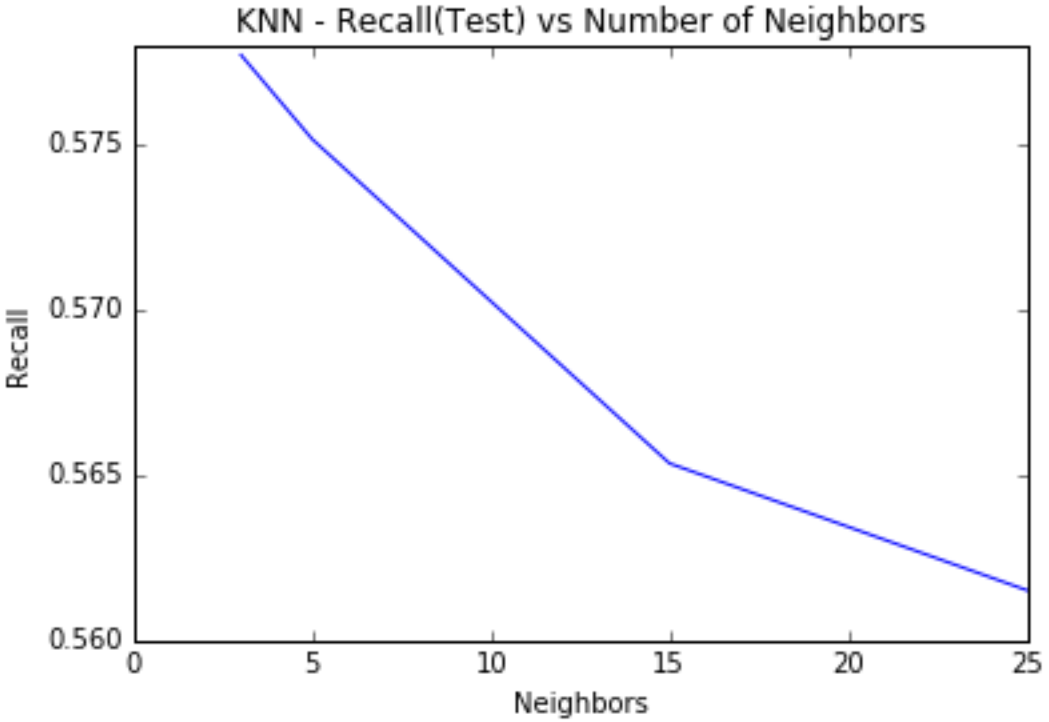
Index	<= 50K	> 50K
<= 50K	0.933	0.0666
> 50K	0.393	0.607

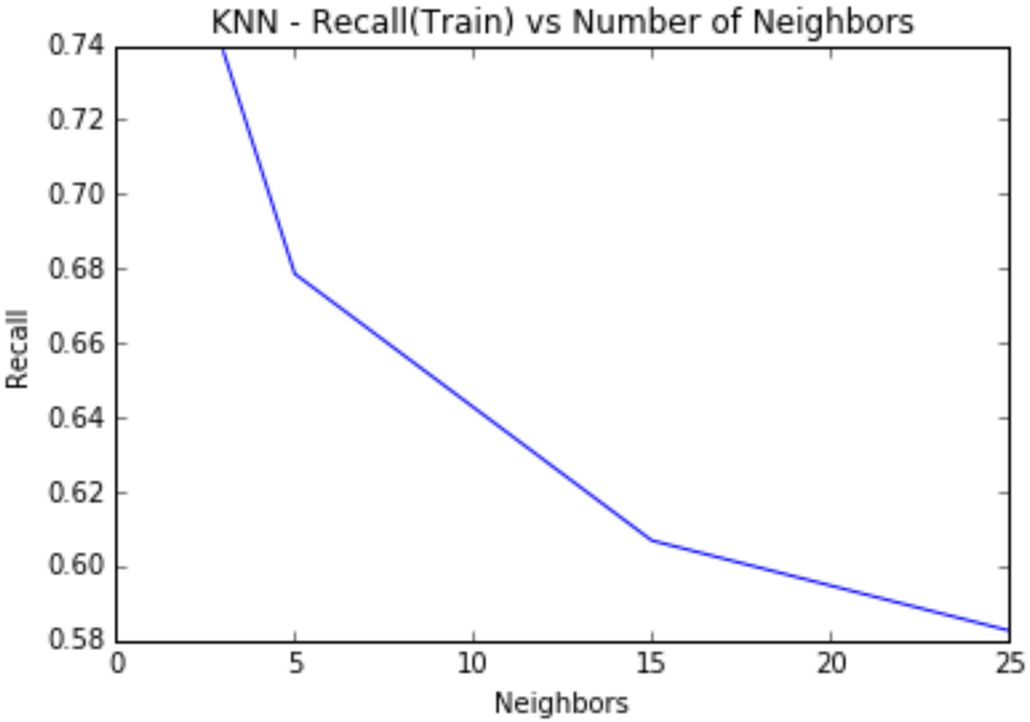
```
In [118]: print(classification_report(y_test, y_pred_knn_15))
     ...:
             precision    recall  f1-score   support

        0.0       0.87      0.92      0.89      7420
        1.0       0.69      0.57      0.62      2349

avg / total       0.83      0.83      0.83      9769

In [119]: print(classification_report(y_train, y_train_knn_15))
     ...:
             precision    recall  f1-score   support

        0.0       0.88      0.93      0.91     29735
        1.0       0.74      0.61      0.67      9338

avg / total       0.85      0.86      0.85     39073
```

acc_test_knn_15	...	1	0.83416931108608861
acc_train_knn_15	...	1	0.85537327566350163

rec_test_knn_15	...	1	0.56534695615155384
rec_train_knn_15	...	1	0.6068751338616406

Knn Recall


KNN - Recall(Test) vs Number of Neighbors
Recall
0.575
0.570
0.565
0.560
0
5
10
15
20
25
Neighbors


KNN - Recall(Train) vs Number of Neighbors
Recall
0.74
0.72
0.70
0.68
0.66
0.64
0.62
0.60
0.58
0
5
10
15
20
25
Neighbors

Evaluation - Compare Models

- For all of the classifiers, the main problem was not being able to detect positive values. We see this by looking at the recall for the test set. Mostly, the negative values are predicted as negative but the positive values are predicted as negative as well. The main reason could be the skewness of the output variable.
- For the logistic regression, we see that changing the split doesn’t affect the result, and also the precision and recall rates for test and train sets are similar as well as the confusion matrices. So, I believe that the classifier generalizes well.

Evaluation - Result

- Do you think the model generalizes well or memorizes?
  For the logistic regression models, proportions of the confusion matrices are similar, precisions for the test & train sets are almost the same; so the model generalizes well.
- Does the model produce different confusion matrices with each split?
  Yes but the proportions are similar. (Logistic Regression)