Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего образования
«Московский государственный технический университет имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ Информатика и системы управления

КАФЕДРА Системы обработки информации и управления

РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

К НАУЧНО-ИССЛЕДОВАТЕЛЬСКОЙ РАБОТЕ

НА ТЕМУ:

Анализ информации о производителях косметической продукции

Студент ИУ5-34М _____ **С.С. Винников**
(Группа) (Подпись, дата) (И.О.Фамилия)

Руководитель _____ **Ю.Е. Гапанюк**
(Подпись, дата) (И.О.Фамилия)

Консультант _____ _____
(Подпись, дата) (И.О.Фамилия)

2024 г.

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

УТВЕРЖДАЮ

Заведующий кафедрой _____
(Индекс)

_____ _____
(И.О.Фамилия)

« _____ » _____ 20 ____ г.

З А Д А Н И Е
на выполнение научно-исследовательской работы

по теме Алгоритмы, реализующие операции над метаграфами

Студент группы ИУ5-34М

Винников Степан Сергеевич
(Фамилия, имя, отчество)

Направленность НИР (учебная, исследовательская, практическая, производственная, др.)

учебная

Источник тематики (кафедра, предприятие, НИР) НИР

График выполнения НИР: 25% к ___ нед., 50% к ___ нед., 75% к __ нед., 100% к ___ нед.

Техническое задание *Описать алгоритмы, реализующие операции над метаграфами, дать теоретическую оценку временной сложности, реализовать алгоритмы в виде программы, вычислить реальную временную сложность алгоритмов.*

Оформление научно-исследовательской работы:

Расчетно-пояснительная записка на __ листах формата А4.
Перечень графического (иллюстративного) материала (чертежи, плакаты, слайды и т.п.)

Дата выдачи задания « 10 » сентября 2024 г.

Руководитель НИР _____ Ю.Е. Гапанюк
(Подпись, дата) (И.О.Фамилия)

Студент _____ С.С. Винников
(Подпись, дата) (И.О.Фамилия)

Примечание: Задание оформляется в двух экземплярах: один выдается студенту, второй хранится на кафедре.

Содержание

Введение

Для исследования был взят набор данных Factor-Hair-Revised.csv, содержащий данные о различных производителях косметической продукции и удовлетворенностью покупателями этими производителями.

Описание анализируемых данных

Для исследования был взят набор данных Factor-Hair-Revised.csv, содержащий данные о различных производителях косметической продукции и удовлетворенностью покупателями этими производителями.

Набор данных имеет следующие колонки: "Product Quality", "E-Commerce", "Technical Support", "Complaint Resolution", "Advertising", "Product Line", "Salesforce Image", "Competitive Pricing", "Warranty & Claims", "Order & Billing", "Delivery Speed", "Customer Satisfaction".

В качестве целевого признака возьмем атрибут "Customer Satisfaction" – удовлетворенность покупателей продукции.

Выполнение работы

Загрузка данных для анализа:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import r2_score, mean_absolute_error,
mean_squared_error
from sklearn.cluster import KMeans
from sklearn.preprocessing import StandardScaler
```

```
data = pd.read_csv("Factor-Hair-Revised.csv")
data.head()
```

	ID	ProdQual	Ecom	TechSup	CompRes	Advertising	ProdLine
0	1	8.5	3.9	2.5	5.9	4.8	4.9
1	2	8.2	2.7	5.1	7.2	3.4	7.9
2	3	9.2	3.4	5.6	5.6	5.4	7.4
3	4	6.4	3.3	7.0	3.7	4.7	4.7
4	5	9.0	3.4	5.2	4.6	2.2	6.0

Краткое статистическое описание имеющихся данных:

```
data.describe()
```

	ID	ProdQual	Ecom	TechSup	CompRes
count	100.000000	100.000000	100.000000	100.000000	100.000000
mean	50.500000	7.810000	3.672000	5.365000	5.442000
std	29.011492	1.396279	0.700516	1.530457	1.208403
min	1.000000	5.000000	2.200000	1.300000	2.600000
25%	25.750000	6.575000	3.275000	4.250000	4.600000
50%	50.500000	8.000000	3.600000	5.400000	5.450000
75%	75.250000	9.100000	3.925000	6.625000	6.325000
max	100.000000	10.000000	5.700000	8.500000	7.800000

Заметим, первый столбец является ключом, поэтому его можно убрать. Также заметим, что исходные данные не содержат аномальных значений и пропусков, поэтому нет необходимости в дополнительном редактировании набора данных.

Удаление первого столбца-ключа:

