Лабораторная работа №1

Вариант 3; работу выполнил: **Санчес Орельяна Виктор Антонио**, студент из гр. **М10-414БКИ-19**

In [ ]:
```
from matplotlib import pyplot as plt
import numpy as np
import pandas as pd

dataset = pd.read_csv("datasets/adult_train.csv")
dataset.head()
```

Out[ ]:

	Age	Workclass	fnlwgt	Education	Education_Num	Martial_Status	Occupation	Relationsh
0	39	State-gov	77516	Bachelors	13	Never-married	Adm-clerical	Not-i fam
1	50	Self-emp-not-inc	83311	Bachelors	13	Married-civ-spouse	Exec-managerial	Husbar
2	38	Private	215646	HS-grad	9	Divorced	Handlers-cleaners	Not-i fam
3	53	Private	234721	11th	7	Married-civ-spouse	Handlers-cleaners	Husbar
4	28	Private	338409	Bachelors	13	Married-civ-spouse	Prof-specialty	Wi

На основании этого датасета можно поставить следующую задачу: определить, получает ли человек больше 50 тыс долларов в год или меньше. Эта задача относится к задачам классификации.

In [ ]:
```
dataset.describe()
```

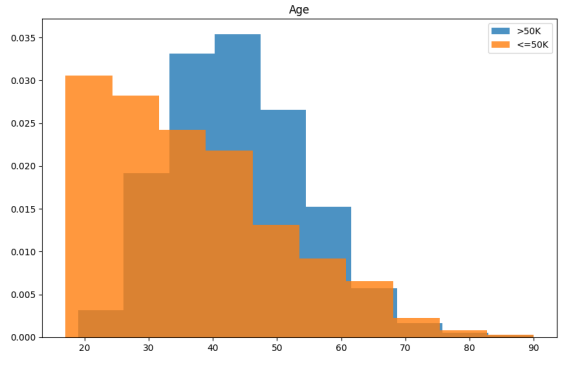
Out[ ]:

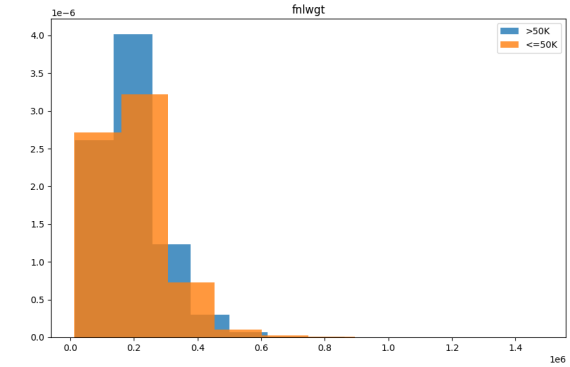
	Age	fnlwgt	Education_Num	Capital_Gain	Capital_Loss	Hours_per_we
count	32561.000000	3.256100e+04	32561.000000	32561.000000	32561.000000	32561.0000
mean	38.581647	1.897784e+05	10.080679	1077.648844	87.303830	40.4374
std	13.640433	1.055500e+05	2.572720	7385.292085	402.960219	12.3474
min	17.000000	1.228500e+04	1.000000	0.000000	0.000000	1.0000
25%	28.000000	1.178270e+05	9.000000	0.000000	0.000000	40.0000
50%	37.000000	1.783560e+05	10.000000	0.000000	0.000000	40.0000
75%	48.000000	2.370510e+05	12.000000	0.000000	0.000000	45.0000
max	90.000000	1.484705e+06	16.000000	99999.000000	4356.000000	99.0000

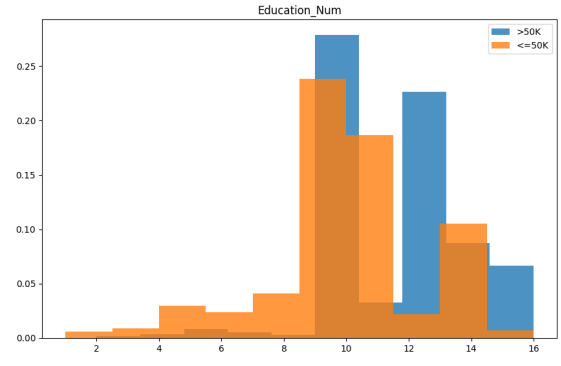
Рассмотрим распределение параметров на гистограммах.

