Задание\ https://abcontest.matemarketing.ru/#rec744235402

Видео\ https://www.youtube.com/watch?v=z6DSVeWyVIk

In [72]:
```
import numpy as np
import pandas as pd
```

Расмотрим выборку

In [73]:
```
file_path = r"Контест_Аха_Самокат_тех_данные_по_мошенникам.csv"
```

In [74]:
```
df = pd.read_csv(file_path, sep=';',  low_memory=False)
df.columns = [x.lower() for x in df.columns]
print('Размер: ', df.shape)
df.head(1)
```

```
Размер:  (35000, 5)
```

Out[74]:

	registration_date	activation_date	merchant_id	type	ind_frod
0	16.12.2023	24.12.2023	1	IE	0.0

Предобработка данных

Приведем данные к типу DATE

Дата регистрации и дата активации

In [75]:
```
df['registration'] = pd.to_datetime(df['registration_date'], format='%d.%m.%Y')\
                                    .dt.normalize()

df['activation'] = pd.to_datetime(df['activation_date'], format='%d.%m.%Y')\
                                    .dt.normalize()
```

Введем дополнительные метрики

Месяц регистрации

In [76]:
```
df['registration_m'] = df['registration'].dt.strftime('%Y-%m-01')
```

Индикатор активации

In [77]:
```
df['activation_ind'] = np.where(df['activation_date'].notnull(), 1, 0)
```

Время активации как разница между регистрацией и активацией выраженный в днях

In [78]:
```
df['activation_time'] = (df['activation'] - df['registration']).dt.days
```

Рассмотрим распределение регистраций/активаций по месяцам

> Данные распределены равномерно. Есть аномальные значения за 1970-01-01. Вероятно сбой процесса регистрации, проставление дефолтного значения. Доля незначительная. Далее исключим данные строки из сета.

In [79]:
```
df.groupby('registration_m', dropna=False)\
    .agg({'registration': 'count', 'activation': 'count', 'ind_frod': 'sum'})\
    .reset_index().head(2)
```

Out[79]:

	registration_m	registration	activation	ind_frod
0	1970-01-01	175	86	14.0
1	2023-01-01	2888	1717	206.0

Пропущенные значения

Для ind_frod есть 700 строк без значений. Вероятно старая модель не определила индекс. Доля незначительная. Исключим данные строки из сета.

In [80]:
```
df.ind_frod.isna().sum()
```

Out[80]:
```
700
```

Рассмотрим распределение времении активации выраженную в днях, как разницу между датой активации и регистрации\ Применяем фильтры на предыдущих шагах по регистрации и активации\ Есть отрицательные значения. 19 строк. Исключим данные строки из сета.

In [81]:
```
df[(df['activation']\
        .notnull()) & (df['registration_date'] != '1970-01-01') & (df['activation_time']
        .registration.count()
```

Out[81]:
```
19
```

Фильтр. Исключаем выбросы.

Сет после применения фильтров. Количество строк: 35000 -> 34108

In [82]:
```
df_f = df[
    (df['registration_m'] != '1970-01-01') &
    (df['ind_frod'].notnull()) &
    ((df['activation_time'].isnull()) | (df['activation_time'] > 0))
]
```

1. Методология и дизайн теста

Основная метрика (дополнительные метрики) и принцип разделения на группы.

Расмотрим статистику для определения возможных метрик для тестирования\ Сгруппируем данные по форме организации бизнеса, выведим следующие признаки:

1. total_count - общее количество продавцов
2. fraud_count - количество мошенников
3. activation_sum - количество активаций
4. fraud_activation_sum - количество активаций мошенников
5. avg_activation_time - среднее время активации в днях
6. fraud_avg_activation_time - среднее время активации мошенников в днях

In [84]:
```
df1 = df_f.groupby('type')['merchant_id'].nunique().reset_index(name='total_count')
```

```
df2 = df_f[df_f['ind_frod'] == 1].groupby('type')['merchant_id']\
                                 .nunique().reset_index(name='fraud_count')
df3 = df_f[df_f['activation_ind'] == 1].groupby('type')['activation_ind']\
                                 .sum().reset_index(name='activation_sum')
df4 = df_f[(df_f['activation_ind'] == 1) & (df_f['ind_frod'] == 1)]\
                                 .groupby('type')['activation_ind'].sum()\
                                 .reset_index(name='fraud_activation_sum')
df5 = df_f.groupby('type')['activation_time'].mean()\
                                 .reset_index(name='avg_activation_time')
df6 = df_f[df_f['ind_frod'] == 1].groupby('type')['activation_time']\
                                 .mean().reset_index(name='fraud_avg_activation_time')

result = pd.merge(df1, df2, on='type', how='left')
result = pd.merge(result, df3, on='type', how='left')
result = pd.merge(result, df4, on='type', how='left')
result = pd.merge(result, df5, on='type', how='left')
result = pd.merge(result, df6, on='type', how='left')
```

