Разведочный Анализ Данных

Пилипенко А.О., Банников Д.А. гр.311, ВМК МГУ

Постановка задачи

1. Необходимо провести Разведочный Анализ Данных (**Exploratory Data Analysis** - **EDA**) над имеющимися данными.

2. Написание программы, выполняющей Разведочный Анализ Данных.
    a. Математическая интерпретация задачи и его решения.
    b. Описание основных функций, используемых при программном решении.
    c. Оформление кода в виде Python Notebook.
3. Результаты анализа и выводы оформляются в виде презентации.
    a. Использование средств LaTeX и Beamer.
4. Описание кода программы, полученного в результате решения построенного примера.

Описание задачи

После оглушительного успеха в освобождении Астапора, Миэрина и Юнкая от власти работорговцев Дейенерис Бурерожденная открыла себе доступ к Летнему морю, а следовательно -- путь в Вестерос.

Для ведения войны с Семью Королевствами нужно оружие, я для оружия нужна сталь. Нет никаких сомнений в кузнечном искусстве Безупречных, однако поставщики стали не столь надежны.

Два основных поставщика стали - это Westeros Inc. и Harpy & Co. На протяжении нескольких месяцев мы закупаем сталь у обеих компаний, и каждая из них предлагает ощутимую скидку при заключении эксклюзивного договора на поставку.

Советник королевы Тирион Ланнистер знает о вашем умении принимать взвешенные рациональные решения и просит помощи в объективном решении вопроса о том, с какой из компаний следует заключить эксклюзивный договор на поставку стали.

У Тириона есть записи о производстве мечей каждым из кузнецов-безупречных, а также данные о количестве сломанных мечей в каждый из месяцев ведения боевых действий.

Вам дан CSV-файл с данными о производстве оружия и количестве единиц сломанного оружия за каждый месяц каждым из кузнецов.

Цель: ответ на вопрос: "С каким из поставщиков стали следует заключить договор?"

Математическое решение задачи

Входные данные представляют собой набор строк формата:
[ДАТА_ПРОИЗ, ДАТА_ОТЧЕТА, ПРОИЗВЕДЕНО, ДЕФЕКТНЫЕ, ПОСТАВЩИК]
где:
ДАТА_ПРОИЗ - месяц, в котором была произведена продукция
(В нашем примере мы рассмотрим статистику на 6 месяцев производства)
ДАТА_ОТЧЕТА - месяц, в котором были собраны данные о дефектности продукции, произведенной в месяце ДАТА_ПРОИЗ
(Для участия данных о произведенных в последнем месяце элементов в анализе данных о качестве продукции компаний, в нашем примере отчеты будут производиться на один месяц дольше - 7 месяцев)
ПРОИЗВЕДЕНО - количество произведенной продукции в месяце ДАТА_ПРОИЗ
ДЕФЕКТНЫЕ - количество обнаруженной дефектной продукции среди изделий, произведенных в месяце ДАТА_ПРОИЗ, после проведения отчета в месяце ДАТА_ОТЧЕТА
ПОСТАВЩИК - наименование компании-поставщика: Westeros Inc. или Harpy & C

Для проведения Разведочного Анализа Данных на наличие дефектности в производимой компаниями продукции, мы используем три экономических инструмента.

Дефектность в среднем

Каждая компания характеризуется скоростью поломки ее продукции. Так как в явном виде мы не можем определить скорость становления продукции дефектной, мы по имеющимся данным можем построить статистику поломки продукции после каждого месяца эксплуатации.
Таким образом мы получим для всей продукции данные после одного месяца эксплуатации.
Для каждой строки из нашего набора данных:
Если (ДАТА_ОТЧЕТА - ДАТА_ПРОИЗ = 1), то суммируем ДЕФЕКТНЫЕ в переменную суммирования СУМ_1.
После двух месяцев эксплуатации:
Если (ДАТА_ОТЧЕТА - ДАТА_ПРОИЗ = 2), то суммируем ДЕФЕКТНЫЕ в переменную суммирования СУМ_2.
При этом надо понимать, что для произведенной на последнем месяце производства продукции статистика об эксплуатации после двух месяцев уже не будет учитываться.
Количество таких сумм определяется максимальным диапазоном эксплуатации:
МАКСИМУМ(ДАТА_ОТЧЕТА - ДАТА_ПРОИЗ) = 6
Объединив данные по компаниям в единый график, мы получим базовое представление о том, как быстро продукция подвергается повреждению в сравниваемых нами компаниях, что станет для нас первым определителем надежности продукции каждой из интересующих нас корпораций.