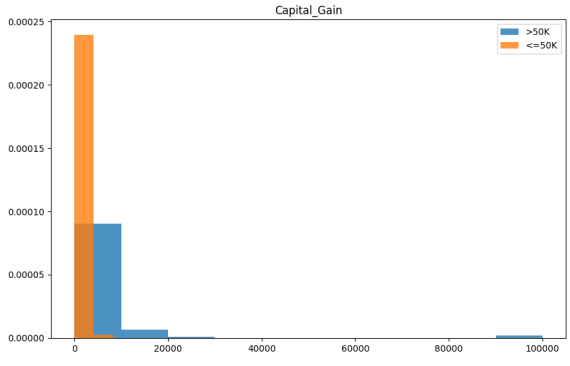
In [ ]:

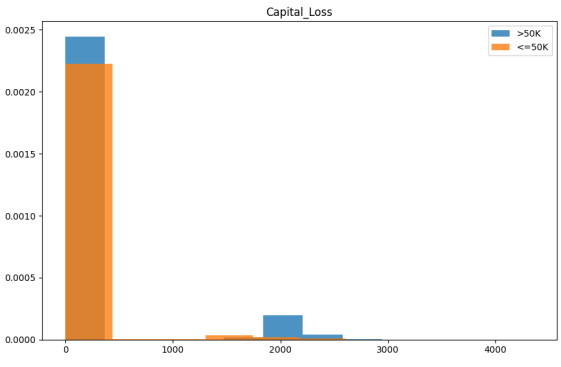
```
%matplotlib inline
fig, axs = plt.subplots(7, 2, figsize=(18,40))
num_cols = ['Age','fnlwgt','Education_Num','Capital_Gain','Capital_Loss','Hours_
str_cols = ['Workclass','Education','Martial_Status','Occupation','Relationship'

rich = dataset[dataset["Target"] == " >50K"]
poor = dataset[dataset["Target"] == " <=50K"]

for i in range(0, 6):
    x = i // 2
    y = i % 2
    axs[x, y].hist(rich[num_cols[i]], alpha = 0.8, density = True, stacked = Tru
    axs[x, y].hist(poor[num_cols[i]], alpha = 0.8, density = True, stacked = Tru
    axs[x, y].set_title(num_cols[i])
    axs[x, y].legend()
for i in range(0, 8):
    x = (i+6) // 2
    y = (i+6) % 2
    labels = dataset[str_cols[i]].unique()
    rich_count = rich[str_cols[i]].value_counts(sort=False).reindex(labels, fill
    poor_count = poor[str_cols[i]].value_counts(sort=False).reindex(labels, fill
    xr = np.arange(len(labels))
    width = 0.35
    rects1 = axs[x, y].bar(xr - width/2, rich_count, width, label='>50K')
    rects2 = axs[x, y].bar(xr + width/2, poor_count, width, label='<=50K')
    axs[x, y].set_title(str_cols[i])
    axs[x, y].set_xticks(xr, labels)
    axs[x, y].legend()

    axs[x, y].bar_label(rects1, padding=3)
    axs[x, y].bar_label(rects2, padding=3)
fig.tight_layout()
plt.show()
```

