Министерство науки и высшего образования Российской Федерации Федеральное государственное бюджетное образовательное учреждение высшего образования «Московский государственный технический университет имени Н.Э. Баумана (национальный исследовательский университет)» (МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ Информатика и системы управления и искусственный интеллект

КАФЕДРА Системы обработки информации и управления

Лабораторная работа № 2

По курсу
«Методы машинного обучения»
На тему:
«Обработка признаков (часть 1)»

Подготовил:

Студент группы

ИУ5-25М Клюкин Н. А.

27.03.2024

Проверил:

Гапанюк Ю.Е.

2024 г.

- **Цель лабораторной работы**: изучение продвинутых способов предварительной обработки данных для дальнейшего формирования моделей.

Задание

- 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.
- Для выбранного датасета (датасетов) на основе материалов лекций решить следующие задачи:
  - устранение пропусков в данных;
  - кодирование категориальных признаков;
  - нормализация числовых признаков.

Подключение библиотек

In [1]:
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

Ход работы

Выбор и описание датасета

- Этот набор данных предоставляет полную информацию о поведении клиентов для типичной страницы туризма в социальных сетях. Датасет содержит в себе следующие поля:
  - UserID (PK) - удаляем
  - Taken_product - купил тур (да/нет)
  - Yearly_avg_view_on_travel_page - Среднегодовое количество просмотров пользователем любой страницы, связанной с путешествиями
  - preferred_device - Предпочтительное устройство для входа пользователя в систему
  - total_likes_on_outstation_checkin_given - Общее количество лайков, поставленных пользователем при регистрации вне станции за последний год
  - yearly_avg_Outstation_checkins - Среднее количество регистраций за пределами станции, выполненных пользователем
  - member_in_family - Общее количество членов семьи, упомянутых пользователем в учетной записи
  - preferred_location_type - Предпочтительный тип местоположения для перемещения пользователя

- Yearly_avg_comment_on_travel_page - Среднегодовые комментарии пользователя на любой странице, связанной с путешествиями
- total_likes_on_outofstation_checkin_received - Общее количество лайков, полученных пользователем при выезде за пределы станции за последний год
- week_since_last_outstation_checkin - Количество недель с момента последнего обновления пользователем регистрации вне станции
- following_company_page - Читает ли клиент страницу компании (Да или Нет)
- montly_avg_comment_on_company_page - Среднее количество комментариев пользователя на странице компании в месяц
- working_flag - работает или нет
- travelling_network_rating - Рейтинг, указывающий, есть ли у пользователя близкие друзья, которые также любят путешествовать. 1 — высокий, 4 — самый низкий
- Adult_flag - взрослый или нет
- Daily_Avg_mins_spend_on_traveling_page - Среднее время, проведенное пользователем на странице путешествия компании

In [3]:
```
# Импорт датасета
df = pd.read_csv('datasets/cust_beh.csv')

# Вывод первых 5 строк
df.head(5)
```

Out[3]:

	Taken_product	Yearly_avg_view_on_travel_page	preferred_device	total_likes_on_outstatior
0	Yes	307.0	iOS and Android	
1	No	367.0	iOS	
2	Yes	277.0	iOS and Android	
3	No	247.0	iOS	
4	No	202.0	iOS and Android	

In [5]:
```
# Используем только некоторые признаки
cols_filter = ['Taken_product', 'preferred_device', 'Yearly_avg_view_on_travel_page
        'travelling_network_rating']
data = df[cols_filter]
data.head()
```

Out[5]:

	Taken_product	preferred_device	Yearly_avg_view_on_travel_page	Yearly_avg_comment_on
0	Yes	iOS and Android	307.0	
1	No	iOS	367.0	
2	Yes	iOS and Android	277.0	
3	No	iOS	247.0	
4	No	iOS and Android	202.0	

Устранение пропусков в данных

In [12]:
```
hdata = data
list(zip(hdata.columns, [i for i in data.dtypes]))
```

Out[12]:
```
[('Taken_product', dtype('O')),
 ('preferred_device', dtype('O')),
 ('Yearly_avg_view_on_travel_page', dtype('float64')),
 ('Yearly_avg_comment_on_travel_page', dtype('float64')),
 ('travelling_network_rating', dtype('int64'))]
```

In [13]:
```
# Колонки с пропусками
hcols_with_na = [c for c in hdata.columns if hdata[c].isnull().sum() > 0]
hcols_with_na
```

Out[13]:
```
['preferred_device',
 'Yearly_avg_view_on_travel_page',
 'Yearly_avg_comment_on_travel_page']
```

In [14]:
```
hdata.shape
```

Out[14]:
```
(11770, 5)
```

In [15]:
```
[(c, hdata[c].isnull().sum()) for c in hcols_with_na]
```

Out[15]:
```
[('preferred_device', 53),
 ('Yearly_avg_view_on_travel_page', 581),
 ('Yearly_avg_comment_on_travel_page', 206)]
```

In [16]:
```
# Доля (процент) пропусков
[(c, hdata[c].isnull().mean()) for c in hcols_with_na]
```