```
data = data.drop(columns=['ID'])
```

Корреляционный анализ

Корреляция – это статистическая зависимость между случайными величинами, при которой изменение одной из случайных величин приводит к изменению математического ожидания другой.

Целью корреляционного анализа является выявление оценки силы связи между случайными величинами (признаками), которые характеризует некоторый реальный процесс.

Задачи корреляционного анализа:

а) Измерение степени связности (тесноты, силы, строгости, интенсивности) двух и более явлений.

б) Отбор факторов, оказывающих наиболее существенное влияние на результативный признак, на основании измерения степени связности между явлениями.

в) Обнаружение неизвестных причинных связей.

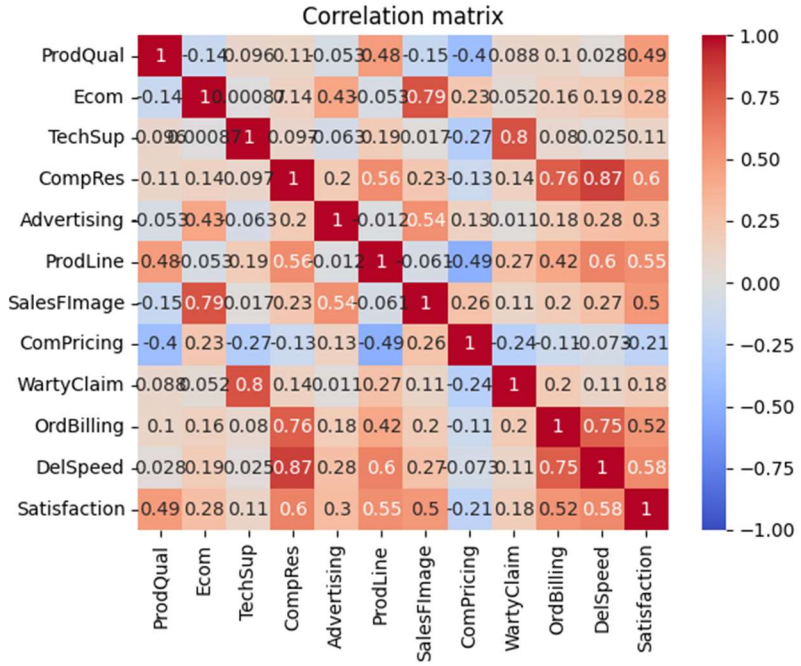
Коэффициент корреляции принимает значения от -1 до 1. Значение ближе к 1 означает сильную положительную корреляцию (т.е. при увеличении одной переменной, другая также увеличивается), значение ближе к -1 означает сильную отрицательную корреляцию (т.е. при увеличении одной переменной, другая уменьшается), а значение около 0 означает отсутствие корреляции.

Корреляционную матрицу найдем с помощью метода corr:

```
data.corr()
```

	ProdQual	Ecom	TechSup	CompRes	Advertising
ProdQual	1.000000	-0.137163	0.095600	0.106370	-0.053473
Ecom	-0.137163	1.000000	0.000867	0.140179	0.429891
TechSup	0.095600	0.000867	1.000000	0.096657	-0.062870
CompRes	0.106370	0.140179	0.096657	1.000000	0.196917
Advertising	-0.053473	0.429891	-0.062870	0.196917	1.000000
ProdLine	0.477493	-0.052688	0.192625	0.561417	-0.011551
SalesFImage	-0.151813	0.791544	0.016991	0.229752	0.542204
ComPricing	-0.401282	0.229462	-0.270787	-0.127954	0.134217
WartyClaim	0.088312	0.051898	0.797168	0.140408	0.010792
OrdBilling	0.104303	0.156147	0.080102	0.756869	0.184236
DelSpeed	0.027718	0.191636	0.025441	0.865092	0.275863
Satisfaction	0.486325	0.282745	0.112597	0.603263	0.304669

Построим тепловую карту:



Отметим, что данные, в основном, коррелируют слабо и средне. Целевой признак наиболее сильно коррелирует с признаком "Complaint Resolution", который характеризует качество и быстроту разрешения покупательских жалоб.

Линейная регрессия

Регрессия ищет отношения между переменными. В рамках регрессионного анализа, находится функция, которая отображает зависимость одних переменных или данных от других. Зависимые данные называются зависимыми переменными, выходами или ответами. Независимые данные называются независимыми переменными, входами или предсказателями.

Линейная регрессия некоторой зависимой переменной y на набор независимых переменных $x = (x_1, \ldots, x_r)$, где r – это число предсказателей, предполагает, что линейное отношение между y и x: $y = \beta_0 + \beta_1 x_1 + \cdots + \beta_r x_r + \varepsilon$. Это уравнение регрессии. β_0, β_1, …, β_r – коэффициенты регрессии, и ε – случайная ошибка.

Линейная регрессия вычисляет оценочные функции коэффициентов регрессии или просто прогнозируемые весы измерения, обозначаемые как b_0, b_1, …, b_r. Они определяют оценочную функцию регрессии $f(x) = b_0 + b_1 x_1 + \cdots + b_r x_r$. Эта функция захватывает зависимости между входами и выходом достаточно хорошо.

Для нашего набора данных в качестве зависимой переменной выберем целевой признак – "Customer Satisfaction", в качестве независимых – "Product Quality", "Complaint Resolution", "Product Line", "Order & Billing", "Delivery Speed".