Совокупный срок службы

Одним из параметров продукции является ожидаемое время целостности продукции. При закупке продукции в больших объемах часто используют совокупный срок службы для получения более точных оценок реального объема получаемого товара.

Определить совокупный срок службы для каждой компании можно так: Предположим, что продукция была произведена в первом месяце и у нее не возникало дефектов все семь месяцев эксплуатации. Тогда продукция будет иметь совокупный срок службы:

ПРОИЗВЕДЕНО * 7, где 7 = МАКСИМУМ(ДАТА_ОТЧЕТА)

Однако, часть данной продукции станет дефектной, предположим, в месяц ДАТА_ОТЧЕТА_1, потому вычитается величина:

ДЕФЕКТНЫЕ * (7 - (ДАТА_ОТЧЕТА_1 - 1))

Т.е. если часть продукции сломается в первом же месяце, тогда же, когда ее произвели, то получится:

(ПРОИЗВЕДЕНО - ДЕФЕКТНЫЕ) * 7

Часть сломалась во втором месяце:

ПРОИЗВЕДЕНО * 7 - ДЕФЕКТНЫЕ * 6

Что можно интерпретировать так:

(ПРОИЗВЕДЕНО - ДЕФЕКТНЫЕ) * 6 + ПРОИЗВЕДЕНО * 1

Т.е. часть целой продукции прожила на один месяц дольше.

Учитывая, что производство происходит не только в первом месяце, а также повторив рассуждения выше, получим итоговую формулу:

Отдельно для каждого месяца ДАТА_ПРОИЗ производства продукции

Суммируем по всем месяцам ДАТА_ОТЧЕТА величины

(ПРОИЗВЕДЕНО - ДЕФЕКТНЫЕ) * (8 - ДАТА_ОТЧЕТА)

Эта величина сродне Человеко-Часам в сфере трудоустройства описывает средний объем Продукто-Месяцев, предоставляемых компанией производителем. Совокупный срок службы является экономической интерпретацией объема закупленного товара. Таким образом мы получили второй инструмент сравнения интересующих нас фирм.

Продукция без дефектов

Последним критерием для нас станут данные об объемах закупленной продукции, еще не ставшей дефектной, на каждый месяц эксплуатации:

Суммируем для каждого месяца ДАТА_ОТЧЕТА_1 величины

(ПРОИЗВЕДЕНО - ДЕФЕКТНЫЕ)

по всем месяцам ДАТА_ОТЧЕТА с первого по ДАТА_ОТЧЕТА_1

Полученные данные позволят нам определить количество имеющейся в результате всех закупок к моменту времени ДАТА_ОТЧЕТА_1 целой продукции. При относительно малых скоростях порчи продукции данные величины показывают результаты всех закупок, произведенных с первого месяца, а в целом они определяют объем продукции, находящейся на складах покупателя, еще не подвергшейся поломке.

Таким образом, мы определили три статистических метода интерпретации данных, позволяющие получить аналитические сведения о надежности обеих компаний: Westeros Inc. и Harpy & Co. Не является редкостью создание функций, по данным экономическим статистикам определяющих надежность компании в виде одного числа, например, при помощи весовых коэффициентов. Но мы предоставляем дальнейший анализ полученных данных пользователю.

Реализация анализа описываемого нами примера и полученные из него выводы оформлены в виде презентации.

Основные функции

Разработка программы, реализующей полученные теоретические результаты и производящей графическое представление их пользователю, проходит в среде *Python Notebook*.

Основная задача программы - визуализация полученных нами результатов в виде двух типов графиков: стандартного с данными по месяцам и объемам, а так же Ящик с усами (boxplot).

Используемые библиотеки: *numpy, pandas, matplotlib.pyplot, seaborn*.

Библиотека *numpy* очень популярна, благодаря возможности выполнять различные математические операции с нестандартными для системы данными - массивами, списками и т.п. Функционал данной библиотеки интуитивно понятен, потому каждую функцию из данной библиотеки, используемую в нашем коде, мы не будет описывать сейчас.

Библиотека *pandas* используется для анализа данных, в частности, она позволяет нам работать с файлами CSV. Для этого используются функции:

read_csv(adr) - Чтение данных из CSV файла, расположенного по адресу adr.

DataFrame( ) - Конструктор объекта DataFrame (основной тип данных в *pandas*)

X.groupby(Y) - Объединение всех строк с одинаковыми элементами в столбцах Y в одну.

X.mean( ) - Округление всех данных в объекте.