Вычисляем сумму для всех столбцов

In [ ]:
```
total_row = {
    'type': 'Total',
    'total_count': result['total_count'].sum(),
    'fraud_count': result['fraud_count'].sum(),
    'activation_sum': result['activation_sum'].sum(),
    'fraud_activation_sum': result['fraud_activation_sum'].sum(),
    'avg_activation_time': result['avg_activation_time'].mean(),
    'fraud_avg_activation_time': result['fraud_avg_activation_time'].mean()
}
result = result.append(total_row, ignore_index=True)
```

Расчитаем процент

In [87]:
```
result['fraud_count_%'] = round(100 * result['fraud_count']\
                                / result['total_count'],2)
result['activation_sum_%'] = round(100 * result['activation_sum']\
                                / result['total_count'],2)
result['fraud_activation_sum_%'] = round(100 * result['fraud_activation_sum']\
                                / result['total_count'],2)
```

Результирующая таблица

Пройдемся по воронке продавцов выборки - 34108.\ 8,24% от от всех продавцов модель определяет как мошенник.\ Больший процент определяется для продавцов по форме 'ИП' 9,61%, меньше 4,69% для формы 'ООО'.\ Считаем, что человек и модель определяют фрод безошибочно.\ Новая модель будет считатья лучше старой, когда данный процент будет выше 8,24%.

58,04% от всех продавцов получают статус активации.\ Дополнителья метрика качества новой модели, это процент активации мошенников.\ Сейчас 4,69%, новая модель считается лучше если выдаст процент меньше данной цифры.

In [88]:
```
result.iloc[:, :6].head()
```

Out[88]:

	type	total_count	fraud_count	activation_sum	fraud_activation_sum	avg_activation_time
0	IE	24583	2363	14256	1362	5.986813
1	LLC	9525	447	5541	237	6.091500
2	Total	34108	2810	19797	1599	6.039156

In [92]:
```
result.iloc[:, 6:].head()
```

Out[92]:

	fraud_avg_activation_time	fraud_count_%	activation_sum_%	fraud_activation_sum_%
0	5.864170	9.61	57.99	5.54
1	6.278481	4.69	58.17	2.49
2	6.071326	8.24	58.04	4.69

1.1 Формулирование гипотезы

Нулевая гипотеза (H0):\ **H0**: Доля мошенников, выявленных новой моделью, не больше доли мошенников, выявленных старой моделью.

Альтернативная гипотеза (H1):\ **H1**: Доля мошенников, выявленных новой моделью, больше доли мошенников, выявленных старой моделью.

Дополнительная метрика (доля активаций мошенников)

Нулевая гипотеза (H0):\ **H0**: Доля активаций мошенников в тестовой группе (новая модель) не больше доли активаций мошенников в контрольной группе (старая модель).

Альтернативная гипотеза (H1):\ **H1**: Доля активаций мошенников в тестовой группе (новая модель) больше доли активаций мошенников в контрольной группе (старая модель).

1.2 Определение необходимого количества наблюдений и продолжительности теста

Остановимся на одностороннем тесте\ При использовании одностороннего теста проверяем, является ли доля мошенников в тестовой группе больше, чем в контрольной группе.\ Это позволяет увеличить мощность теста или уменьшить необходимый размер выборки при сохранении того же уровня значимости.

In [ ]:
```
Проведем расчет мощности теста, который учитывает несколько параметров:
Входные данные:
baseline = 0.0824 - доля мошенников определенные моделью на исторических данных
MDE = 0.03 - минимальный эффект (абсолют)
alpha = 0.05 - уровень значимости
power = 0.8 - мощность теста
```

Расчет размера выборки

In [347…
```
from statsmodels.stats.power import NormalIndPower

analysis = NormalIndPower()
effect_size = MDE / ((baseline * (1 - baseline)) ** 0.5)
sample_size_per_group = analysis.solve_power(effect_size=effect_size, alpha=alpha,
                                             power=power, alternative='larger')

print(f"Необходимое количество наблюдений на каждую группу: {int(sample_size_per_group)}")
```

```
Необходимое количество наблюдений на каждую группу: 1038
```

