****Московский государственный технический университет им. Н.Э. Баумана  
Кафедра «Системы обработки информации и управления»

Лабораторная работа №2  
по дисциплине  
«Методы машинного обучения»  
на тему

# «Обработка признаков (часть 1)»

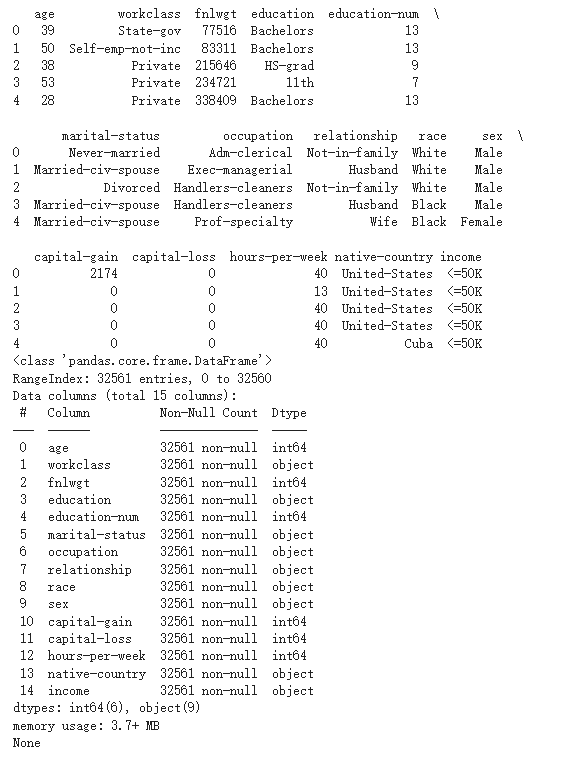
Выполнил:  
студент группы ИУ5И-22М  
Лу Жуньда

Москва — 2024 г.

1. **Цель лабораторной работы**

Изучение продвинутых способов предварительной обработки данных для дальнейшего формирования моделей.

1. **Задание**
2. Выбрать набор данных (датасет), содержащий категориальные и числовые признаки и пропуски в данных. Для выполнения следующих пунктов можно использовать несколько различных наборов данных (один для обработки пропусков, другой для категориальных признаков и т.д.) Просьба не использовать датасет, на котором данная задача решалась в лекции.
3. Для выбранного датасета (датасетов) на основе материалов лекций решить следующие задачи:
4. устранение пропусков в данных;
5. кодирование категориальных признаков;
6. нормализация числовых признаков.
7. **Текст программы**



import pandas as pd

# URL для загрузки датасета

url = "https://archive.ics.uci.edu/ml/machine-learning-databases/adult/adult.data"

# Загрузка данных

columns = [

    "age", "workclass", "fnlwgt", "education", "education-num", "marital-status",

    "occupation", "relationship", "race", "sex", "capital-gain", "capital-loss",

    "hours-per-week", "native-country", "income"

]

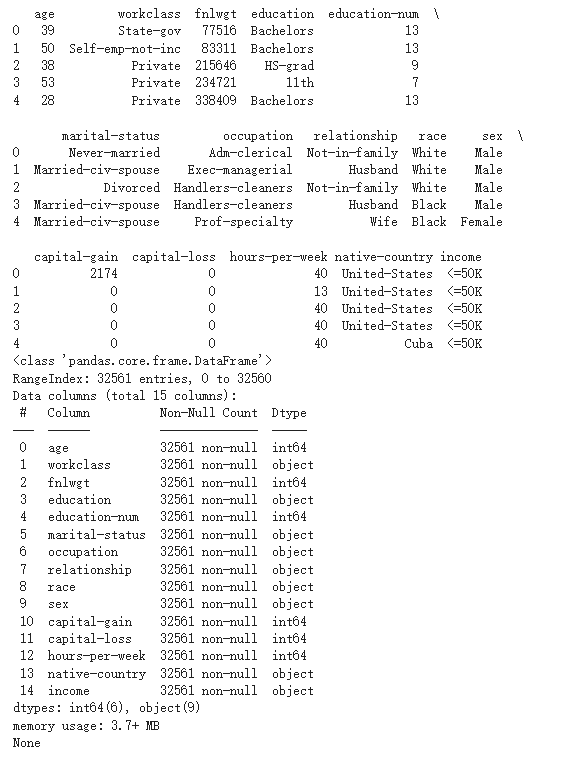
df = pd.read\_csv(url, names=columns, na\_values=" ?", skipinitialspace=True)

