**Липецкий государственный технический университет**

Факультет автоматизации и информатики

Кафедра автоматизированных систем управления

ЛАБОРАТОРНАЯ РАБОТА №1

по дисциплине

«Прикладные интеллектуальные системы и экспертные системы»

Студент Бубырь Д. А.

Группа М-ИАП-23-1

Руководитель Кургасов В. В.

доцент, канд. пед. наук

Липецк 2023 г.

Задание кафедры

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

Ход работы

Импортируем необходимые модули и библиотеки

import numpy as np

import pandas as pd

import seaborn

import random

Напишем на языке Python функцию для генерации последовательности заданного количество целых чисел из заданного диапазона.

def random\_sales(min, max):  
 li = []  
 for i in range(0, 12):  
 element = random.randint(min, max)  
 li.append(element)  
 return li

Сгенерируем и выведем количество продаж за год по месяцам для 10-ти товаров.

sales = pd.DataFrame({  
 'Computers': random\_sales(5, 10),  
 'Laptops': random\_sales(10, 25),  
 'Smartphones': random\_sales(15, 45),  
 'Smart Watch': random\_sales(10, 15),  
 'TVs': random\_sales(25, 100),  
 'Dishwashers': random\_sales(10, 50),  
 'Microwave Ovens': random\_sales(45, 80),  
 'Refrigerators': random\_sales(15, 80),  
 'Washing machines': random\_sales(10, 35),  
 'Coffee Makers': random\_sales(2, 15)  
})

print(sales)

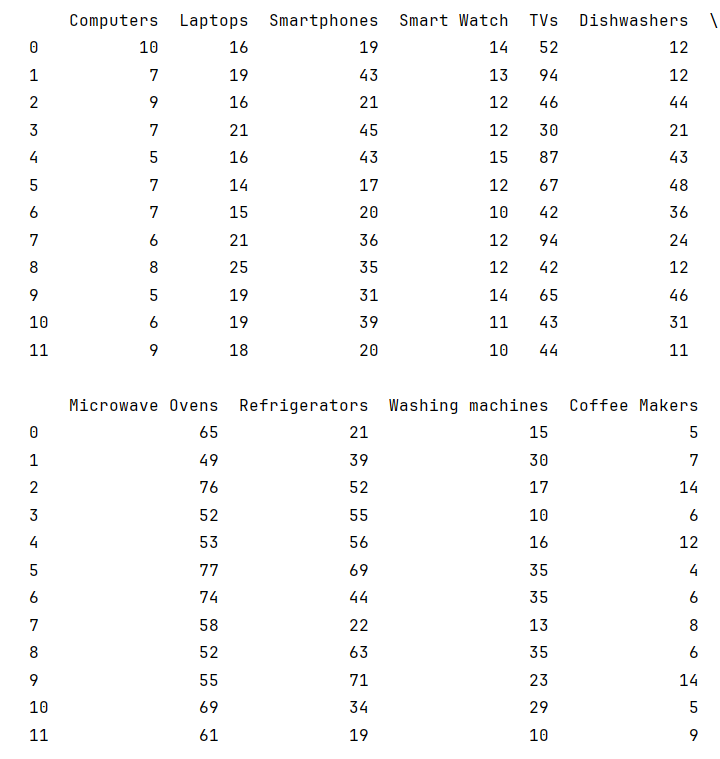


Рисунок 1 — Количество продаж продуктов по месяцам

Построим график, отображающий нашу выборку

lp = seaborn.lineplot(sales)

seaborn.move\_legend(lp, 'upper left', bbox\_to\_anchor=(1, 1))

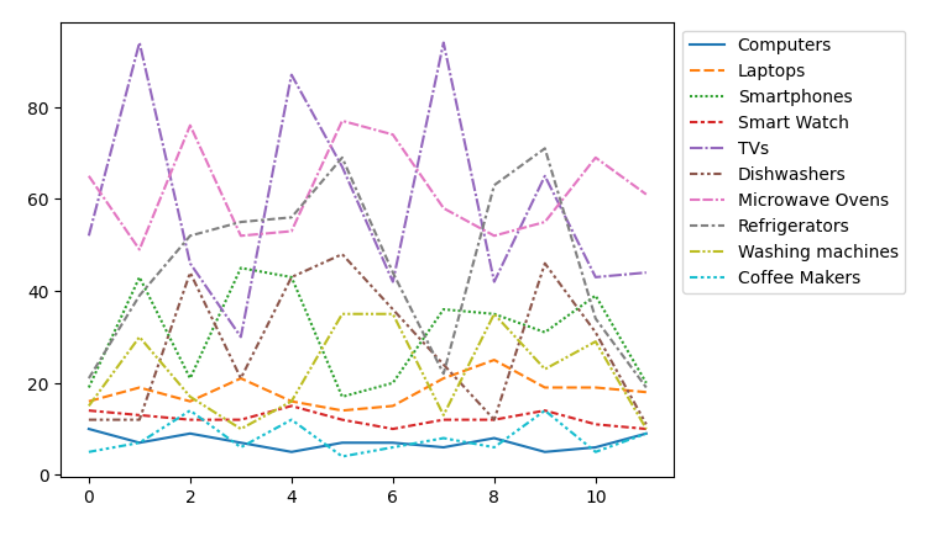


Рисунок 2 — Выборка данных на графике

Рассчитаем

p0 = sales.sum() / sales.shape[0]

print(p0)