Out[16]:
```
[('preferred_device', 0.004502973661852166),
 ('Yearly_avg_view_on_travel_page', 0.04936278674596432),
 ('Yearly_avg_comment_on_travel_page', 0.01750212404418012)]
```

In [18]:
```
# Колонки для которых удаляются пропуски
hcols_with_na_temp = ['preferred_device', 'Yearly_avg_view_on_travel_page', 'Yearly
```

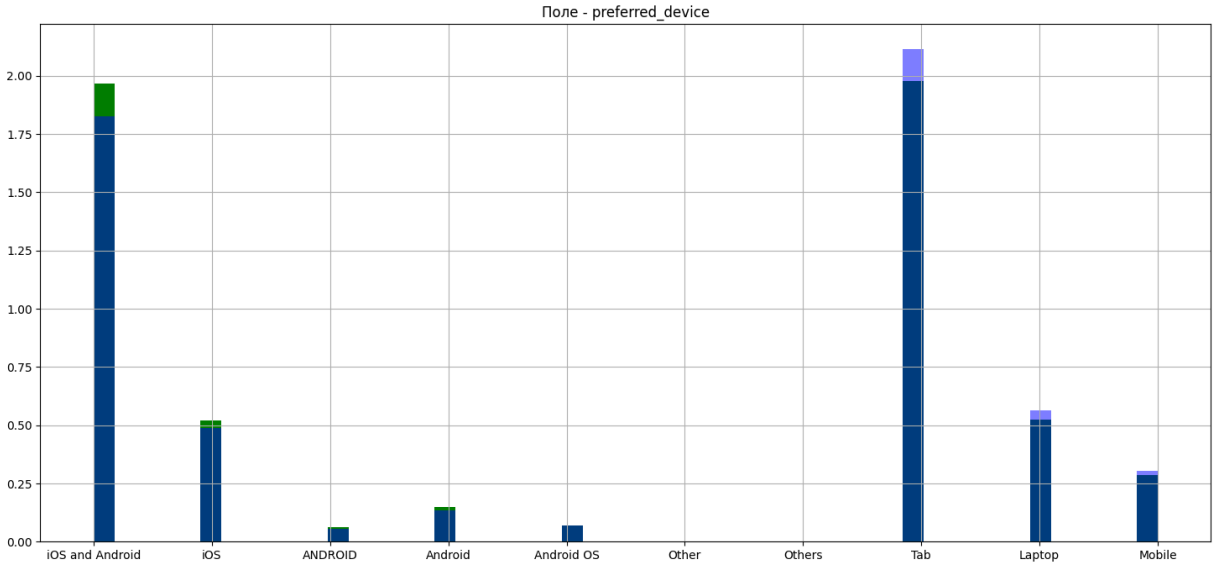
In [19]:
```
# Удаление пропусков
hdata_drop = hdata[hcols_with_na_temp].dropna()
```