# Просмотр первых нескольких строк

print(df.head())

# Информация о датасете

print(df.info())



# Просмотр пропущенных значений по каждому столбцу

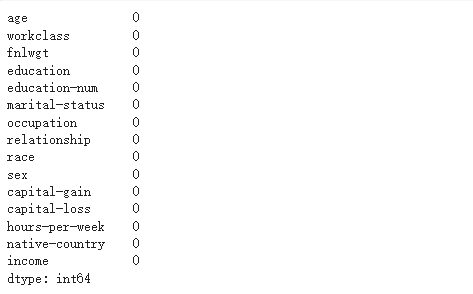
print(df.isnull().sum())

# Заполнение пропусков

df['workclass'].fillna(df['workclass'].mode()[0], inplace=True)

df['occupation'].fillna(df['occupation'].mode()[0], inplace=True)

df['native-country'].fillna(df['native-country'].mode()[0], inplace=True)



# One-Hot Encoding для категориальных признаков

df\_encoded = pd.get\_dummies(df, columns=['workclass', 'education', 'marital-status', 'occupation', 'relationship', 'race', 'sex', 'native-country'])

# Нормализация числовых признаков

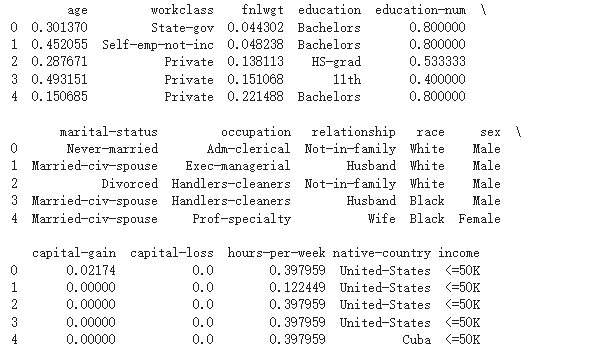
from sklearn.preprocessing import MinMaxScaler

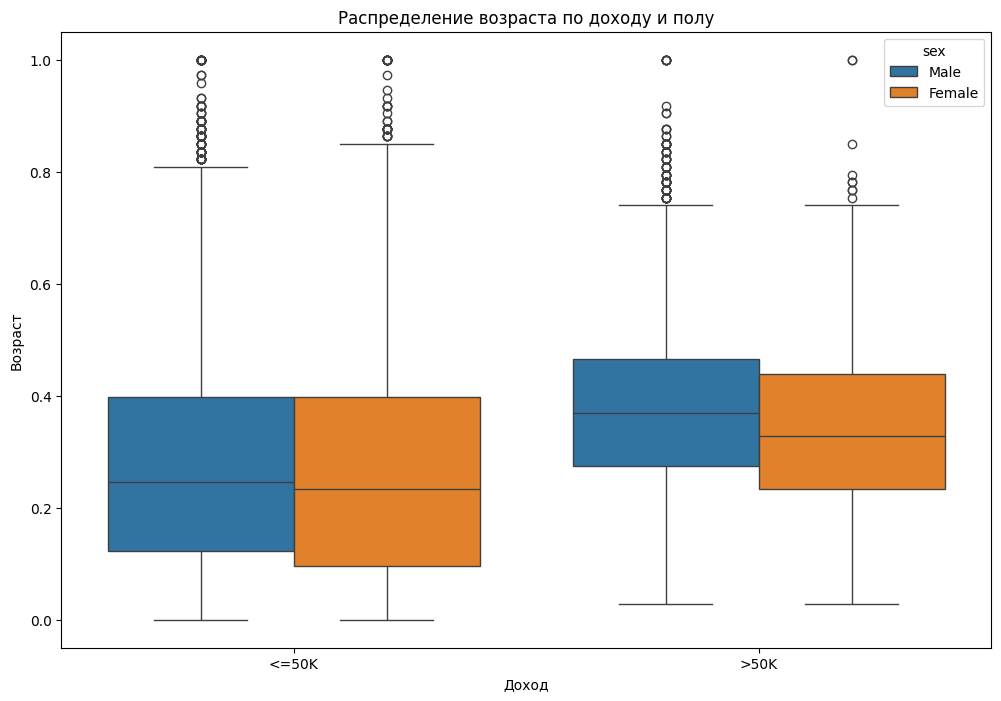
scaler = MinMaxScaler()