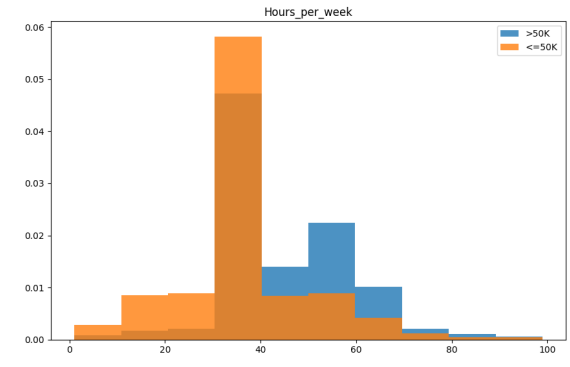

Age
>50K
<=50K

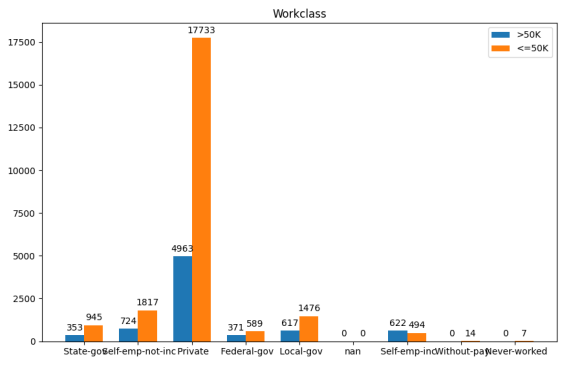

fnlwgt
1e−6
1e6
>50K
<=50K

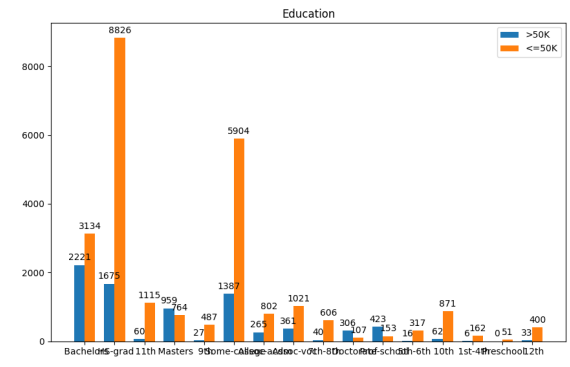

Education_Num
>50K
<=50K

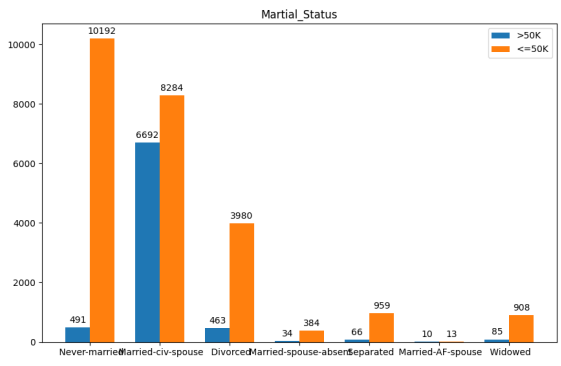

Capital_Gain
>50K
<=50K

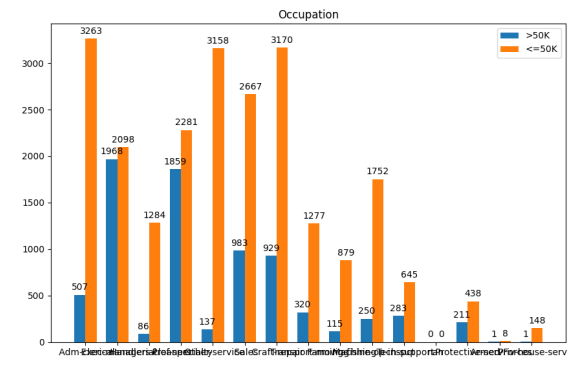

Capital_Loss
>50K
<=50K

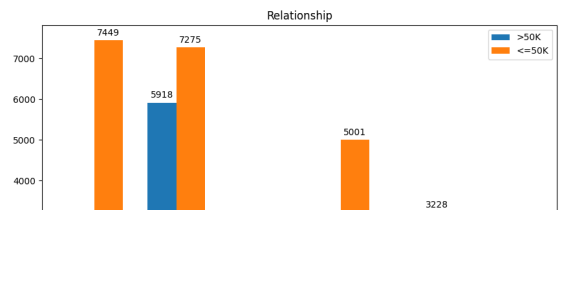

Hours_per_week
>50K
<=50K

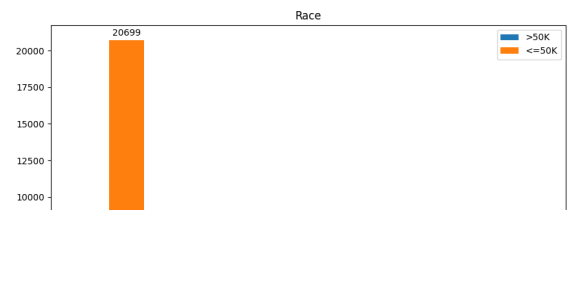

Workclass
>50K
<=50K
State-gov
Self-emp-not-inc
Private
Federal-gov
Local-gov
nan
Self-emp-inc
Without-pay
Never-worked

