Тестовое задание на позицию Junior Analyst в Topface Media

**1. Вопросы:**

**1) Уровень владения Python:**

Оценка: 8/10

Обоснование: Я оцениваю свой уровень Python как 8, потому что я уверенно использую его для решения задач по анализу данных, написанию скриптов, работы с библиотеками, такими как pandas, numpy, matplotlib и scikit-learn. У меня есть опыт работы с автоматизацией задач и написанием тестов, но у меня меньше опыта в глубоком программировании (например, создание сложных API или написание больших приложений).

**2) Уровень владения Google Sheets:**

Оценка: 7/10

Обоснование: Я оцениваю свой уровень владения Google Sheets как 7, потому что я хорошо знаком с основными функциями, такими как формулы, создание диаграмм, условное форматирование и базовая обработка данных. Однако, я использую более продвинутые функции, такие как сценарии Google Apps Script, реже.

**2. Задачи на логику:**

**1) Задача 1:**

Рекламная кампания началась с дневным бюджетом 40 $. Половина бюджета была израсходована к полудню, а 80% оставшегося бюджета было потрачено после полудня. Сколько долларов не было потрачено?

Решение:

Изначальный бюджет: 40 $

Потрачено к полудню: 40 / 2 = 20 $

Осталось после полудня: 20 $

Потрачено после полудня: 20 \* 0.8 = 16 $

Осталось непотраченным: 20 - 16 = 4 $

Ответ: 4 $

**2) Задача 2:**

5 рекламных кампаний работали 24 дня по 6 часов в день и потратили 120 $. Сколько дней они работали бы на 216 $, если 9 таких же кампаний работали бы по 8 часов в день?

Решение:

Общее количество часов для 5 кампаний: 5 \* 24 \* 6 = 720 часов

Стоимость одного часа работы всех 5 кампаний: 120 / 720 = 0.1667 $/час

Общее количество часов для 9 кампаний: 216 / 0.1667 = 1296 часов

Количество дней: 1296 / (9 \* 8) = 18 дней

Ответ: 18 дней

**3) Задача 3:**

200 баннеров, 80 не использовались, 60 использовались только в кампании No1. Сколько баннеров использовалось в обеих кампаниях?

Решение:

Использованных баннеров всего: 200 - 80 = 120

Использованные только в No1: 60

Пусть x - количество баннеров, использованных в обеих кампаниях. Тогда баннеров, использованных только в No2: 3x.

Составляем уравнение: 60 + 3x + x = 120

Решение: 4x = 60, x = 15

Ответ: 15 баннеров использовалось в обеих кампаниях.

**4) Задача 4:**

Энн использует Instagram и не Facebook, Джон - Youtube и Facebook, Кейт - Youtube и не Instagram, Том - Facebook и не Youtube. Чьи предпочтения совпадают?

Решение:

Энн: Instagram, не Facebook = Youtube

Джон: Youtube, Facebook

Кейт: Youtube, не Instagram = Facebook

Том: Facebook, не Youtube = Instagram

Ответ: Джон, Кейт .

**5) Задача 5:**

Средний итоговый балл стажера по 4 модулям - 78 баллов. Сколько баллов должен получить стажер за 5-й модуль, чтобы средний балл составил 80?

Решение:

Общая сумма баллов за 4 модуля: 78 \* 4 = 312

Пусть x - необходимый балл за 5-й модуль. Тогда:

(312 + x) / 5 = 80

Решаем уравнение:

312 + x = 400 => x = 88

Ответ: 88 баллов.

**6) Задача 6:**

Автомобиль проезжает 260 км со средней скоростью 80 км/ч, на обратном пути - со средней скоростью 100 км/ч. Насколько быстрее был обратный путь?

Решение:

Время на первый путь: 260 / 80 = 3.25 часа

Время на обратный путь: 260 / 100 = 2.6 часа

Разница во времени: 3.25 - 2.6 = 0.65 часа = 0.65 \* 60 = 39 минут

Ответ: Обратный путь был быстрее на 39 минут.

Вопросы:

1)Общая выручка за июль 2021: 757830.74 руб.

2)



3)Менеджер, привлекший больше всего средств в сентябре 2021: Петрова, сумма: 219240.12 руб.

