

Липецкий государственный технический университет

Факультет автоматизации и информатики

Кафедра автоматизированных систем управления

ЛАБОРАТОРНАЯ РАБОТА №0

по дисциплине «Прикладные интеллектуальные системы и экспертные

системы»

Экспертные оценки достоверности планирования продаж

Студент

Посаднев В.В.

Группа М-ИАП-22-1

Руководитель

Кургасов В.В.

Доцент

Липецк 2022 г.

Задание кафедры

Задать значения количества продаж по 10 товаром в течение 12 месяцев (помесячно). Для каждого из товаров спрогнозировать количество продаж на следующий, 13 месяц, и провести анализ достоверности планирования продаж.

Ход работы

Создадим класс для описания продукта. В конструкторе для класса будем принимать название продукта и необязательно – количество этого товара, в ином случае оно будет генерироваться самостоятельно. Также добавим в класс методы для вычисления вспомогательных значений – среднее арифметическое продаж и среднеквадратичное отклонение. Полученный набор продуктов и их количество продаж за двенадцать месяцев представлен на рисунке 1.

	Смартфон	Часы	Монитор	Наушники	Колонка	Камера	Принтер	Игровая приставка	Клавиатура	Видеокарта
0	107.797538	108.828393	111.517454	91.753091	105.888442	141.286059	160.863102	100.653641	109.348079	113.832060
1	114.300693	147.897367	112.336208	81.217090	80.014152	118.419907	151.666494	135.581925	86.514337	120.532220
2	88.969663	100.039051	130.178690	132.610398	91.674800	91.543693	151.672429	142.003629	82.207643	152.231935
3	97.521681	113.324344	101.442678	116.030000	89.726388	67.754344	110.703364	78.407494	72.594437	135.936610
4	108.583710	79.655912	111.339162	141.409745	109.501133	84.092554	101.506653	125.012368	109.298959	138.391197
5	92.790554	113.311479	154.145705	116.602160	116.295712	94.605159	173.315840	109.304726	108.571836	100.521924
6	154.222494	116.871422	121.538640	99.987600	117.003092	117.637874	118.282098	136.870125	109.417633	115.394561
7	136.331836	98.233132	110.619466	127.921305	113.309305	83.859146	150.001348	97.445316	111.635821	119.075770
8	148.849774	127.533032	88.891045	132.213614	109.118592	147.543864	159.289163	90.860107	57.024327	81.339770
9	109.690467	109.333243	112.985867	125.309692	111.721195	78.975491	161.328173	113.776524	101.372978	112.272967
10	106.181194	94.764757	123.761902	103.646519	105.534934	110.287042	166.629360	128.178790	115.978978	101.999385
11	100.614436	106.277570	89.521491	107.265738	120.231139	83.136050	145.869088	77.069406	94.968280	93.955731

Рисунок 1 – Набор продуктов с количеством продаж за двенадцать месяцев

Для генерации значения на тринадцатый месяц вычислим среднее арифметическое для каждого продукта (оценка тренда) и оценку среднеквадратичного отклонения. Спрогнозированное количество продаж на тринадцатый месяц будет равняться среднему значению плюс случайное значение в диапазоне от нуля до среднеквадратичного отклонения. Результат вычисления среднего значения продаж по каждому товару представлен на рисунке 2. Результат вычисления среднеквадратичного отклонения продаж по каждому продукту представлен на рисунке 3.

```

In 6 1 p0 = [product.avg() for product in products]
      2 [f'{p.name}: {p_avg}' for p_avg, p in zip(p0, products)]

Out 6  ✓ ['Смартфон: 113.8212',
          'Часы: 109.6725',
          'Монитор: 114.0232',
          'Наушники: 114.6639',
          'Колонка: 105.8349',
          'Камера: 101.5951',
          'Принтер: 145.9273',
          'Игровая приставка: 111.2637',
          'Клавиатура: 96.5778',
          'Видеокарта: 115.457']

