

Липецкий государственный технический университет

Факультет автоматизации и информатики

Кафедра автоматизированных систем управления

ЛАБОРАТОРНАЯ РАБОТА №0

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

Экспертные системы. Оценка планирования продаж

Студент

Сухоруких А.О.

Группа М-ИАП-22

Руководитель

Кургасов В.В.

доцент

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

1 Теоретическая часть

Существует множество подходов к решению задачи планирования продаж выбор в пользу того или иного из них определяется стратегией работы предприятия. В алгоритмическом плане значительная часть методик основана на ретроспективном анализе результатов продаж за некоторый период и экстраполяции этих результатов. При этом основной проблемой становится выбор таких параметров задачи как модель эволюции данных во времени, периодичность данных для анализа и интервал экстраполяции.

В задачах экономического анализа минимальным периодом оценивания какого-либо показателя являются, как правило, сутки, соотнесённые с конкретной датой. Часто используются и другие периоды: «недельный», «месячный», «квартальный» и т. д. Модель изменения значения показателя X во времени может быть выражена формулой:

$$X_k = G(k) + \eta(k), \quad (1)$$

Где $G(k)$ - функция, выражающая детерминированный закон эволюции величины X (тренд), $\eta(k)$ - случайная величина, характеризующая отклонение фактического значения показателя от его тренда (здесь и далее будем считать, что $\eta(k)$ - некоррелированная случайная величина с нулевым математическим ожиданием).

Функция $G(k)$ может быть использована для экстраполяции значений показателя X . Анализ свойств $\eta(k)$ может быть положен в основу оценки достоверности планирования.

$$G(k) = \sum_{j=0}^m p_j \cdot k^j. \quad (2)$$

Решение уравнения (3) методом наименьших квадратов относительно векторар имеет вид:

$$\hat{p} = (K^T R^{-1} K)^{-1} K^T R^{-1} X, \quad (3)$$

Наряду с оценкой вектора p может быть получена и оценка в виде формулы

$$\hat{\sigma}_\eta^2 = \frac{1}{J - m} (X - K\hat{p})^T (X - K\hat{p}). \quad (4)$$

Для одномерного случая, то есть когда $m=0$ совокупность формул (2-8) вырождается следующим образом:

$$G(k) = p_0 \text{ (то есть тренд есть константа);}$$

$X_i = p_0 + \eta_i$ (то есть значение планируемого показателя равно константе плюс погрешность);

$\hat{p}_0 = \frac{\sum_{i=0}^J X_i}{J}$ (то есть оценка тренда есть среднее арифметическое величины X_i);

$\hat{\sigma}_p^2 = \frac{\sum_{i=0}^J (X_i - \hat{p}_0)^2}{J-1}$ (это оценка среднеквадратичного отклонения величины X_i , то есть, фактически, погрешности нахождения p_0)

Суть проблемы прогнозирования в том, что найденная величина \hat{p}_0 имеет смысл только вместе с информацией о погрешности ее определения $\hat{\sigma}_p^2$. В случае малого их соотношения можно считать, что прогноз достоверен, в случае, если величины соизмеримы – прогноз недостоверен.

2 Используемые пакеты и библиотеки

NumPy — это расширение языка Python, добавляющее поддержку больших многомерных массивов и матриц, вместе с большой библиотекой высокоуровневых математических функций для операций с этими массивами.

Команда для установки:

```
pip install numpy
```

Pandas — это библиотека Python для обработки и анализа структурированных данных, её название происходит от «panel data» («панельные данные»). Панельными данными называют информацию, полученную в результате исследований и структурированную в виде таблиц. Для работы с такими массивами данных и создан Pandas.

Команда для установки:

```
pip install pandas
```

Seaborn — это библиотека для создания статистических графиков на Python. Она основывается на matplotlib и тесно взаимодействует со структурами данных pandas. Архитектура Seaborn позволяет вам быстро изучить и понять свои данные. Seaborn захватывает целые фреймы данных или массивы, в которых содержатся все ваши данные, и выполняет все внутренние функции, нужные для семантического маппинга и статистической агрегации для преобразования данных в информативные графики.