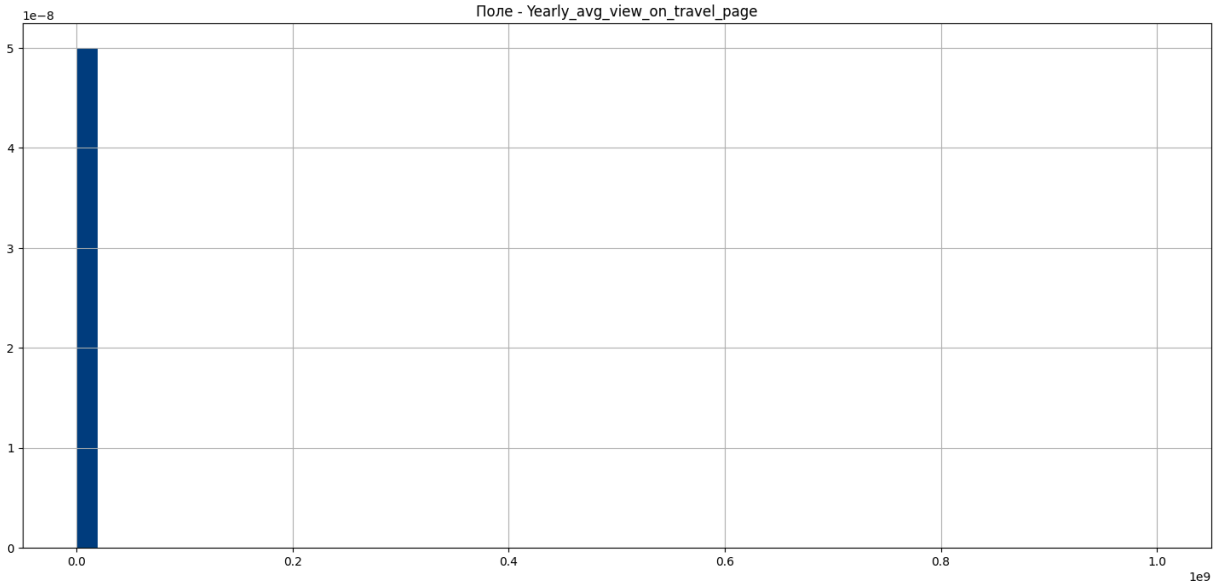
```
hdata_drop.shape
```

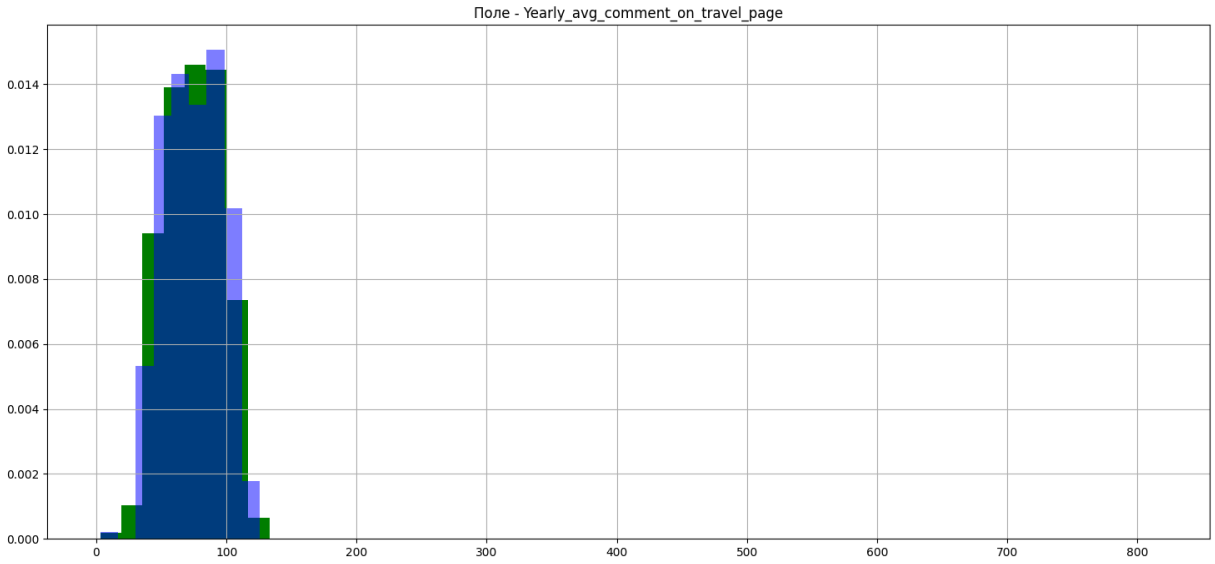
Out[19]: (10956, 3)

In [23]:
```
def plot_hist_diff(old_ds, new_ds, cols):
    """
    Разница между распределениями до и после устранения пропусков
    """
    for c in cols:
        fig = plt.figure(figsize=(18, 8))
        ax = fig.add_subplot(111)
        ax.title.set_text('Поле - ' + str(c))
        old_ds[c].hist(bins=50, ax=ax, density=True, color='green')
        new_ds[c].hist(bins=50, ax=ax, color='blue', density=True, alpha=0.5)
        plt.show()
```

In [24]:
```
plot_hist_diff(hdata, hdata_drop, hcols_with_na_temp)
```







Кодирование категориальных признаков

- Проведём кодирование категорилального признака preferred_device с использованием метода "one-hot encoding".
- One-hot encoding предполагает, что значение категории заменяется на отдельную колонку, которая содержит бинарные значения.
- Преимущества:
  - Простота реализации.
  - Подходит для любых моделей, так как НЕ создает фиктивное отношение порядка между значениями.
- Недостатки:
  - Расширяется признаковое пространство.

In [6]:
```
pd.get_dummies(data[['preferred_device']]).head()
```

Out[6]:

	preferred_device_ANDROID	preferred_device_Android	preferred_device_Android OS	preferre
0	False	False	False	
1	False	False	False	
2	False	False	False	
3	False	False	False	
4	False	False	False	

In [7]:
```
# Добавление отдельной колонки, признака пустых значений
pd.get_dummies(data[['preferred_device']], dummy_na=True).head()
```

Out[7]:

	preferred_device_ANDROID	preferred_device_Android	preferred_device_Android OS	preferre
0	False	False	False	
1	False	False	False	
2	False	False	False	
3	False	False	False	
4	False	False	False	

Нормализация числовых признаков

- Проведём нормализацию числовых признаков через использование межквартильного размаха
- Межквартильный размах IQR (interquartile range, IQR) - это разность третьего квартиля и первого квартиля:

In [8]:
```
def remove_outliers_iqr(data, column):
    Q1 = data[column].quantile(0.25)
    Q3 = data[column].quantile(0.75)
    IQR = Q3 - Q1
    lower_bound = Q1 - 1.5 * IQR
    upper_bound = Q3 + 1.5 * IQR
    filtered_data = data[(data[column] >= lower_bound) & (data[column] <= upper_bou
    return filtered_data
```

In [9]:
```
data.shape
```

Out[9]: (11770, 5)

In [10]:
```
filtered_dataset = remove_outliers_iqr(data, 'Yearly_avg_view_on_travel_page')
filtered_dataset.shape
```

Out[10]: (11168, 5)