4)Преобладающий тип сделок в октябре 2021: текущая

5)Количество оригиналов договоров по майским сделкам, полученных в июне 2021: 76

import pandas as pd

# Загрузка данных

file\_path = 'data.xlsx'

data = pd.read\_excel(file\_path)

# Очистка данных

data\_cleaned = data.dropna(subset=['client\_id']).reset\_index(drop=True)

data\_cleaned['receiving\_date'] = pd.to\_datetime(data\_cleaned['receiving\_date'], format='%Y-%m-%d', errors='coerce')

# Уже выполнена загрузка и очистка данных

# Фильтрация по июлю 2021 и статусу оплаты

july\_data = data\_cleaned[

(data\_cleaned['receiving\_date'].dt.month == 7) &

(data\_cleaned['receiving\_date'].dt.year == 2021) &

(data\_cleaned['status'] != "ПРОСРОЧЕНО")

]

# Вычисление общей суммы выручки

total\_revenue\_july = july\_data['sum'].sum()

print(f"Общая выручка за июль 2021: {total\_revenue\_july:.2f} руб.")

import matplotlib.pyplot as plt

# Уже выполнена загрузка и очистка данных

data\_cleaned['month\_year'] = data\_cleaned['receiving\_date'].dt.to\_period('M')

# Группировка данных по месяцу и году, суммирование выручки

monthly\_revenue = data\_cleaned.groupby('month\_year')['sum'].sum()

# Построение графика

plt.figure(figsize=(10, 6))

monthly\_revenue.plot(kind='line', marker='o')

plt.title('Изменение выручки компании за рассматриваемый период')

plt.xlabel('Период')

plt.ylabel('Выручка (в рублях)')

plt.grid(True)

plt.xticks(rotation=45)

plt.tight\_layout()

plt.show()

# Уже выполнена загрузка и очистка данных

# Фильтрация данных за сентябрь 2021

september\_data = data\_cleaned[

(data\_cleaned['receiving\_date'].dt.month == 9) &

(data\_cleaned['receiving\_date'].dt.year == 2021)

]

# Поиск менеджера с максимальной выручкой

top\_manager\_september = september\_data.groupby('sale')['sum'].sum().idxmax()

max\_sum\_september = september\_data.groupby('sale')['sum'].sum().max()

print(f"Менеджер, привлекший больше всего средств в сентябре 2021: {top\_manager\_september}, сумма: {max\_sum\_september:.2f} руб.")

# Уже выполнена загрузка и очистка данных

# Фильтрация данных за октябрь 2021

october\_data = data\_cleaned[

(data\_cleaned['receiving\_date'].dt.month == 10) &

(data\_cleaned['receiving\_date'].dt.year == 2021)

]

# Определение преобладающего типа сделок

dominant\_deal\_type\_october = october\_data['new/current'].value\_counts().idxmax()

print(f"Преобладающий тип сделок в октябре 2021: {dominant\_deal\_type\_october}")

# Уже выполнена загрузка и очистка данных

# Фильтрация данных по майским сделкам с получением оригинала в июне 2021

may\_data = data\_cleaned[

(data\_cleaned['receiving\_date'].dt.month == 6) &

(data\_cleaned['receiving\_date'].dt.year == 2021) &

(data\_cleaned['document'] == 'оригинал')

]

# Подсчет количества сделок

may\_deals\_june = may\_data.shape[0]

print(f"Количество оригиналов договоров по майским сделкам, полученных в июне 2021: {may\_deals\_june}")

Задание:

За каждую заключенную сделку менеджер получает бонус, который рассчитывается следующим образом....

Ответ:

Остаток каждого менеджера на 01.07.2021:

sale

Андреев 26175.9483

Васильев 6934.5985

Иванов 22254.2570

Кузнецова 16108.7711

Михайлов 607.0680

Петрова 27753.6542

Селиванов 7102.4570

Смирнов 30073.1380

Соколов 3730.7090

Филимонова 10830.5305

Name: bonus, dtype: float64

import pandas as pd

# Загрузка данных

file\_path = 'data.xlsx'

data = pd.read\_excel(file\_path)

# Очистка данных

data\_cleaned = data.dropna(subset=['client\_id']).reset\_index(drop=True)

data\_cleaned['receiving\_date'] = pd.to\_datetime(data\_cleaned['receiving\_date'], format='%Y-%m-%d', errors='coerce')

# Фильтрация сделок до 1 июля 2021, у которых оригиналы договоров еще не получены на тот момент

pending\_deals = data\_cleaned[

(data\_cleaned['receiving\_date'] > '2021-06-30') &

(data\_cleaned['document'] == 'оригинал')

].copy()

# Функция для расчета бонусов на будущие периоды

def calculate\_future\_bonus(row):

if row['new/current'] == 'новая' and row['status'] == 'ОПЛАЧЕНО':

return row['sum'] \* 0.07

elif row['new/current'] == 'текущая' and row['status'] != 'ПРОСРОЧЕНО':

return row['sum'] \* (0.05 if row['sum'] > 10000 else 0.03)

return 0

# Применение функции к отложенным сделкам

pending\_deals.loc[:, 'bonus'] = pending\_deals.apply(calculate\_future\_bonus, axis=1)

# Подсчет остатка бонусов по каждому менеджеру

bonus\_by\_manager\_on\_july\_1st = pending\_deals.groupby('sale')['bonus'].sum()

# Вывод остатка бонусов

print("Остаток каждого менеджера на 01.07.2021:")

print(bonus\_by\_manager\_on\_july\_1st)