```

Рисунок 2 – Результат вычисления среднего количества продаж по каждому продукту

```

In 7 1 msd_products = [product.msd() for product in products]
      2 [f'{product.name}: {msd_value}' for msd_value, product in zip(msd_products, products)]

Out 7  ✓ ['Смартфон: 21.33148027442298',
          'Часы: 17.103411782511806',
          'Монитор: 17.687587291184247',
          'Наушники: 18.27872023301185',
          'Колонка: 12.369112499450388',
          'Камера: 25.353911148873852',
          'Принтер: 23.116768205045965',
          'Игровая приставка: 22.65535053030387',
          'Клавиатура: 18.364525568905574',
          'Видеокарта: 19.95421403892538']

```

Рисунок 3 – Результат вычисления среднеквадратичного отклонения продаж по каждому продукту

После получения количества продаж за тринадцатый месяц перейдем к анализу достоверности планирования продаж. Для этого необходимо проверить три условия:

- величины X_i лежат близко к своему среднему значению. Данное условие означает, что продажи стабильны;

- экстраполируемые значения X_i с большей степенью достоверности не равны 0. Данное условие означает, что период для прогноза не слишком велик;
- среди X_i не встречаются нулевые значения. Данное условие означает, что периодичность анализа данных выбрана правильно.

Результат проверки первого условия представлен на рисунке 4, второго условия на рисунке 5, третьего условия на рисунке 6.

In 13

1

`p0_np = np.array(p0)`

`msd_np = np.array(msd_products)`

`products_df = pd.DataFrame(convert_list_products_to_dict(products))`

`condition_one = products_df - p0_np < (2 * msd_np)`

`condition_one`

Out 13

Смартфон

Часы

Монитор

Наушники

Колонка

Камера

Принтер

Игровая приставка

Клавиатура

Видеокарта

2	True	True	True	True	True	True	True	True	True	True
3	True	True	True	True	True	True	True	True	True	True
4	True	True	True	True	True	True	True	True	True	True
5	True	True	False	True	True	True	True	True	True	True
6	True	True	True	True	True	True	True	True	True	True
7	True	True	True	True	True	True	True	True	True	True
8	True	True	True	True	True	True	True	True	True	True
9	True	True	True	True	True	True	True	True	True	True
10	True	True	True	True	True	True	True	True	True	True
11	True	True	True	True	True	True	True	True	True	True
12	True	True	True	True	True	True	True	True	True	True

13 rows × 10 columns [Open in new tab](#)

In 14

1

`condition_one.all()`

Out 14

Смартфон

Часы

Монитор

Наушники

Колонка

Камера

Принтер

Игровая приставка

Клавиатура

Видеокарта

True	False	False	True	True	True	True	True	True	True
------	-------	-------	------	------	------	------	------	------	------

dtype: bool

Рисунок 4 – Проверка первого условия

```

In 15 1 condition_two = p0_np > (2 * msd_np)
      2 condition_two

Out 15 array([ True,  True,  True,  True,  True,  True,  True,  True,  True,
        True])

In 16 1 [f'{product.name}: {condition_value}' for product, condition_value in zip(products, condition_two)]

Out 16 1 ['Смартфон: True',
        'Часы: True',
        'Монитор: True',
        'Наушники: True',
        'Колонка: True',
        'Камера: True',
        'Принтер: True',
        'Игровая приставка: True',
        'Клавиатура: True',
        'Видеокарта: True']

```

Рисунок 5 – Проверка второго условия

```

In 17 1 condition_three = products_df > 0
      2 condition_three

Out 17 1

```

	Смартфон	Часы	Монитор	Наушники	Колонка	Камера	Принтер	Игровая приставка	Клавиатура	Видеокарта
2	True	True	True	True	True	True	True	True	True	True
3	True	True	True	True	True	True	True	True	True	True
4	True	True	True	True	True	True	True	True	True	True
5	True	True	True	True	True	True	True	True	True	True
6	True	True	True	True	True	True	True	True	True	True
7	True	True	True	True	True	True	True	True	True	True
8	True	True	True	True	True	True	True	True	True	True
9	True	True	True	True	True	True	True	True	True	True
10	True	True	True	True	True	True	True	True	True	True
11	True	True	True	True	True	True	True	True	True	True
12	True	True	True	True	True	True	True	True	True	True

13 rows × 10 columns [Open in new tab](#)

```

In 18 1 condition_three.all()

Out 18 1
Смартфон      True
Часы          True
Монитор       True
Наушники      True
Колонка       True
Камера        True
Принтер       True
Игровая приставка True
Клавиатура    True
Видеокарта    True
dtype: bool

```

Рисунок 6 – Проверка третьего условия

После проверки вышеперечисленных условий решение о достоверности планирования по совокупности этих значений может быть сведено в таблицу, представленную на рисунке 7. Предложенная цветовая интерпретация представляется удобной для практики, когда нужно наглядно

проинформировать менеджера о том, для каких номенклатурных позиций автоматизированное планирование продаж достоверно («зеленый» уровень), а для каких (возможно) необходима «интуитивная», «ручная» корректировка полученных данных.