Расчет продолжительности теста

In [353…
```
total_observations_needed = 2 * sample_size_per_group
registrations_per_month = 2900 # цифра на основе данных выборки, среднее по регистрациям

# Расчет времени в днях
months_needed = int(round(total_observations_needed / registrations_per_month * 30, 0))
print(f"Необходимое количество дней для проведения теста: {months_needed}")
```

```
Необходимое количество дней для проведения теста: 21
```

1.3 Сбор данных

Применение моделей:\ В контрольной группе используем старую модель для определения мошенников.\ В тестовой группе используем новую модель для определения мошенников.

Сбор данных:\ Сбор данных по мере регистрации и активации продавцов в обеих группах.\ Проверяем, что группы сбалансированы по ключевой характеристике, по форме бизнес-организации (IE и LLC).

1.4 Проведение теста

Основная метрика (доля регистрации мошенников)

Для проверки гипотезы используем Z-тест для пропорций, так как мы сравниваем доли мошенников между двумя группами.

Определение параметров

Количество мошенников в контрольной группе**:** `cnt_frod_control` **\ Количество мошенников в тестовой группе**: `cnt_frod_test` \ Общее количество наблюдений в контрольной группе**:** `cnt_control` **\ Общее количество наблюдений в тестовой группе**: `cnt_test`

In [ ]:
```
from statsmodels.stats.proportion import proportions_ztest
```

Количество мошенников и общее количество наблюдений в каждой группе

In [ ]:
```
cnt_frod_control = df[(df['group'] == 'A') & (df['ind_frod'] == 1)]\
                                           ['merchant_id'].nunique()
cnt_frod_test = df[(df['group'] == 'B') & (df['ind_frod'] == 1)]\
                                           ['merchant_id'].nunique()
cnt_control = df[df['group'] == 'A']['merchant_id'].nunique()
cnt_test = df[df['group'] == 'B']['merchant_id'].nunique()
```

Проведение Z-теста для пропорций (односторонний тест)

In [ ]:
```
stat, p_value = proportions_ztest([cnt_frod_test, cnt_frod_control],
                                  [cnt_test, cnt_control], alternative='larger')
print(f"P-значение: {p_value}")
```

Дополнительная метрика (доля активаций мошенников)

Для проверки дополнительной гипотезы используем аналогичный подход,\ сравнивая доли активаций мошенников между группами.

Количество активаций мошенников и общее количество наблюдений в каждой группе

In [ ]:
```
cnt_activation_control = df[(df['group'] == 'A') & \
                            (df['activation_ind'] == 1) & (df['ind_frod'] == 1)]\
                            ['activation_ind'].sum()
cnt_activation_test = df[(df['group'] == 'B') & (df['activation_ind'] == 1) & \
                         (df['ind_frod'] == 1)]['activation_ind'].sum()
```

Проведение Z-теста для пропорций (односторонний тест)

In [ ]:
```
stat_activation, p_value_activation = proportions_ztest([cnt_activation_test, cnt_activa
print(f"P-значение (активации): {p_value_activation}")
```

1.5 Выводы

1. Если p-значение < 0.05:\ Новая модель показывает значительное улучшение в выявлении мошенников по сравнению со старой моделью.\ Рекомендуется внедрить новую модель.
2. Если p-значение ≥ 0.05:\ Новая модель не показывает значительного улучшения в выявлении мошенников.\ Возможно, потребуется дополнительная оптимизация модели или пересмотр гипотезы.

Дополнительная метрика (доля активаций мошенников):

1. Если p-значение < 0.05:\ Новая модель успешно увеличивает долю активаций мошенников по сравнению со старой моделью.\ Это дополнительное подтверждение эффективности новой модели.
2. Если p-значение ≥ 0.05:\ Новая модель не увеличивает долю активаций мошенников значительным образом.\ Возможно, потребуется дополнительный анализ и оптимизация модели.

1.6 Дополнительно

Сокращение времени проведения A/B-теста

1. Увеличение размера выборки\ Увеличение размера выборки позволяет привлечь больше участников за короткий период, это поможет сократить время теста.
2. Использование одностороннего теста\ Односторонний тест требует меньшего размера выборки для достижения той же мощности, что и двусторонний тест.
3. Использование исторических данных Использование исторических данных для предварительного анализа может помочь понять текущие тенденции и сократить время тестирования.\ Например вычислить baseline основной метрики.
4. Адаптивный метод тестирования Адаптивный метод, Sequential Testing, позволяет прекратить тестирование раньше, если достигаются значимые результаты.