numerical\_features = ['age', 'fnlwgt', 'education-num', 'capital-gain', 'capital-loss', 'hours-per-week']

df[numerical\_features] = scaler.fit\_transform(df[numerical\_features])

print(df.head())





# Создание графика с распределением возраста по доходу и полу

plt.figure(figsize=(12, 8))

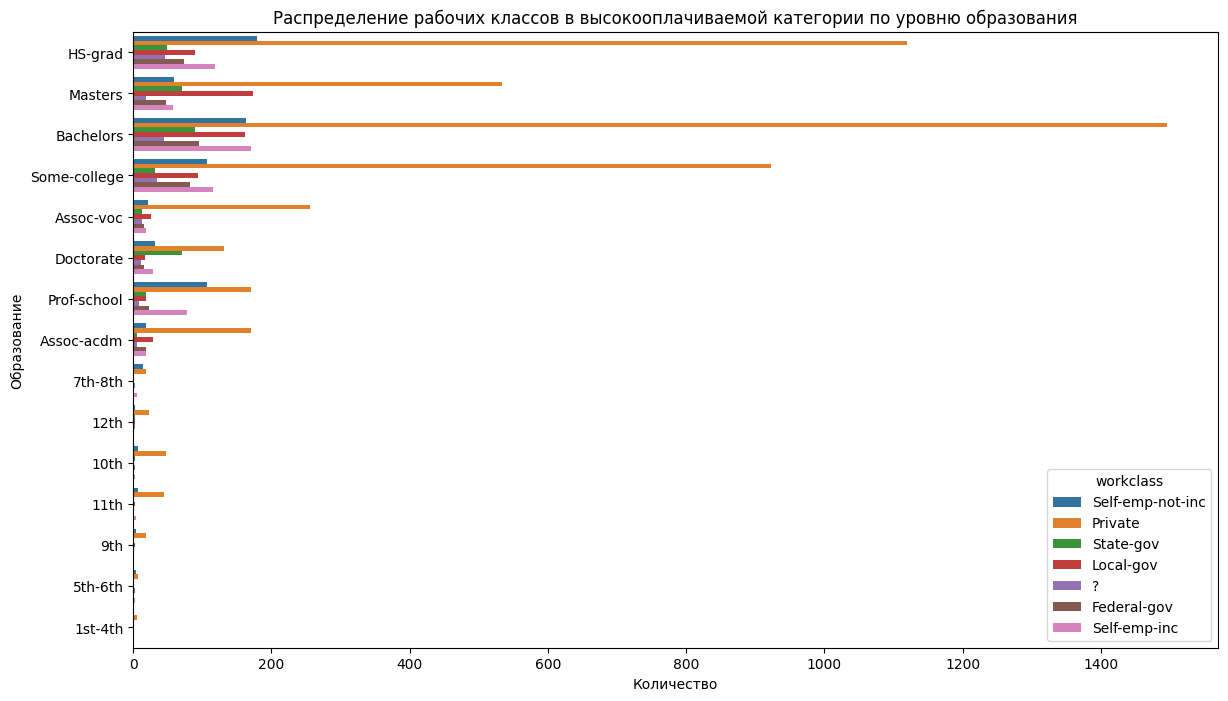
sns.boxplot(x="income", y="age", hue="sex", data=df)