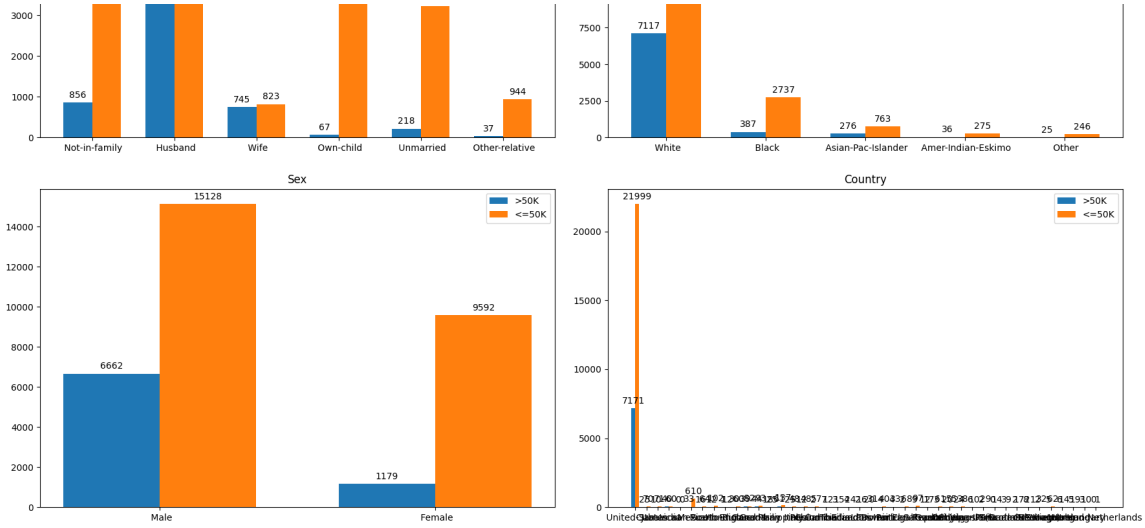

Education
>50K
<=50K
Bachelors
HS-grad
11th
Masters
9th
Some-college
Assoc-acdm
Assoc-voc
7th-8th
Doctorate
Prof-school
5th-6th
10th
1st-4th
Preschool
12th


Martial_Status
>50K
<=50K
Never-married
Married-civ-spouse
Divorced
Married-spouse-absent
Separated
Married-AF-spouse
Widowed


Occupation
>50K
<=50K


Relationship
>50K
<=50K


Race
>50K
<=50K



Из этих графиков можно выделить следующие зависимости:

- пиковый заработок приходится на средний возраст (35-50 лет).
- в среднем образование положительно влияет на размер заработной платы
- большинство как получающих больше 50K, так и меньше, работают 40 часов в неделю

In [ ]:
```
from sklearn.compose import ColumnTransformer
from sklearn.preprocessing import OneHotEncoder, MinMaxScaler, OrdinalEncoder
from sklearn.impute import SimpleImputer
from sklearn.pipeline import Pipeline

col_transform = ColumnTransformer([
    ("encode", OneHotEncoder(sparse_output=False), str_cols),
    ("scale", MinMaxScaler(), num_cols)],
    remainder=OrdinalEncoder()
)
col_transform.set_output(transform="pandas")

pipeline = Pipeline([
    ('preproc', col_transform),
    ("impute", SimpleImputer())
])

pipeline.set_output(transform="pandas")

proc_data = pipeline.fit_transform(dataset)
proc_data
```

Out[ ]:

	encode__Workclass_ Federal-gov	encode__Workclass_ Local-gov	encode__Workclass_ Never-worked	encode__Workclass_ Private	enco
0	0.0	0.0	0.0	0.0	
1	0.0	0.0	0.0	0.0	
2	0.0	0.0	0.0	1.0	
3	0.0	0.0	0.0	1.0	
4	0.0	0.0	0.0	1.0	
...	...	...	...	...	
32556	0.0	0.0	0.0	1.0	
32557	0.0	0.0	0.0	1.0	
32558	0.0	0.0	0.0	1.0	
32559	0.0	0.0	0.0	1.0	
32560	0.0	0.0	0.0	0.0	

