Программа Профессиональной Переподготовки
Аналитик данных: с нуля до разработки прикладных решений для бизнеса

Итоговый проект:

Выявление признаков, которые оказывают влияние на факт закрытия кредита

Выполнил: Седаш Ольга Станиславовна
Номер потока: АнД-802
Преподаватель: Туманян Симон Рафаэлович

СОДЕЙСТВИЕ ЗАНЯТОСТИ | Федеральный проект

Цель и задачи

- Цель: Провести анализ данных с целью выделения признаков, которые оказывают влияние на факт закрытия кредита.

Задачи:

1. Загрузить dataset segment bank и провести чтение данных.
2. Выполнить предварительную обработку данных
3. Составить гипотезы о данных и выполнить их проверку
4. Полученные результаты интерпретировать в соответствии поставленной бизнес – задачей, подготовить и опубликовать дашборд.

Исходные данные:

- dataset: предоставлен преподавателем
- print(df.info()) (скрин)

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 50224 entries, 0 to 50223
Data columns (total 14 columns):
 #   Column                                        Non-Null Count  Dtype
---  ------                                        --------------  -----
 0   Идентификатор                                 50224 non-null  int64
 1   Дата рождения                                 50224 non-null  object
 2   Дисциплина клиентов без просрочки по кредиту  50223 non-null  object
 3   Количество переводов                          50158 non-null  float64
 4   Тип переводов                                 50191 non-null  float64
 5   География переводов                           50191 non-null  float64
 6   География телефона                            48324 non-null  float64
 7   Сумма перевода                                50158 non-null  float64
 8   Максимальная сумма перевода                   50158 non-null  float64
 9   Средняя сумма перевода                        50158 non-null  float64
 10  Полная сумма перевода                         50158 non-null  float64
 11  Канал, через который пришел клиент            50193 non-null  object
 12  Оператор связи                                49579 non-null  object
 13  Пол                                           50215 non-null  object
dtypes: float64(8), int64(1), object(5)
memory usage: 5.4+ MB
None
```

Предобработка данных

```python
#проверка пропусков
print(df.isnull().sum())
```

Показать скрытые выходные данные

```python
#обработка пропусков
df['Количество переводов'].fillna(df['Количество переводов'].mean(),inplace=True)
df['Тип переводов'].fillna(df['Тип переводов'].mean(),inplace=True)
df['География переводов'].fillna(df['География переводов'].mean(),inplace=True)
df['География телефона'].fillna(df['География телефона'].mean(),inplace=True)
df['Сумма перевода'].fillna(df['Сумма перевода'].mean(),inplace=True)
df['Максимальная сумма перевода'].fillna(df['Максимальная сумма перевода'].mean(),inplace=True)
df['Средняя сумма перевода'].fillna(df['Средняя сумма перевода'].mean(),inplace=True)
df['Полная сумма перевода'].fillna(df['Полная сумма перевода'].mean(),inplace=True)
df = df.dropna()
```