plt.title('Распределение возраста по доходу и полу')

plt.xlabel('Доход')

plt.ylabel('Возраст')

plt.show()



plt.figure(figsize=(14, 8))

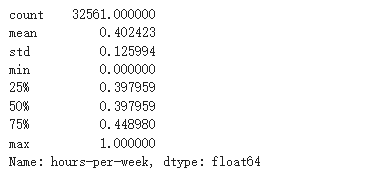
sns.countplot(y="education", hue="workclass", data=df[df['income'] == '>50K'])

plt.title('Распределение рабочих классов в высокооплачиваемой категории по уровню образования')

plt.xlabel('Количество')

plt.ylabel('Образование')

plt.show()



import matplotlib.pyplot as plt

import seaborn as sns

# Проверка, что DataFrame содержит нужный столбец и он не пустой

print(df['hours-per-week'].describe())

# Гистограмма распределения рабочих часов по доходам

plt.figure(figsize=(12, 6))

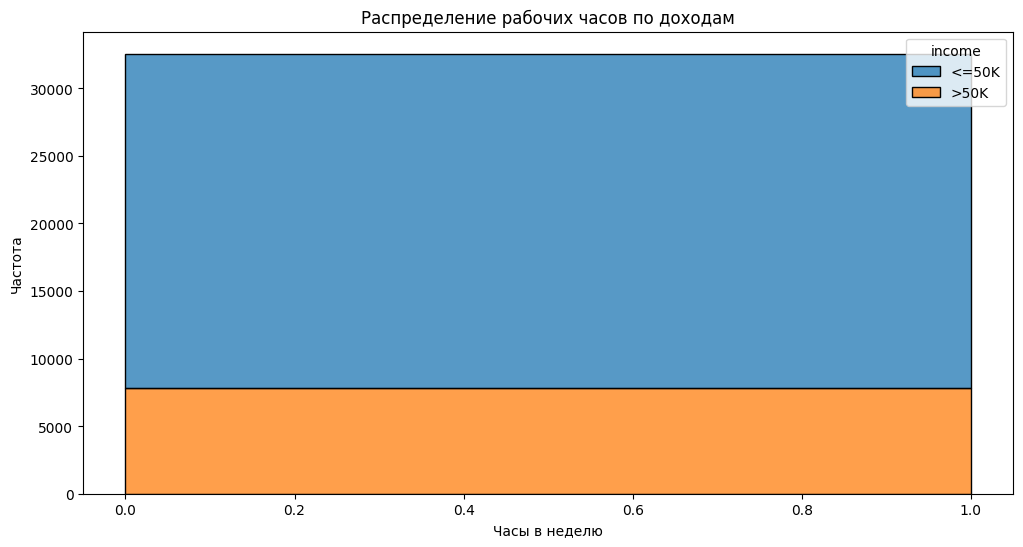
sns.histplot(data=df, x='hours-per-week', hue='income', multiple="stack", binwidth=1)  # Использование более малого и явного значения для binwidth

plt.title('Распределение рабочих часов по доходам')

plt.xlabel('Часы в неделю')

plt.ylabel('Частота')

plt.show()