	Товар	Уровень достоверности	Условие 1	Условие 2	Условие 3
0	Смартфон	Зеленый	True	True	True
1	Часы	Желтый	False	True	True
2	Монитор	Желтый	False	True	True
3	Наушники	Зеленый	True	True	True
4	Колонка	Зеленый	True	True	True
5	Камера	Зеленый	True	True	True
6	Принтер	Зеленый	True	True	True
7	Игровая приставка	Зеленый	True	True	True
8	Клавиатура	Зеленый	True	True	True
9	Видеокарта	Зеленый	True	True	True

Рисунок 7 – Цветовая интерпретация полученных результатов

Как видно из рисунка 7, условие стабильности продаж не выполняются для товаров: часы и мониторы. Поэтому их уровень достоверности – желтый.

Полный скрипт для выполнения данных расчетов представлен в приложении А.

Заключение

В ходе выполнения данной лабораторной работы мною были составлены и проверены экспертные оценки достоверности планирования продаж для десяти товаров на срок двенадцать месяцев. В результате интерпретации полученных результатов условие стабильности продаж не выполняются для товаров: часы и мониторы. Поэтому их уровень достоверности – желтый. Для остальных товаров уровень достоверности – зеленый.

Приложение А

Код для данной лабораторной работы

```
#!/usr/bin/env python
# coding: utf-8

# # Задание
# ## Задать значения количества продаж по 10 товарам в течении 12 месяцев
# (помесячно). Для каждого из товаров спрогнозировать количество продаж на
# следующий, 13 месяц и провести анализ достоверности планирования продаж

# In[1]:

import random
# Для удобной работы с массивами
import numpy as np
# Для представления данных в виде датафреймов
import pandas as pd
# Для рисования графиков
import matplotlib.pyplot as plt
# Для отключения предупреждений
import warnings
warnings.simplefilter(action='ignore', category=FutureWarning)

class Product:
    def __init__(self, name, count=None):
        """
        Конструктор класса
        """
        self.name = name
        if count is not None:
            self.count = count
        else:
            min_rand = random.randint(100, 150)
            delt_rand = 20
            self.count = np.random.normal(min_rand, delt_rand, 12)

    def __str__(self):
        """
        Метод для конвертации в строку
        """
        return f'{self.name}: {[i for i in self.count]}'

    def __repr__(self):
        """
        Метод для представления в виде строки из дебаг-консоли
        """
        return f'{self.name}: {[i for i in self.count]}'

    def to_dict(self):
        """
        Возвращает класс в представлении словаря
        """
        return {
            self.name: self.count
        }

    def sum(self):
        """
        Возвращает общую стоимость за все месяца
```

```

        """
        return sum(self.count)

def avg(self):
    """
    Возвращает среднюю цену
    """
    return round(sum(self.count) / len(self.count), 4)

def msd(self):
    """
    Возвращает СКО для продукции
    """
    avg_value = self.avg()
    upper_value = sum([(v - avg_value) ** 2 for v in self.count])
    msd_square = upper_value / (len(self.count) - 1)
    return msd_square ** 0.5

# In[2]:

products = [
    Product(name='Смартфон'),
    Product(name='Часы'),
    Product(name='Монитор'),
    Product(name='Наушники'),
    Product(name='Колонка'),
    Product(name='Камера'),
    Product(name='Принтер'),
    Product(name='Игровая приставка'),
    Product(name='Клавиатура'),
    Product(name='Видеокарта')
]

products

# In[3]:

def convert_list_products_to_dict(p_list: list):
    """
    Конвертирует лист продуктов в словарь для визуализации в датафрейме
    """
    result = {}
    for p in p_list:
        result[p.name] = p.count
    return result

# In[4]:

df = pd.DataFrame(convert_list_products_to_dict(products))
df

# In[5]:

for product in products:
    plt.plot([i for i in range(12)], product.count, label=product.name)