```python
#преобразование колонки 'дата рождения'
df['Дата рождения'] = pd.to_datetime(df['Дата рождения'],errors ='coerce')
print(df.isnull().sum())
#создание новую колонку 'возраст'
df['Возраст'] = pd.to_datetime('today').year - df['Дата рождения'].dt.year
print(df['Возраст'].head(10))
print(df['Дата рождения'].head(10))
```

Математическая статистика

4. Применены методы математической статистики для обработки данных.

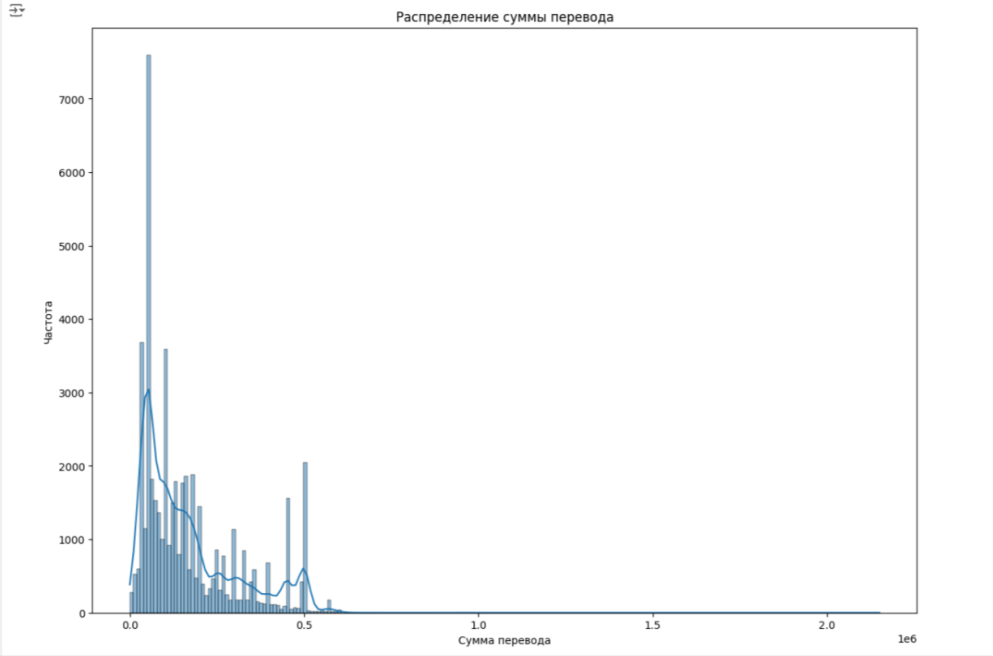
```
print(df.describe())
```

```
       Идентификатор                  Дата рождения  Количество переводов  \
count   4.954000e+04                          49540          49540.000000
mean    1.097132e+07  1970-09-18 07:57:17.222446508             17.356971
min     1.400402e+06            1940-01-20 00:00:00              1.000000
25%     6.378468e+06            1962-05-15 00:00:00              4.000000
50%     1.132780e+07            1971-06-03 00:00:00              9.000000
75%     1.444145e+07            1979-07-27 00:00:00             18.000000
max     3.006527e+07            1991-09-20 00:00:00           2220.000000
std     5.762637e+06                            NaN             34.807097

       Тип переводов  География переводов  География телефона  Сумма перевода  \
count   49540.000000         49540.000000        49540.000000    4.954000e+04
mean       13.177710           110.880475           58.138782    2.998784e+05
min        -1.000000             0.000000            0.000000    3.000000e+01
25%         2.000000            48.000000           42.000000    9.314975e+04
50%         5.000000            73.000000           64.000000    1.890235e+05
75%         6.000000            77.000000           77.000000    3.840442e+05
max        69.000000        498002.000000           78.000000    1.953691e+07
std        22.271453          5002.320434           22.277671    4.467451e+05

       Максимальная сумма перевода  Средняя сумма перевода  \
count                 4.954000e+04            49540.000000
mean                  1.681174e+05            34448.297846
min                   3.000000e+01               30.000000
25%                   5.500000e+04            10096.164025
50%                   1.200000e+05            19819.097850
75%                   2.350000e+05            41484.011825
max                   2.150000e+06           644225.000000
std                   1.411254e+05            43559.027113