# Корреляционная матрица числовых признаков

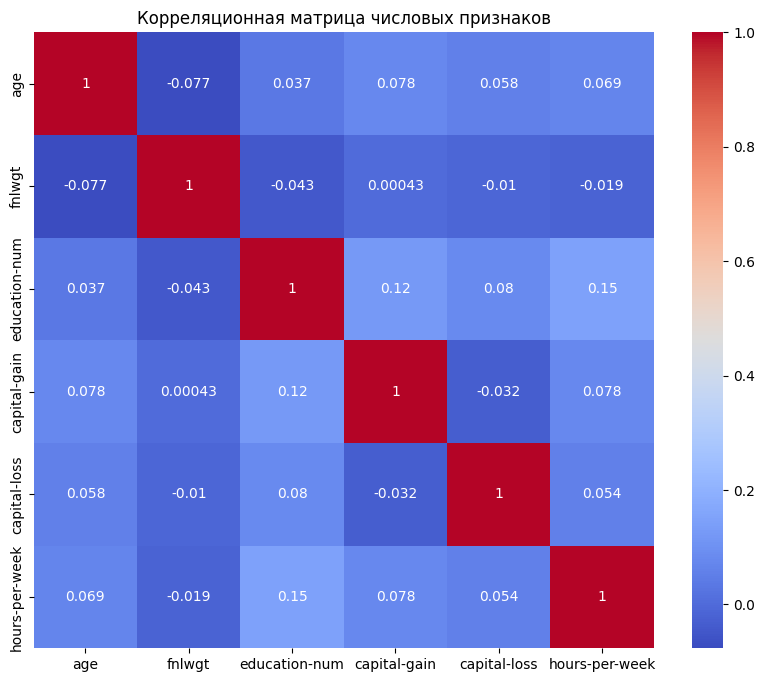
correlation = df[numerical\_features].corr()

plt.figure(figsize=(10, 8))

sns.heatmap(correlation, annot=True, cmap='coolwarm')

plt.title('Корреляционная матрица числовых признаков')

plt.show()



# Столбчатая диаграмма с возрастом и образованием как факторами дохода

g = sns.FacetGrid(df, col="income", height=5)

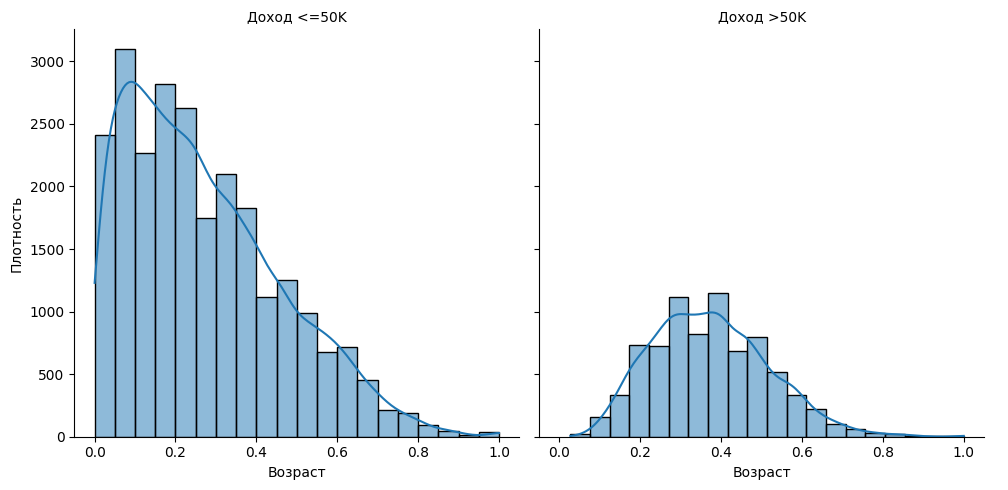
g.map\_dataframe(sns.histplot, "age", kde=True, bins=20)

g.set\_titles("Доход {col\_name}")

g.set\_axis\_labels("Возраст", "Плотность")

g.add\_legend()

plt.show()



plt.figure(figsize=(14, 10))