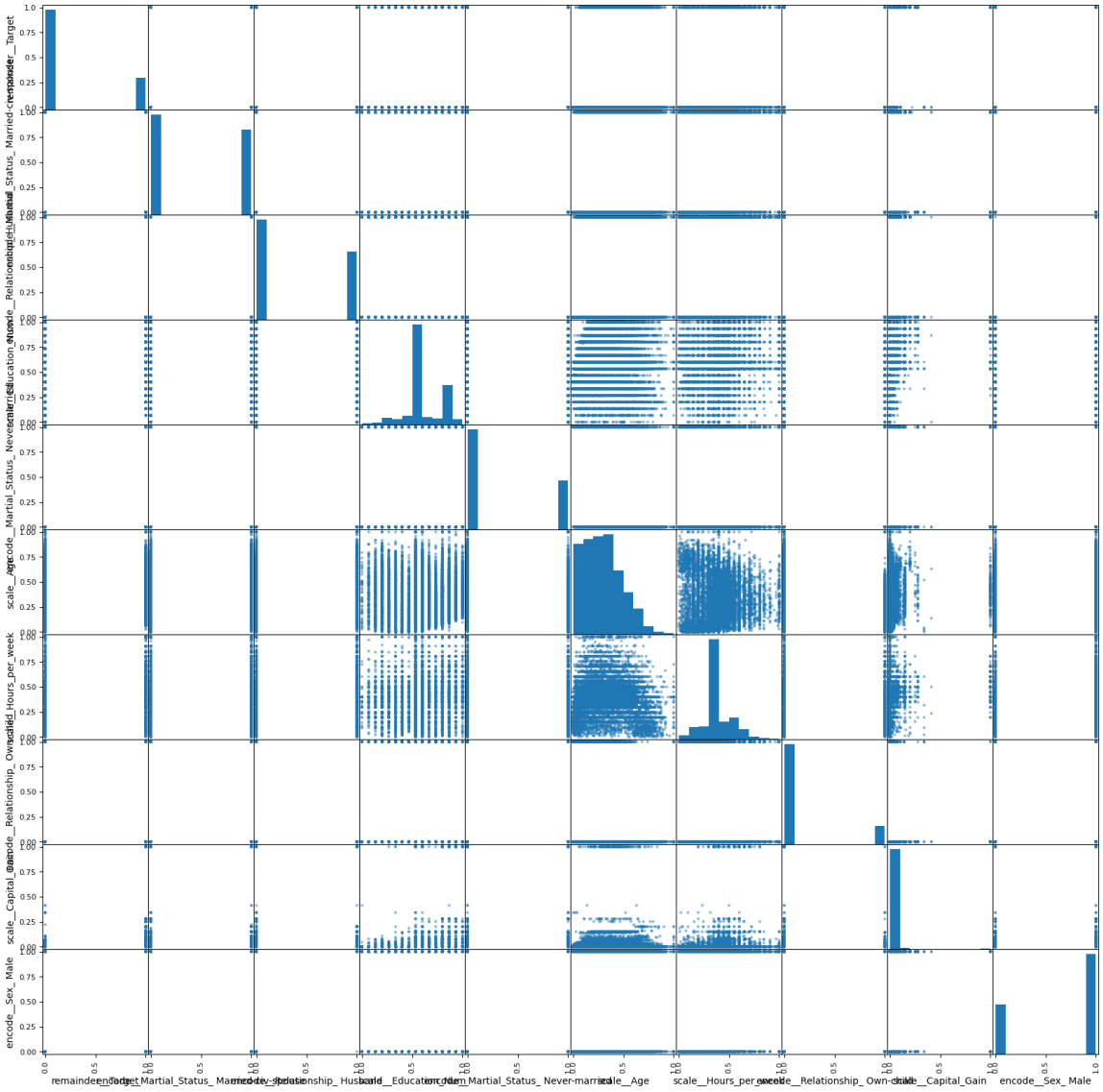
32561 rows × 109 columns

In [ ]:
```
corr_matrix = proc_data.corr()
corr_matrix['remainder__Target'].sort_values(ascending=False)
```

Out[ ]:
```
remainder__Target                            1.000000
encode__Martial_Status_ Married-civ-spouse    0.444696
encode__Relationship_ Husband                 0.401035
scale__Education_Num                          0.335154
scale__Age                                    0.234037
                                                ...
encode__Occupation_ Other-service            -0.156348
encode__Relationship_ Not-in-family          -0.188497
encode__Sex_ Female                          -0.215980
encode__Relationship_ Own-child              -0.228532
encode__Martial_Status_ Never-married        -0.318440
Name: remainder__Target, Length: 109, dtype: float64
```

Выделим самые значимые элементы:

In [ ]:
```
from pandas.plotting import scatter_matrix
imp_features = abs(corr_matrix['remainder__Target']).sort_values(ascending=False

scatter_matrix(proc_data[imp_features.keys()],figsize=(20,20))
plt.show()
```



По такому графику достаточно сложно сделать выводы о зависимостях между параметрами.

В итоге, мною были выполнены следующие действия:

- проанализированы распределения параметров
- проведена предварительная обработка данных, с обработкой пропусков и преобрзаованием категорий в числа
- построена корреляционная матрица и найдены самые важные для решения задачи параметры