       Полная сумма перевода       Возраст
count           4.954000e+04  49540.000000
mean            2.998784e+05     53.773415
min             3.000000e+01     33.000000
25%             9.314975e+04     45.000000
50%             1.890235e+05     53.000000
75%             3.840442e+05     62.000000
```

Математическая статистика

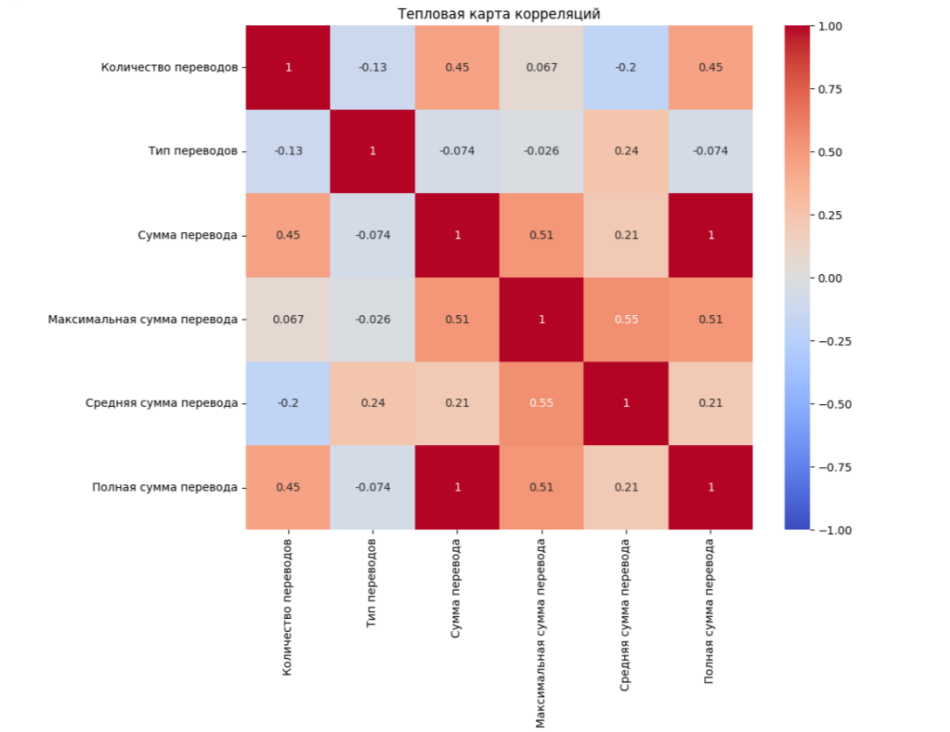
```
columns = ['Количество переводов','Тип переводов','Сумма перевода','Максимальная сумма перевода','Средняя сумма перевода','Полная сумма перевода']
corr_matrix = df[columns].corr()
group_data = df.groupby('Возраст').agg({'Сумма перевода':'mean','Максимальная сумма перевода':'mean'})
print(group_data.round(2))
df['Год рождения'] = df['Дата рождения'].dt.year
trend_data = df.groupby('Год рождения').agg({'Максимальная сумма перевода':'mean'})
print(trend_data.round(2))
```

```
#Создаем гистограмму
plt.figure(figsize=(14,10))
sb.histplot(df['Максимальная сумма перевода'],kde=True)
plt.title("Распределение суммы перевода")
plt.xlabel('Сумма перевода')
plt.ylabel('Частота')
plt.show()
```




```
plt.figure(figsize=(10,8))
sb.heatmap(corr_matrix,annot=True,cmap='coolwarm',vmin=-1,vmax=1)
plt.title("Тепловая карта корреляций")
```

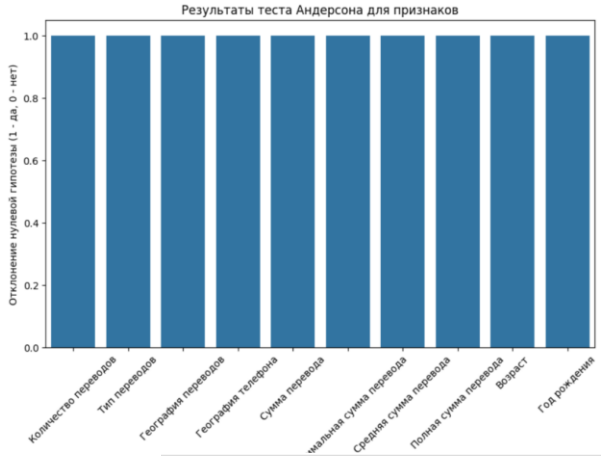
```
Text(0.5, 1.0, 'Тепловая карта корреляций')
```



Исследовательский анализ данных:

7. Составлена гипотеза о данных и выполнена проверка соответствующей гипотезы

```
[ ]  #Нулевая гипотеза (H0): Нет статистически значимых различий в суммах перевода в ворасте до 50 лет и старше 50 лет.

     target = df['Дисциплина клиентов без просрочки по кредиту']

     # Список признаков для проверки
     features = df.select_dtypes(include=['float64', 'int32']).columns.tolist()

     # Результаты теста
     results = {}

     for feature in features:
         # Выполняем тест Андерсона
         stat, critical_values, significance_level = st.anderson(df[feature])