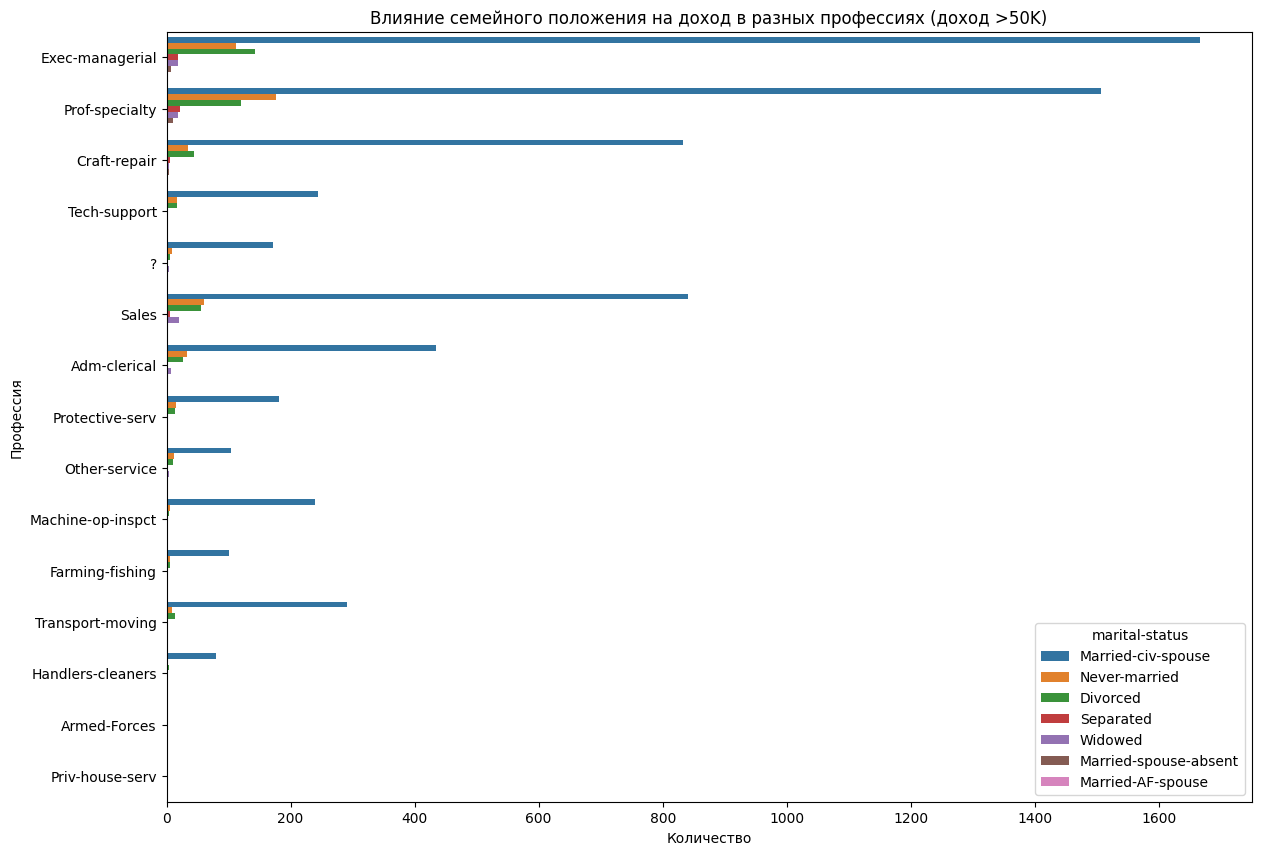
sns.countplot(y="occupation", hue="marital-status", data=df[df['income'] == '>50K'])