```
X = data[["ProdQual", "CompRes", "ProdLine", "OrdBilling", "DelSpeed"]]
Y = data["Satisfaction"]
```

Создаем модель линейной регрессии:

```
model = LinearRegression()
```

Рассчитываем модель:

```
model.fit(X, Y)
```

Определим коэффициент детерминации:

```
r_sq = model.score(X, Y)
print(f"коэффициент детерминации: {r_sq}")
```

```
coefficient of determination: 0.5733214080682038
```

Коэффициент детерминации больше, чем 50%, модель можно считать приемлемой.

Определим коэффициенты линейной регрессии:

```
print("Коэффициенты линейной регрессии:", model.coef_, model.intercept_)
```

Коэффициенты линейной регрессии: [0.39073616 0.24182234 -0.02303973 0.06820165 0.53168155] 0.3262178925289927

Используем метод predict для предсказания значения и оценим средние и средне-квадратичные ошибки:

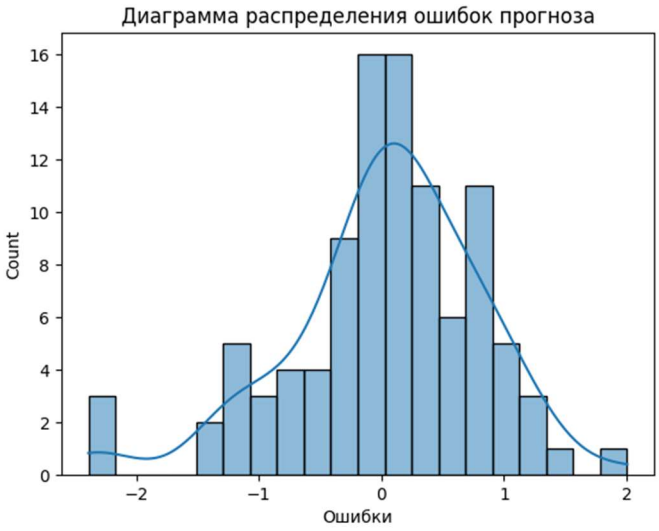
```
Ypred = model.predict(X)

MeanAE = mean_absolute_error(Y, Ypred)
MeanSE = mean_squared_error(Y, Ypred)
print("mae:", MeanAE)
print("mse:", MeanSE)
```