         # Определяем, отклоняем ли мы нулевую гипотезу
         if stat > critical_values[2]:  # Используем 5% уровень значимости
             results[feature] = 'Отклоняем нулевую гипотезу'
         else:
             results[feature] = 'Не отклоняем нулевую гипотезу'

     # Вывод результатов
     for feature, result in results.items():
         print(f'Признак: {feature}, \nРезультат: {result}')

     # Визуализация результатов
     plt.figure(figsize=(10, 6))
     sb.barplot(x=list(results.keys()), y=[1 if result == 'Отклоняем нулевую гипотезу' else 0 for result in results.values()])
     plt.title('Результаты теста Андерсона для признаков')
     plt.xlabel('Признаки')
     plt.ylabel('Отклонение нулевой гипотезы (1 - да, 0 - нет)')
     plt.xticks(rotation=45)
     plt.show()
```

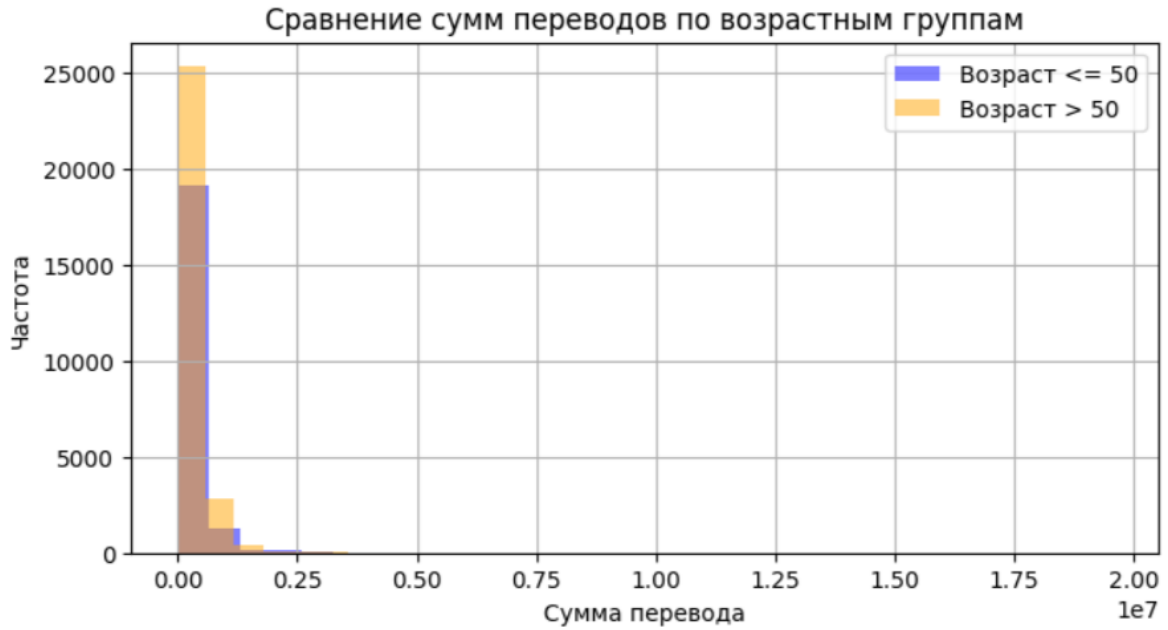



```
Признак: Количество переводов,
Результат: Отклоняем нулевую гипотезу
Признак: Тип переводов,
Результат: Отклоняем нулевую гипотезу
Признак: География переводов,
Результат: Отклоняем нулевую гипотезу
Признак: География телефона,
Результат: Отклоняем нулевую гипотезу
Признак: Сумма перевода,
Результат: Отклоняем нулевую гипотезу
Признак: Максимальная сумма перевода,
Результат: Отклоняем нулевую гипотезу
Признак: Средняя сумма перевода,
Результат: Отклоняем нулевую гипотезу
Признак: Полная сумма перевода,
Результат: Отклоняем нулевую гипотезу
Признак: Возраст,
Результат: Отклоняем нулевую гипотезу
Признак: Год рождения,
Результат: Отклоняем нулевую гипотезу
```

Исследовательский анализ данных:

