

Липецкий государственный технический университет

Факультет автоматизации и информатики

Кафедра автоматизированных систем управления

ЛАБОРАТОРНАЯ РАБОТА №1

**по дисциплине «Прикладные интеллектуальные системы и экспертные
системы»**

Экспертные системы. Оценка планирования продаж.

Студент

Глебов Д.А.

Группа М-ИАП-22

Руководитель

Кургасов В.В.

Липецк 2022 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 месяц и провести анализ достоверности планирования продаж.

Ход работы

Используемые библиотеки:

- numpy предназначена для поддержки многомерных массивов (включая матрицы), поддержки высокоуровневых математических функций;
- pandas предназначена для обработки и анализа данных.

Сгенерируем информацию о продажах 10-ти видов продуктов за 12 месяцев

```
[17]: product = {
      'Железо': numpy.random.normal(20, 5, 12),
      'Золото': numpy.random.normal(30, 6, 12),
      'Олово': numpy.random.normal(70, 5, 12),
      'Медь': numpy.random.normal(20, 5, 12),
      'Титан': numpy.random.normal(90, 6, 12),
      'Магний': numpy.random.normal(60, 20, 12),
      'Натрий': numpy.random.normal(40, 5, 12),
      'Калий': numpy.random.normal(80, 10, 12),
      'Алюминий': numpy.random.normal(90, 15, 12),
      'Серебро': numpy.random.normal(30, 10, 12),
    }
```

Рисунок 1 – Генерация данных

Сформируем единую таблицу из сгенерированных данных.

```
[18]: dataFrame = pandas.DataFrame(product)
      dataFrame
```

	Железо	Золото	Олово	Медь	Титан	Магний	Натрий	Калий	Алюминий	Серебро
0	20.861599	34.057796	67.225067	25.712593	86.527630	51.852943	41.205927	83.240289	64.829479	41.331414
1	32.821820	43.074205	74.660225	13.741756	82.166603	50.549070	39.010028	87.282785	96.165074	24.830946
2	26.635332	33.491668	78.843697	29.177742	94.692438	65.400062	43.095610	66.192259	88.364196	43.253521
3	17.308731	22.618430	73.065443	25.656845	89.127013	67.711176	40.185644	71.459329	91.378358	32.498206
4	24.623129	38.519894	74.077854	22.644149	84.551686	27.303267	35.630223	87.552313	106.858758	12.181447
5	26.664846	38.954189	73.138186	27.002016	87.517013	43.102512	32.578320	73.564394	95.317367	30.011359
6	15.837382	20.457264	71.898798	12.646459	90.806385	49.700890	37.938995	75.964011	88.889263	17.897234
7	21.093492	31.503713	71.235024	16.889756	89.273249	66.600635	29.338824	70.553618	87.486251	27.679251
8	25.698198	25.819908	70.825767	20.142329	86.617617	113.775397	39.047212	78.333963	88.977117	8.450054
9	15.319809	34.761281	66.289477	19.037028	91.810721	61.591872	37.775668	90.351799	83.224185	30.096186
10	27.682190	31.184897	67.912980	31.274121	109.153680	26.718043	33.805644	78.653468	110.123279	28.095570
11	26.300819	25.027676	62.826310	16.648715	94.319351	48.959396	40.585304	88.647617	71.666607	39.232939

Рисунок 2 – Сформированные данные

▼ Рассчитаем среднее арифметическое величины X_i

```
[19]: p0 = dataFrame.sum() / 12
      p0
```

```
[19]: Железо      23.403946
      Золото     31.622577
      Олово     70.999902
      Медь      21.714459
      Титан     90.546949
      Магний    56.105438
      Натрий    37.516450
      Калий     79.316320
      Алюминий  89.439995
      Серебро   27.963177
      dtype: float64
```

Рисунок 3 – Расчёт среднего арифметического (\hat{p}_0)

Рассчитаем среднее квадратичное отклонение величины X_i и сразу извлечем квадратный корень

```
[20]: sigma = (((dataFrame - p0) ** 2).sum() / (12 - 1)) ** (1 / 2)
sigma
```

```
[20]: Железо      5.349396
Золото      6.972992
Олово      4.336980
Медь       6.118602
Титан      6.940513
Магний     22.744547
Натрий     3.985064
Калий      8.045517
Алюминий   12.722343
Серебро    10.912380
dtype: float64
```

Рисунок 4 – Расчет среднеквадратичного отклонения

Значение планируемого показателя на 13-ый месяц

```
[21]: x_i = p0 + numpy.random.normal(0, sigma, len(p0))
columns = x_i.index
pandas.concat([dataFrame, pandas.DataFrame([x_i], columns=columns)]).reset_index(drop=True)
```

```
[21]:
```

	Железо	Золото	Олово	Медь	Титан	Магний	Натрий	Калий	Алюминий	Серебро
0	20.861599	34.057796	67.225067	25.712593	86.527630	51.852943	41.205927	83.240289	64.829479	41.331414
1	32.821820	43.074205	74.660225	13.741756	82.166603	50.549070	39.010028	87.282785	96.165074	24.830946
2	26.635332	33.491668	78.843697	29.177742	94.692438	65.400062	43.095610	66.192259	88.364196	43.253521
3	17.308731	22.618430	73.065443	25.656845	89.127013	67.711176	40.185644	71.459329	91.378358	32.498206
4	24.623129	38.519894	74.077854	22.644149	84.551686	27.303267	35.630223	87.552313	106.858758	12.181447
5	26.664846	38.954189	73.138186	27.002016	87.517013	43.102512	32.578320	73.564394	95.317367	30.011359
6	15.837382	20.457264	71.898798	12.646459	90.806385	49.700890	37.938995	75.964011	88.889263	17.897234
7	21.093492	31.503713	71.235024	16.889756	89.273249	66.600635	29.338824	70.553618	87.486251	27.679251
8	25.698198	25.819908	70.825767	20.142329	86.617617	113.775397	39.047212	78.333963	88.977117	8.450054
9	15.319809	34.761281	66.289477	19.037028	91.810721	61.591872	37.775668	90.351799	83.224185	30.096186
10	27.682190	31.184897	67.912980	31.274121	109.153680	26.718043	33.805644	78.653468	110.123279	28.095570
11	26.300819	25.027676	62.826310	16.648715	94.319351	48.959396	40.585304	88.647617	71.666607	39.232939
12	24.951469	29.841810	76.803627	24.746439	83.850417	14.563066	41.227801	67.406280	77.289800	35.364414