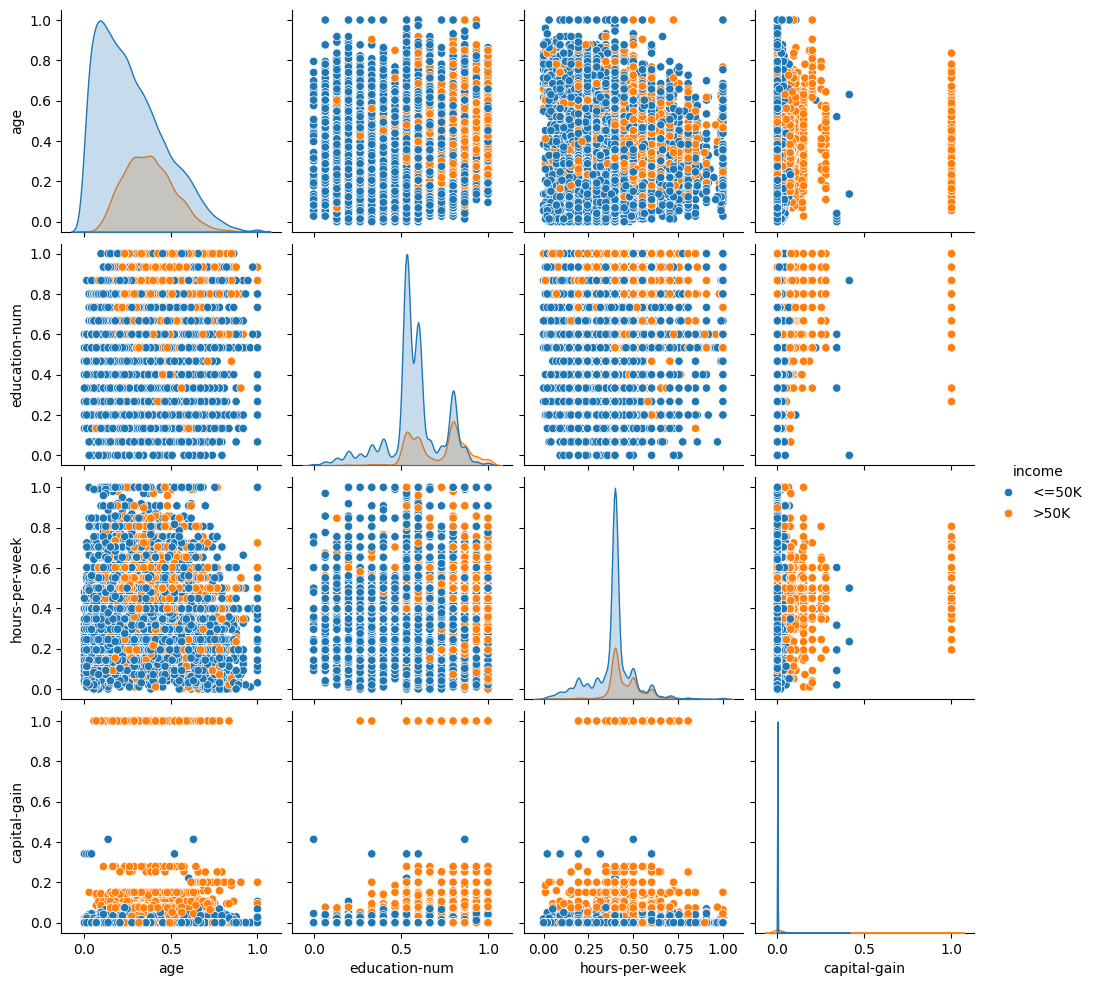
plt.title('Влияние семейного положения на доход в разных профессиях (доход >50K)')

plt.xlabel('Количество')

plt.ylabel('Профессия')