```python
# Список признаков для проверки
group_a = df[df['Возраст'] <= 50]['Сумма перевода']
group_b = df[df['Возраст'] > 50]['Сумма перевода']

# Проведение теста Андерсона
result = stats.anderson_ksamp([group_a, group_b])

# Вывод результатов
print('Статистика теста:', result.statistic)
print('Критические значения:', result.critical_values)
print('Уровни значимости:', result.significance_level)

# Проверка нулевой гипотезы
results = {}
alpha = 0.05  # Уровень значимости

if result.statistic > result.critical_values[2]:
    results['Решение'] = 'Отклоняем нулевую гипотезу'
else:
    results['Решение'] = 'Не отклоняем нулевую гипотезу'

# Вывод результатов проверки гипотезы
print(results)
```

```
Статистика теста: 136.3485552549639
Критические значения: [0.325 1.226 1.961 2.718 3.752 4.592 6.546]
Уровни значимости: 0.001
{'Решение': 'Отклоняем нулевую гипотезу'}
```



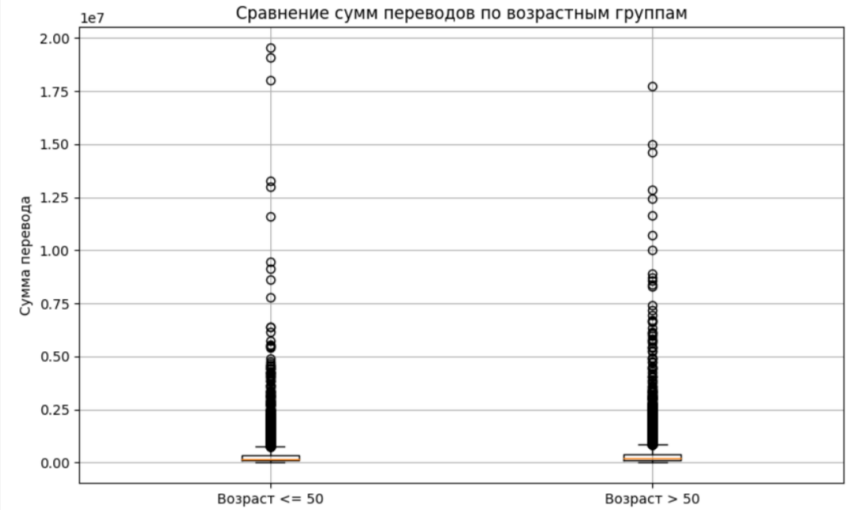
Вывод по тесту Андерсона: Отклоняем нулевую гипотезу

Исследовательский анализ данных:

Проведем Тест Манна-Уйтни, чтобы понять влияет ли возраст и сумма перевода на факт закрытия кредита

```python
# Выполнение теста Манна-Уитни
stat, p_value = st.mannwhitneyu(group_a, group_b)

# Вывод результатов
print(f'Статистика теста: {stat}')
print(f'p-значение: {p_value}')

# Интерпретация результатов
alpha = 0.05
if p_value < alpha:
    print("Имеются статестически значимые различия, между возрастами клиента.")
else:
    print("Не отклоняем нулевую гипотезу: нет достаточных оснований считать, что группы имеют разные распределения.")

# Визуализация
plt.figure(figsize=(10, 6))
plt.boxplot([group_a, group_b], labels=['Возраст <= 50', 'Возраст > 50'])
plt.title('Сравнение сумм переводов по возрастным группам')
plt.ylabel('Сумма перевода')
plt.grid()
plt.show()
```

```
Статистика теста: 276362008.5
p-значение: 5.44509588882173e-46
Имеются статестически значимые различия, между возрастами клиента.
```



Результаты и выводы:

В зависимости от результатов обоих тестов, можно сделать обоснованные выводы о различиях в суммах переводов между клиентами разного возраста. Если оба теста указывают на наличие различий, это может быть важным для дальнейшего анализа клиентской базы и разработки стратегии взаимодействия с различными возрастными группами.

Дашборд

➜Опубликованный дашборд
https://datalens.yandex/53n28b1nbvi8s

Спасибо
за внимание!