Команда для установки:

```
pip install seaborn
```

3 Программная реализация

3.1 Создание данных

Процесс создания данных с помощью языка программирования python показан на рисунке 1

Создание данных

```
In [114]: input_data = {
    'Хлеб черный': np.random.normal(200, 5, 12),
    'Хлеб белый': np.random.normal(200, 10, 12),
    'Чай черный': np.random.normal(185, 8, 12),
    'Чай зеленый': np.random.normal(135, 5, 12),
    'Шоколад': np.random.normal(100, 5, 12),
    'Зефир': np.random.normal(120, 8, 12),
    'Кофе': np.random.normal(150, 5, 12),
    'Конфеты': np.random.normal(175, 10, 12),
    'Корм для котов': np.random.normal(180, 15, 12),
    'Вафли': np.random.normal(115, 8, 12),
}

In [115]: dataframe = pd.DataFrame(input_data)
dataframe

Out[115]:
```

	Хлеб черный	Хлеб белый	Чай черный	Чай зеленый	Шоколад	Зефир	Кофе	Конфеты	Корм для котов	Вафли
0	205.429191	193.367650	160.767363	141.956082	106.472670	140.776385	148.835215	183.569873	173.456821	125.317394
1	194.506797	190.133561	186.118433	136.087868	103.231337	115.697107	148.484748	203.177669	167.057990	120.524477
2	196.758885	212.435281	190.442735	136.943244	84.930401	127.164914	152.842627	162.497705	158.861979	116.154575
3	204.505951	211.486902	178.842638	130.616344	100.846369	129.998950	142.693845	172.966223	176.700121	110.916353
4	195.615854	190.189593	193.737063	135.300294	103.366260	126.178926	159.430614	173.269291	185.154325	120.977461
5	190.496279	190.539886	189.014211	133.471239	103.876211	127.804144	162.764282	178.223779	207.955666	118.177614
6	197.076813	191.528343	189.300888	140.087955	102.839398	125.050746	145.076314	175.212790	168.732272	126.337672
7	202.628125	202.266250	191.012350	135.500040	96.112581	128.836065	142.134074	192.723604	181.268679	114.111191
8	202.510113	201.513609	185.267620	132.459546	108.966152	134.224876	155.635829	177.423127	162.587954	108.588938
9	205.352583	193.696520	185.781140	138.119932	104.556395	126.257246	144.561867	172.725118	194.139502	123.829776
10	197.019446	202.416554	204.144172	139.351912	101.428658	130.314046	157.496152	182.510822	173.001661	117.046084
11	199.162047	187.993748	175.734652	140.314729	92.213365	110.870745	153.126072	177.414468	188.599990	139.317439

Рисунок 1 - Создание данных

Составим графическое представление наших данных, графическое представление данных показано на рисунке 2.

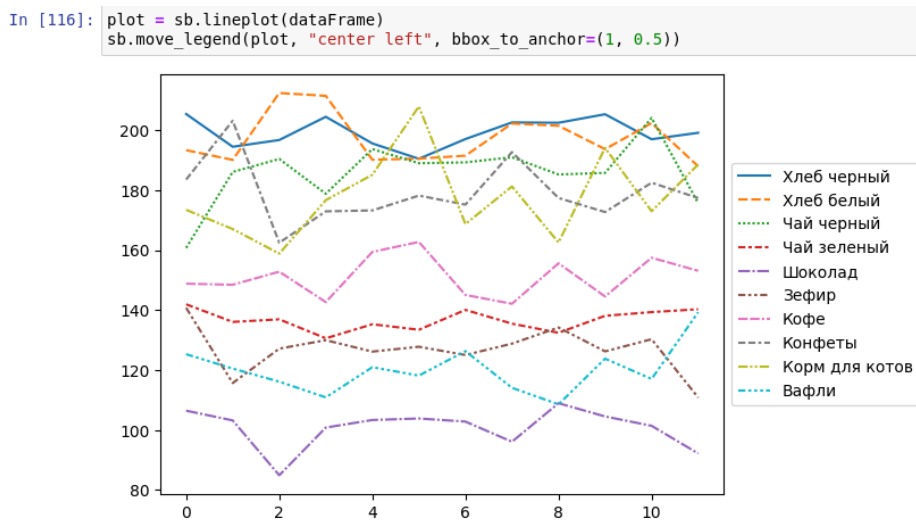


Рисунок 2 - Графическое представление генерируемых данных

3.2 Поиск основных величин для прогнозирования продаж

Поиск основных величин из теоретической части показан на рисунке ниже.

Поиск основных величин для прогнозирования продаж

```
In [117]: p0 = dataframe.sum()/dataframe.shape[0]
p0
```

```
Out[117]: Хлеб черный      199.255174
Хлеб белый      197.297325
Чай черный      185.846939
Чай зеленый     136.684099
Шоколад        100.736650
Зефир          126.931179
Кофе           151.090137
Конфеты        179.309539
Корм для котов  178.126413
Вафли          120.108248
dtype: float64
```

```
In [118]: sdq = ((dataframe - p0)**2).sum()/(dataframe.shape[0]-1)
sdq
```

```
Out[118]: Хлеб черный      23.030254
Хлеб белый      72.265795
Чай черный     113.411305
Чай зеленый     11.860626
Шоколад        44.271873
Зефир          59.898733
Кофе           46.921065
Конфеты        109.622738
Корм для котов  198.962638
Вафли          66.508940
dtype: float64
```