X.transpose( ) - Транспонирование (разворот) данных (столбцы < - > строки).

X.columns - Доступ к списку названий столбцов

Matplotlib.pyplot - упрощенный интерфейс библиотеки *matplotlib* создания интерактивных графиков программными средствами. Используемые нами функции:

title(X) - Создание заголовка графика - X.

show( ) - Отображение поля графиков.

X.plot( ) - Построение стандартного графика x на y по данным X.

Ax.set_yticklabels(Y) - выставляет на оси y (набора данных об осях - Ax) точечные пометки по списку Y.

Ax.set_xlim(X) - Выставляет границы оси x (набора данных об осях - Ax) по промежутку X.

Библиотека *seaborn* является расширением библиотеки *matplotlib* для визуализации статистических данных. Нас интересует только одна функция:

boxplot(Data) - Построение графика Ящик с усами (boxplot) по данным Data.

Код программы

Далее мы предоставляем текст программы, написанной нами для решения описанной выше задачи. Среда разработки - *Python Notebook*.
***

```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
import seaborn as sns

pdata = pd.read_csv('production-data.csv', delimiter=',')
pdata['new'] = pdata['report.date'] - pdata['production.date']
gb = pdata.groupby(['supplier', 'new'])['defects']
gb = gb.mean()
harp_co = np.array(gb['harpy.co'][1:])
west_inc = np.array(gb['westeros.inc'][1:])

d = (np.array([np.array(gb['harpy.co'][1:]),
    np.array(gb['westeros.inc'][1:])])).transpose()
df = pd.DataFrame(data=d)
df.columns = ['harpy.co', 'westeros.inc']
ax = df.plot(kind='barh', stacked=False)

ax.set_yticklabels(['1 month', '2 month', '3 month', '4 month', '5 month', '6 month'])
ax.set_xlim([0, 20])#пределы для оси x
plt.title('Average number of defects per month')
plt.show()

#После строиться график box
d = {'harpy.co': harp_co, 'westeros.inc': west_inc}
df = pd.DataFrame(data=
plt.title('Average number of defects per month')
sns.boxplot(data=df)
plt.show()

sum_def_harp = np.zeros(6)
sum_def_west = np.zeros(6)
test_harp = np.zeros(6)
test_west = np.zeros(6)

pdata = pd.read_csv('production-data.csv', delimiter=',', header=None)
pdata = np.array(pdata[1:])
n, m = np.shape(pdata)

for i in range(n):
```

```
    t_1 = float(pdata[i, 3]) - float(pdata[i, 4]) #produce - deffects
    t_2 = t_1 * (8 - float(pdata[i, 2]))#float(pdata[i, 3]) * (8 - float(pdata[i, 2]))
    #- float(pdata[i, 4]) * (8 - float(pdata[i, 2])) #how longe was working blade
    if pdata[i, 5] == 'harpy.co':#fill if harpy.co
        j = int(pdata[i,1]) - 1
        test_harp[j] += t_1
        sum_def_harp[j] += t_2
    if pdata[i, 5] == 'westeros.inc':#fill if westeros.inc
        j = int(pdata[i,1]) - 1
        test_west[j] += t_1
        sum_def_west[j] += t_2

d = (np.array([sum_def_harp, sum_def_west])).transpose()
df = pd.DataFrame(data=d)
df.columns = ['harpy.co', 'westeros.inc']

ax = df.plot(kind='barh', stacked=False)
ax.set_yticklabels(['1 month', '2 month', '3 month', '4 month', '5 month', '6 month'])
ax.set_xlim([0, 40000])#пределы для оси x
plt.title('Aggregate service life')
plt.show()

#После строится график box
plt.title('Aggregate service life')
d = {'harpy.co': sum_def_harp, 'westeros.inc': sum_def_west}
df = pd.DataFrame(data=d)
sns.boxplot(data=df)
plt.show()

d = (np.array([test_harp, test_west])).transpose()
df = pd.DataFrame(data=d)
df.columns = ['harpy.co', 'westeros.inc']
ax=df.plot(kind='barh', stacked=False)
ax.set_yticklabels(['1 month', '2 month', '3 month', '4 month', '5 month', '6 month'])
ax.set_xlim([0,6000])#пределы для оси x
plt.title('Summary number of swords without defects')
plt.show()

#После строиться график box
plt.title('Summary number of swords without defects')
d = {'harpy.co': test_harp, 'westeros.inc': test_west}
df = pd.DataFrame(data=d)
sns.boxplot(data=df)
plt.show()
```

***