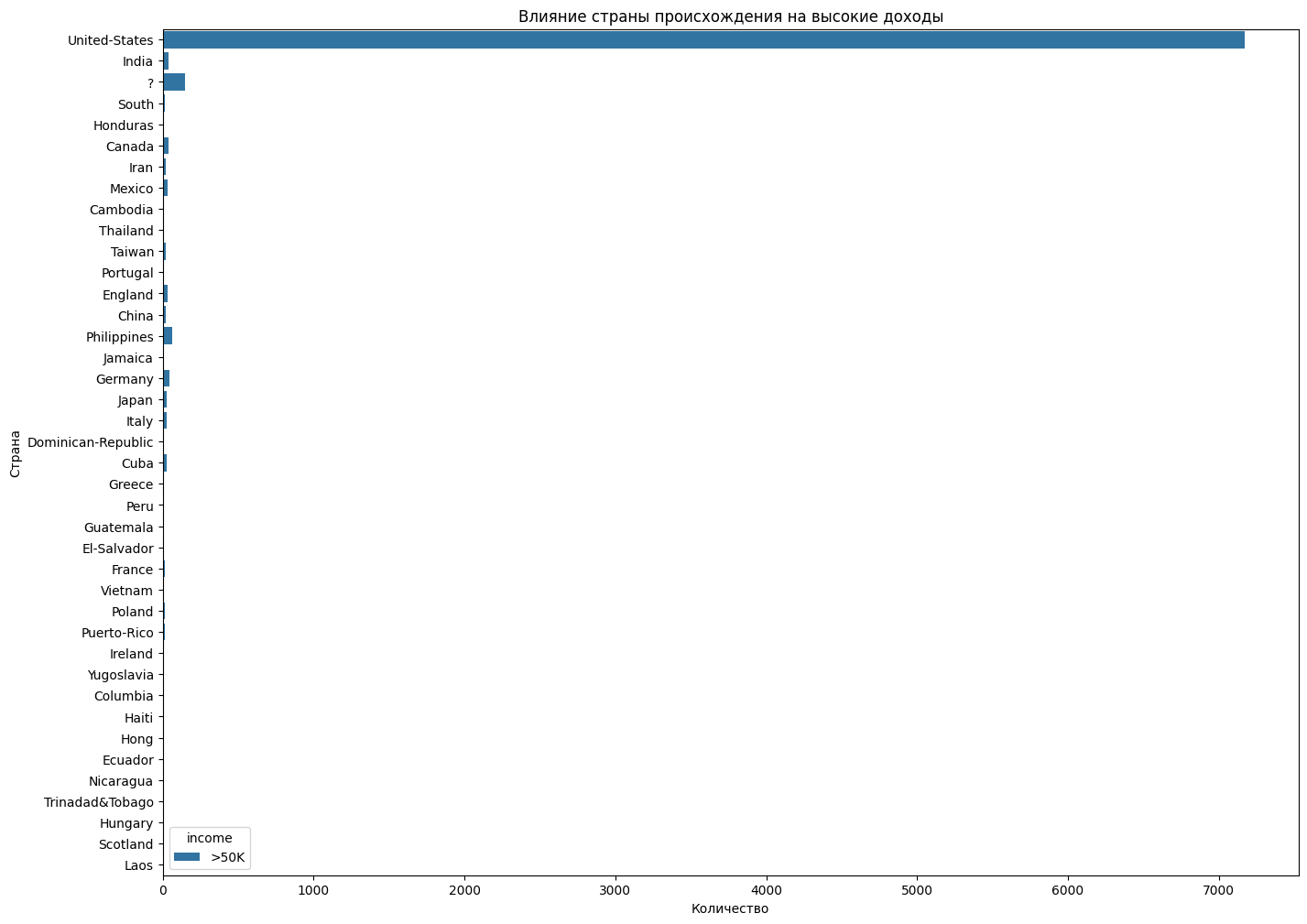
plt.show()





sns.pairplot(df, hue='income', vars=['age', 'education-num', 'hours-per-week', 'capital-gain'])

plt.show()



plt.figure(figsize=(16, 12))

sns.countplot(y='native-country', hue='income', data=df[df['income'] == '>50K'])

plt.title('Влияние страны происхождения на высокие доходы')

plt.xlabel('Количество')

plt.ylabel('Страна')

plt.show()