Рисунок 3 - Поиск \hat{p}_0 и $\hat{\sigma}_p^2$

Проведем прогнозирование продаж, данный процесс и его результат показан на рисунке 4.

```
In [119]: x_pred = p0 + np.random.normal(0, sdq, len(p0))
pd.concat([dataframe, pd.DataFrame([x_pred], columns=x_pred.index)]).reset_index(drop=True)
```

```
Out[119]:
```

	Хлеб черный	Хлеб белый	Чай черный	Чай зеленый	Шоколад	Зефир	Кофе	Конфеты	Корм для котов	Вафли
0	205.429191	193.367650	160.767363	141.956082	106.472670	140.776385	148.835215	183.569873	173.456821	125.317394
1	194.506797	190.133561	186.118433	136.087868	103.231337	115.697107	148.484748	203.177669	167.057990	120.524477
2	196.758885	212.435281	190.442735	136.943244	84.930401	127.164914	152.842627	162.497705	158.861979	116.154575
3	204.505951	211.486902	178.842638	130.616344	100.846369	129.998950	142.693845	172.966223	176.700121	110.916353
4	195.615854	190.189593	193.737063	135.300294	103.366260	126.178926	159.430614	173.269291	185.154325	120.977461
5	190.496279	190.539886	189.014211	133.471239	103.876211	127.804144	162.764282	178.223779	207.955666	118.177614
6	197.076813	191.528343	189.300888	140.087955	102.839398	125.050746	145.076314	175.212790	168.732272	126.337672
7	202.628125	202.266250	191.012350	135.500040	96.112581	128.836065	142.134074	192.723604	181.268679	114.111191
8	202.510113	201.513609	185.267620	132.459546	108.966152	134.224876	155.635829	177.423127	162.587954	108.588938
9	205.352583	193.696520	185.781140	138.119932	104.556395	126.257246	144.561867	172.725118	194.139502	123.829776
10	197.019446	202.416554	204.144172	139.351912	101.428658	130.314046	157.496152	182.510822	173.001661	117.046084
11	199.162047	187.993748	175.734652	140.314729	92.213365	110.870745	153.126072	177.414468	188.599990	139.317439
12	209.260602	128.664521	316.420825	142.324598	66.231422	75.111973	75.145042	69.979165	14.181885	129.996195

Рисунок 4 - Предсказание результатов продаж

Проведем проверку полученных результатов на достоверность и соответствие условия стабильности продаж, периода прогноза и периодичность анализа. Выполнение проверок показано на рисунках 5-6.

```
In [120]: rel = sdq / p0
rel

Out[120]: Хлеб черный      0.115582
Хлеб белый      0.366279
Чай черный      0.610240
Чай зеленый     0.086774
Шоколад         0.439481
Зефир           0.471899
Кофе            0.310550
Конфеты         0.611360
Корм для котов  1.116974
Вафли           0.553742
dtype: float64
```

Рисунок 5 - Проверка полученных результатов

```
In [121]: Condition_1 = ((dataFrame - p0) < 2 * sdq).all()
Condition_2 = (p0 > 2 * sdq)
Condition_3 = (dataFrame > 0).all()
Product_color = pd.Series(dtype='string')
for nameProduct in dataFrame.columns:
    if (not Condition_3[nameProduct]):
        Product_color[nameProduct] = 'Красный'
    elif (not Condition_1[nameProduct] and not Condition_2[nameProduct]):
        Product_color[nameProduct] = 'Оранжевый'
    elif (not Condition_1[nameProduct] or not Condition_2[nameProduct]):
        Product_color[nameProduct] = 'Желтый'
    else:
        Product_color[nameProduct] = 'Зеленый'
df_result = {
    'Оценка':rel,
    'Условие№1':Condition_1,
    'Условие№2':Condition_2,
    'Условие№3':Condition_3,
    'Цвет':Product_color
}
df_result = pd.DataFrame(data=df_result)
df_result
```

```
Out[121]:
```

	Оценка	Условие№1	Условие№2	Условие№3	Цвет
Хлеб черный	0.115582	True	True	True	Зеленый
Хлеб белый	0.366279	True	True	True	Зеленый
Чай черный	0.610240	True	False	True	Желтый
Чай зеленый	0.086774	True	True	True	Зеленый
Шоколад	0.439481	True	True	True	Зеленый
Зефир	0.471899	True	True	True	Зеленый
Кофе	0.310550	True	True	True	Зеленый
Конфеты	0.611360	True	False	True	Желтый
Корм для котов	1.116974	True	False	True	Желтый
Вафли	0.553742	True	False	True	Желтый

Рисунок 6 - Проверка полученных результатов

Вывод

В ходе выполнения данной лабораторной работы было проведено прогнозирование продаж для сгенерированных данных. В результате получились данные, которые соответствуют, которые в большинстве условиях достоверности.