```
r2: 0.5681212504497853
mae: 0.5803806653145462
mse: 0.6073407204024776
```

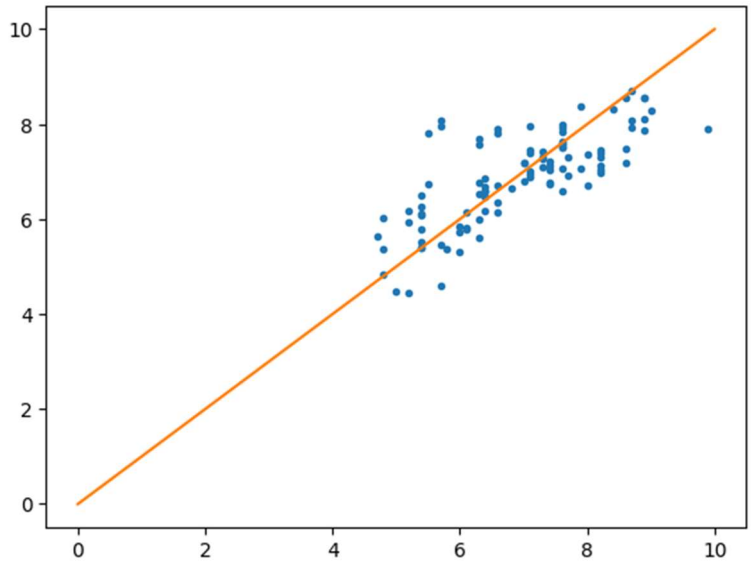
Построим гистограмму ошибок прогнозов модели.

```
sns.histplot(Y-Ypred, bins=20, kde=True)
plt.xlabel("Ошибки")
plt.title("Диаграмма распределения ошибок прогноза")
plt.show()
```



Для наглядности визуализируем зависимость предсказанных значений от реальных значений целевого признака.

```
plt.plot(Y, Ypred, '.')
x = np.linspace(0, 10, 100)
y = x
plt.plot(x, y)
plt.show()
```



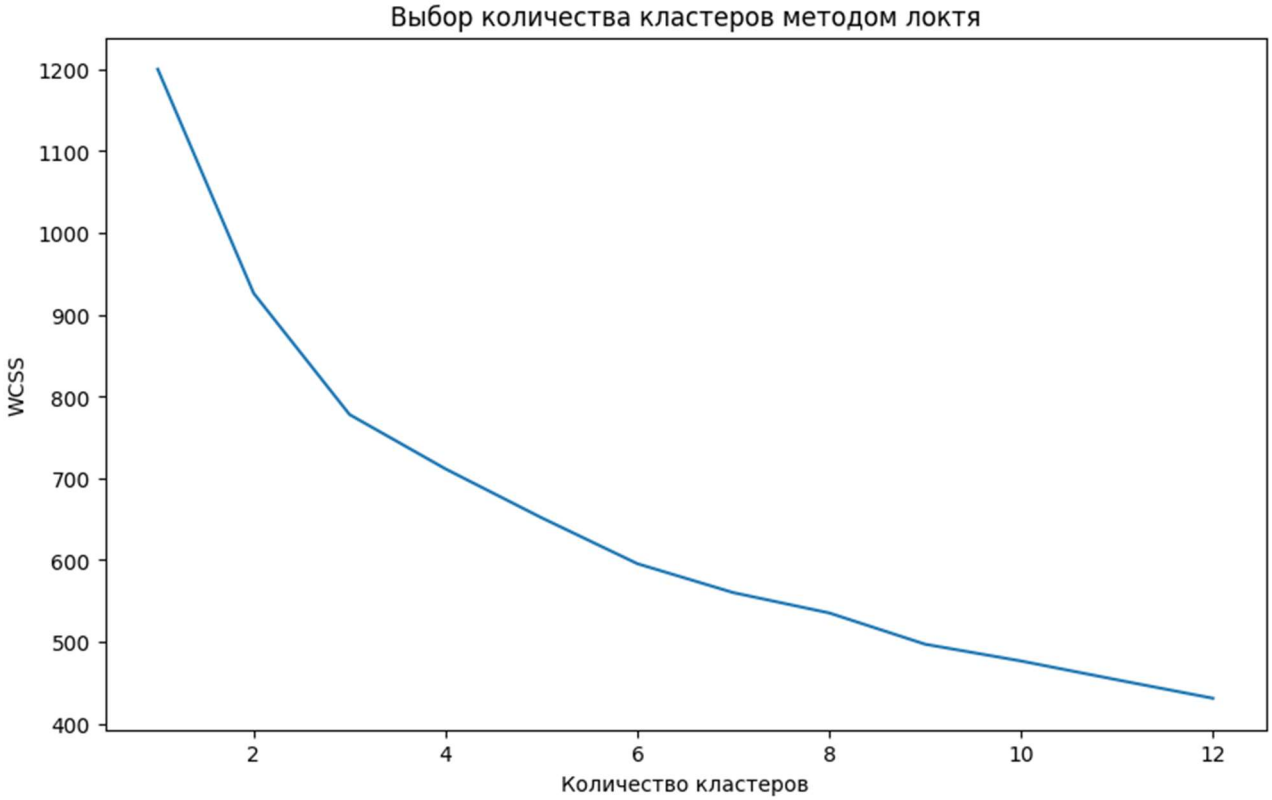
Кластерный анализ

Кластерный анализ – это многомерная статистическая процедура, которая выполняет сбор данных, содержащих информацию о выборке объектов, и затем упорядочивает объекты в сравнительно однородные группы.

Проведем нормализацию кластеров.