Уменьшение дисперси

Снижение дисперсии позволяет уменьшить необходимый размер выборки и повысить статистическую мощность теста.

1. Стратифицированная рандомизация\ Стратифицированная рандомизация позволяет разделить участников на подгруппы (страты) на основе ключевых характеристик перед случайным распределением в контрольную и тестовую группы.\ Это помогает обеспечить равномерное распределение участников и уменьшить дисперсию.

2. Удаление выбросов\ Удаление аномальных значений (выбросов) из данных может помочь уменьшить дисперсию и сделать данные более однородными.

1.7 Заключение

Проведен анализ исходной выборки. Подготовлены данные, удалены выбросы.\ Для проверки новой модели выбрана основная метрика - доля регистрации мошенников и дополнительная - доля активированных мошенников.\ Расчитаны необходимый размер группы - 1038 и продолжительность теста 21 день. \ Использование одностороннего теста позволяет увеличить мощность теста.\ Для проверки гипотезы используем Z-тест для пропорций.\ Расчитываем p-значение для основной и дполнительной метрики. Делаем выводы об эффективности новой модели.\ Дополнительно приведены меры по уменьшению дисперсии и сокращению времени теста.

2. Ответы на продуктовые вопросы

1. Как определить, какой продавец мошенник, а какой — нет? Какие ещё могут быть схемы мошенничества?

Строгая верификация продавцов: Улучшение процесса проверки продавцов при регистрации, включая проверку документов и дополнительных данных.

На первом этапе регистрации включить проверку не только номера ИНН, но и самого документа скана ИНН для подтверждения наличия данного документа. Процесс возможно автоматизировать. Данный признак позволит исключить тех, кто не умеет фоторедактивровать/подделывать документы и тех у кого их нет в наличии.

Использование трекеров поведения мошенников: Проведение анализа действий продавцов и покупателей на платформе.

Признаки поведения мошенника:

1. Высокая скорость добавления карточек.
2. Низкая стоимость по сравнению с аналогичными товарами
3. Короткий срок между создланием карточки товара и предоставлением скидки
4. Низкий LIFETINE продавца - дата регистрации продавца, сколько дней на маркете
5. Часто отправляет сообщения продавцам на внешние ресурсы

Еще схема мошенничества - **Фальшивые отзывы и рейтинги**:

- **Положительные отзывы:** Мошенники могут создавать фальшивые учетные записи для написания положительных отзывов о своих товарах и повышения рейтинга.
- **Негативные отзывы:** Мошенники могут оставлять негативные отзывы о товарах конкурентов, чтобы уменьшить их рейтинг и привлечь покупателей к своим товарам.

2. Какие продуктовые фичи могут помочь нашим клиентам избежать неприятных ситуаций с мошенничеством?

Коммуникация и информирование покупателя:

- **Уведомления о подозрительных действиях:** Всплывающее окно предупреждение, что уходим с маркета на внешние непроверенные ресурсы. Будьте внимательны.
- **Предупреждения о скидках:** Всплывающее окно по определению цены по сравнению с товарами одной категории/вида. Предупреждение что цена находится в аномально низкой красной зоне.

Верификация продавцов:

- **Значок "Проверенный продавец":** Отличительный значок для проверенных продавцов, прошедших многоуровневую верификацию.
- **Информация о продавце:** Показ подробной информации о продавце, включая дату регистрации, количество продаж, рейтинг и отзывы.

Форма подозрение:

- **Проверим продавца за Вас":** Простая форма для пользователей, позволяющая сообщать о подозрительных сообщениях и ссылках. Далее проверка продавца вручную или моделью на стороне Маркета.

3. Через какую механику мошенник узнает контакты покупателя? Что можем сделать, чтобы усложнить жизнь фродерам?

1. Украденные базы данных регистраций покупателей.
2. В личных сообщения с покупателем.

- **Мониторинг сообщений**: Внедрить системы анализа сообщений между продавцами и покупателями на предмет подозрительных предложений о переходе на внешние сайты.
- **Обучение пользователей**: Информировать пользователей о том, что все транзакции и общение должны происходить исключительно через платформу маркетплейса.
- **Фальшивые сайты**: Мошенники создают поддельные сайты, которые выглядят как настоящие маркетплейсы, и убеждают пользователей вводить свои контактные данные.