Рисунок 5 – Расчёт планируемого показателя на 13-ый месяц

Проверим три условия, характеризующие достоверность прогноза

```
[22]: authenticity = sigma / p0
authenticity
```

```
[22]: Железо      0.228568
Золото      0.220507
Олово      0.061084
Медь       0.281775
Титан      0.076651
Магний     0.405389
Натрий     0.106222
Калий      0.101436
Алюминий   0.142244
Серебро    0.390241
dtype: float64
```

Рисунок 6 – Рассчитаем соотношение $\frac{\hat{\sigma}_p}{\hat{p}_0}$

Условие 1

```
[23]: condition1 = ((dataFrame - p0) < 2 * sigma).all()  
condition1
```

```
[23]: Железо      True  
Золото      True  
Олово       True  
Медь        True  
Титан       False  
Магний      False  
Натрий      True  
Калий       True  
Алюминий    True  
Серебро     True  
dtype: bool
```

Рисунок 7 – Проверка первого условия

▼ Условие 2

```
[24]: condition2 = p0 > 2 * sigma  
condition2
```

```
[24]: Железо      True  
Золото      True  
Олово       True  
Медь        True  
Титан       True  
Магний      True  
Натрий      True  
Калий       True  
Алюминий    True  
Серебро     True  
dtype: bool
```

Рисунок 8 – Проверка второго условия

Условие 3

```
[25]: condition3 = (dataFrame > 0).all()  
condition3
```

```
[25]: Железо      True  
Золото      True  
Олово       True  
Медь        True  
Титан       True  
Магний      True  
Натрий      True  
Калий       True  
Алюминий    True  
Серебро     True  
dtype: bool
```

Рисунок 9 – Проверка третьего условия

26]:	keyword	color
0	Железо	green
1	Золото	green
2	Олово	green
3	Медь	green
4	Титан	yellow
5	Магний	yellow
6	Натрий	green
7	Калий	green
8	Алюминий	green
9	Серебро	green

Рисунок 10 – Сведем итоги в таблицу

Таким образом, автоматизированное планирование продаж достоверно у всех, кроме позиции «ментай», для нее нужна корректировка полученных данных.

Код программы

```
import numpy as numpy  
import pandas as pandas  
import seaborn as sns
```

```

## Сгенерируем информацию о продажах 10-ти товаров за 12 месяцев
product = {
    'Железо': numpy.random.normal(20, 5, 12),
    'Золото': numpy.random.normal(30, 6, 12),
    'Олово': numpy.random.normal(70, 5, 12),
    'Медь': numpy.random.normal(20, 5, 12),
    'Титан': numpy.random.normal(90, 6, 12),
    'Магний': numpy.random.normal(60, 20, 12),
    'Натрий': numpy.random.normal(40, 5, 12),
    'Калий': numpy.random.normal(80, 10, 12),
    'Алюминий': numpy.random.normal(90, 15, 12),
    'Серебро': numpy.random.normal(30, 10, 12),
}

## Сформируем единую таблицу из сгенерированных данных.
dataFrame = pandas.DataFrame(product)

dataFrame

## Рассчитаем среднее арифметическое величины  $X_i$ 
p0 = dataFrame.sum() / 12
p0

## Рассчитаем среднее квадратичное отклонение величины  $X_i$  и сразу
извлечем квадратный корень
sigma = (((dataFrame - p0) ** 2).sum() / (12 - 1)) ** (1 / 2)
sigma

## Значение планируемого показателя на 13-ый месяц
x_i = p0 + numpy.random.normal(0, sigma, len(p0))
columns = x_i.index
pandas.concat([dataFrame, pandas.DataFrame([x_i],
columns=columns)]).reset_index(drop=True)

## Проверим три условия, характеризующие достоверность прогноза
authenticity = sigma / p0

```

```

authenticity
## Условие 1
condition1 = ((dataFrame - p0) < 2 * sigma).all()
condition1
## Условие 2
condition2 = p0 > 2 * sigma
condition2
## Условие 3
condition3 = (dataFrame > 0).all()
condition3
## Итоги
df_1 = pandas.DataFrame({'keyword':dataFrame.columns,
                          'color': 'green'})

i = 0

for name in dataFrame.columns:
    if (not condition3[name]):
        df_1.color[i] = 'red'
    elif (not condition1[name] and not condition2[name]):
        df_1.color[i] = 'orange'
    elif (not condition1[name] or not condition2[name]):
        df_1.color[i] = 'yellow'
    i += 1

new_dict = pandas.Series(df_1.color.values,index=df_1.keyword).to_dict()
df_1.style.applymap(lambda v: f"background-color: {new_dict.get(v,
'None')}")

```


Вывод

В ходе выполнения данной лабораторной работы мы получили базовые навыки работы с языком python и набором функций для анализа и обработки данных. Спрогнозировали количество продаж по десяти товарам на тринадцатый месяц и провели анализ достоверности планирования продаж.