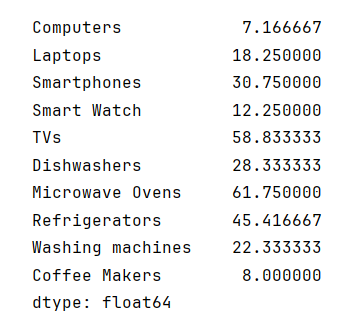


Рисунок 3 — Расчёт

Рассчитаем планируемые показатели продаж на следующий месяц.

predict = pd.DataFrame([p0 + np.random.normal(0, std, len(p0))])

print(predict)

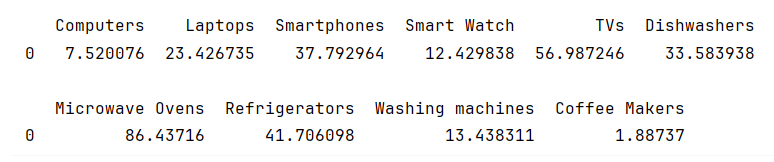


Рисунок 4 — Планируемые показатели продаж

Проведем расчёт отношения std / p0.

square\_std = ((sales - p0) \*\* 2).sum() / (sales.shape[0] - 1)

std = square\_std \*\* 0.5

reliability = std / p0

print(reliability)

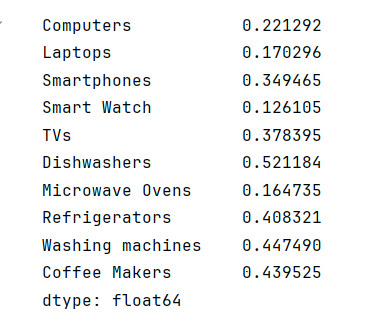
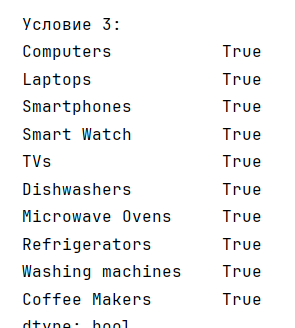
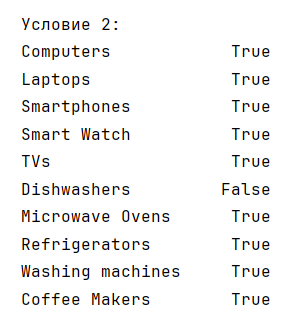
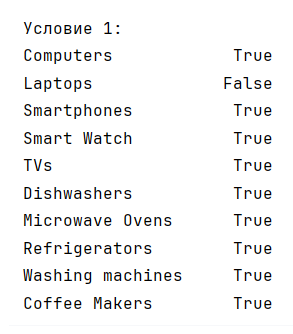


Рисунок 5 — Расчёт отношения (оценки)

Введем условия, требующиеся для «раскраски» показателей.

condition1 = ((sales - p0) < 2 \* std).all()  
condition2 = p0 > 2 \* std  
condition3 = (sales > 0).all()  
  
print('Условие 1: ')  
print(condition1)  
print('\n')  
print('Условие 2: ')  
print(condition2)  
print('\n')  
print('Условие 3: ')  
print(condition3)



|  |
| --- |
|  |

Рисунок 6 — Условия

Добавим цвет товарам в зависимости от условий, перечисленных выше.

product\_color = pd.Series(dtype='string')  
for product in sales.columns:  
 if not condition3[product]:  
 product\_color[product] = 'Red'  
 elif not condition1[product] and not condition2[product]:  
 product\_color[product] = 'Orange'  
 elif not condition1[product] or not condition2[product]:  
 product\_color[product] = 'Yellow'  
 else:  
 product\_color[product] = 'Green'  
  
print(product\_color)



Рисунок 7 — Цветовые метки товаров

Вывод

В ходе лабораторной работы мы спрогнозировали объем продаж товаров на 13-ый месяц и провели анализ достоверности планирования продаж.

Вывод

В ходе лабораторной работы мы спрогнозировали объем продаж товаров на 13-ый месяц и провели анализ достоверности планирования продаж.