```

```

plt.legend(bbox_to_anchor=(1, 1))
plt.show()

# # Вычисляем оценку тренда (среднее арифметическое величины)

# In[6]:

p0 = [product.avg() for product in products]
[f'{p.name}: {p_avg}' for p_avg, p in zip(p0, products)]

# # Вычисляем оценку среднеквадратичного отклонения

# In[7]:

msd_products = [product.msd() for product in products]
[f'{product.name}: {msd_value}' for msd_value, product in zip(msd_products,
products)]

# # Прогнозируем количество продаж на следующий, 13 месяц

# In[8]:

predict_values = p0 + np.random.normal(0, msd_products, len(msd_products))
predict_values

# In[9]:

for product, predict_value in zip(products, predict_values):
    product.count = np.append(product.count, predict_value)
products

# In[10]:

df = pd.DataFrame(convert_list_products_to_dict(products))
df

# In[11]:

for product in products:
    plt.plot([i for i in range(13)], product.count, label=product.name)
plt.legend(bbox_to_anchor=(1, 1))
plt.show()

# # Анализ достоверности планирования продаж

# ### Расчитаем достоверность прогноза

# In[12]:

```

```

reliability = [msd_value / p0_value for msd_value, p0_value in
zip(msd_products, p0)]
[f'{product.name}: {reliability_value}' for product, reliability_value in
zip(products, reliability)]

# ### Первое условие
# ##### Величины  $\{X_i\}$   $\$$  лежат близко к своему среднему значению. Это
условие означает, что продажи стабильные
# #####  $\hat{p}_0 - 2\hat{\sigma} < X_i < \hat{p}_0 + 2\hat{\sigma}$   $\$$  для всех  $i$   $\$$ 

# In[13]:

p0_np = np.array(p0)
msd_np = np.array(msd_products)
products_df = pd.DataFrame(convert_list_products_to_dict(products))

condition_one = products_df - p0_np < (2 * msd_np)
condition_one

# In[14]:

condition_one.all()

# ### Второе условие
# ##### Экстраполируемые значения  $X_i$   $\$$  с большей степенью достоверности не
равны 0. Это условие означает, что период для прогноза не слишком велик
# #####  $\hat{p}_0 > 2\hat{\sigma}$   $\$$ 

# In[15]:

condition_two = p0_np > (2 * msd_np)
condition_two

# In[16]:

[f'{product.name}: {condition_value}' for product, condition_value in
zip(products, condition_two)]

# ### Третье условие
# ##### Среди  $X_i$   $\$$  не встречаются нулевые значения. Это условие означает,
что периодичность анализа данных выбрана правильно
# #####  $X_i > 0$   $\$$  для всех  $i$   $\$$ 

# In[17]:

condition_three = products_df > 0
condition_three

# In[18]:

condition_three.all()

```

```

# ### Составим таблицу для визуализации достоверности планирования

# In[19]:

def get_color_by_condition(c1, c2, c3):
    """
    Получение уровня достоверности по условиям
    """
    if c1 and c2 and c3:
        return "Зеленый"
    elif (not c1 and c2 and c3) or (c1 and not c2 and c3):
        return "Желтый"
    elif not c1 and not c2 and c3:
        return "Оранжевый"
    else:
        return "Красный"

# In[20]:

table_header = ["Товар", "Уровень достоверности", "Условие 1", "Условие 2",
"Условие 3"]
table = pd.DataFrame(columns=table_header)

condition_one_result = condition_one.all()
condition_two_result = condition_two
condition_three_result = condition_three.all()

for i in range(len(products)):
    table = table.append({
        "Товар": products[i].name,
        "Уровень достоверности":
get_color_by_condition(condition_one_result[i], condition_two_result[i],
condition_three_result[i]),
        "Условие 1": condition_one_result[i],
        "Условие 2": condition_two_result[i],
        "Условие 3": condition_three_result[i]
    }, ignore_index=True)

table

```