**Липецкий государственный технический университет**

Факультет автоматизации и информатики

Кафедра автоматизированных систем управления

ЛАБОРАТОРНАЯ РАБОТА №1

по дисциплине

«Прикладные интеллектуальные системы и экспертные системы»

Студент Первушин О. С.

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

# Импорт модулей

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.

# Функция для формирования массива продаж для продукта

def randomSales(count, min, max):

  li = []

  for i in range(0, count):

        tmp = random.randint(min, max)

        li.append(tmp)

  return li

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

# задаем количество продаж на 1 год  по продуктам

sales = pd.DataFrame({

    'Laptops' : randomSales(12, 10, 35),

    'Smartphones' : randomSales(12, 10, 50),

    'Smart Watch' : randomSales(12, 10, 40),

    'Computers' : randomSales(12, 10, 25),

    'Microwave Ovens' : randomSales(12, 10, 30),

    'Refrigerators' : randomSales(12, 10, 20),

    'Washing machines': randomSales(12, 10, 15),

    'Dishwashers' : randomSales(12, 5, 10),

    'TVs' : randomSales(12, 10, 45),

    'Coffee Makers' : randomSales(12, 7, 15)

})

print(sales)

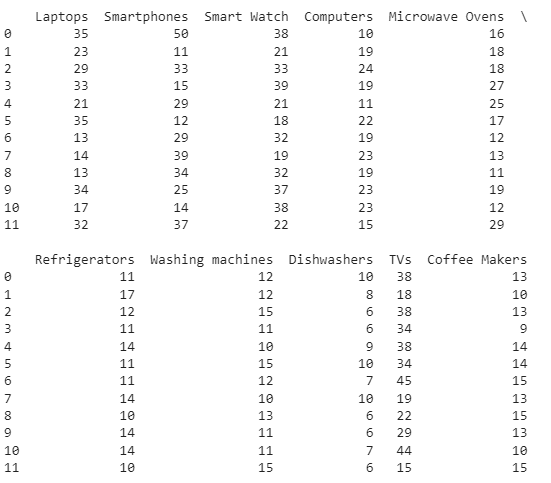


Рисунок 1 — Количество продаж продуктов по месяцам

Построим график, отображающий нашу выборку

# График

lp = seaborn.lineplot(sales)

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

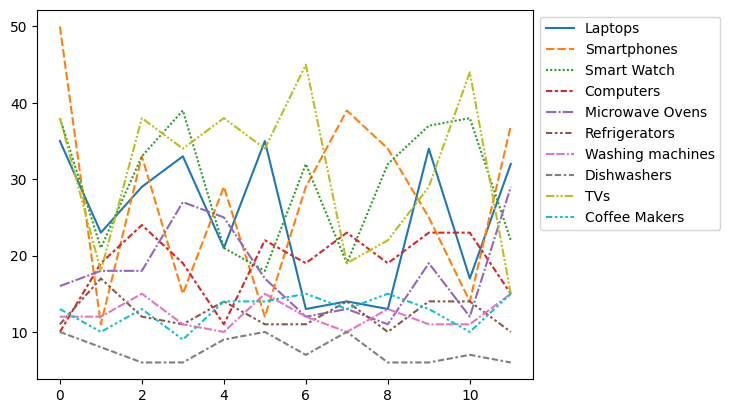


Рисунок 2 — Выборка данных на графике

Рассчитаем

# p0

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

print(p0)

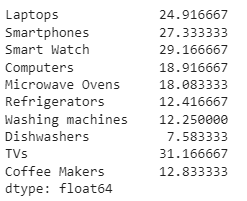


Рисунок 3 — Расчёт

Рассчитаем планируемые показатели продаж на следующий месяц.

# Расчёт планируемого показателя

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

print(predict)

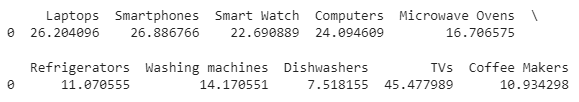


Рисунок 4 — Планируемые показатели продаж

Проведем расчёт отношения std / p0.

# Оценка

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

std = square\_std \*\* 0.5

reliability = std / p0

print(reliability)

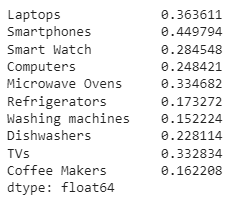


Рисунок 5 — Расчёт отношения (оценки)

Введем условия, требующиеся для «раскраски» показателей.

# Условия

cond\_1 = ((sales - p0) < 2 \* std).all()

cond\_2 = p0 > 2\* std

cond\_3 = (sales > 0).all()

print('Условие 1: ')

print(cond\_1)

print('\n')

print('Условие 2: ')

print(cond\_2)

print('\n')

print('Условие 3: ')

print(cond\_3)

|  |  |  |
| --- | --- | --- |
|  |  |  |

Рисунок 6 — Условия

Добавим цвет товарам в зависимости от условий, перечисленных выше.

# Раскраска товаров

product\_color = pd.Series(dtype='string')

for product in sales.columns:

  if (not cond\_3[product]):

    product\_color[product] = 'Red'

  elif (not cond\_1[product] and not cond\_2[product]):

    product\_color[product] = 'Orange'

  elif (not cond\_1[product] or not cond\_2[product]):

    product\_color[product] = 'Yellow'

  else:

    product\_color[product] = 'Green'

print(product\_color)

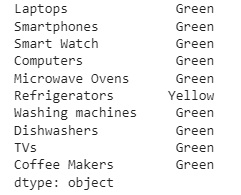


Рисунок 7 — Цветовые метки товаров

Вывод

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