```
scaler = StandardScaler()
scaled_data = scaler.fit_transform(data)
```

Выберем количество кластеров методом локтя:



Метод локтя – это графический метод, который помогает определить количество кластеров на основе доли объясненной дисперсии.

Разобьём данные на 3 кластера.

```
cluster_model = KMeans(n_clusters=3, random_state=1, n_init="auto")
cluster_model.fit(scaled_data)
cluster_model.cluster_centers_
```

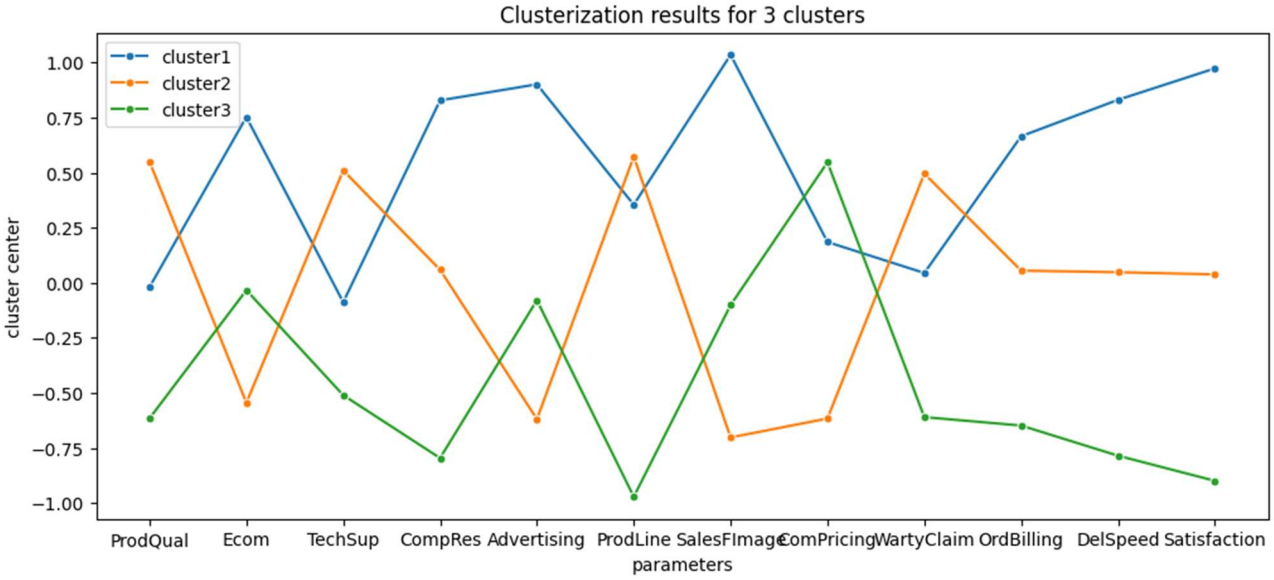
Центры кластеров:

```
array([[-0.01960827,  0.75257955, -0.087974  ,  0.82832313,  0.90136017,
         0.35452611,  1.03527971,  0.18514195,  0.0445182 ,  0.66556386,
         0.83078844,  0.97347819],
       [ 0.54969768, -0.54503874,  0.51031835,  0.05918254, -0.61911559,
         0.57329234, -0.70230049, -0.6164559 ,  0.49577482,  0.05512687,
         0.04796764,  0.03808013],
       [-0.6157537 , -0.03373742, -0.51032882, -0.79606992, -0.07918341,
        -0.97170807, -0.10108161,  0.54715781, -0.61001427, -0.64836888,
        -0.78532228, -0.89933068]])
```

Построим график:

```
plt.figure(figsize=(12, 5))
sns.lineplot(x=range(12), y=cluster_model.cluster_centers_[0], marker='o',
markersize=5, label="cluster1")
```

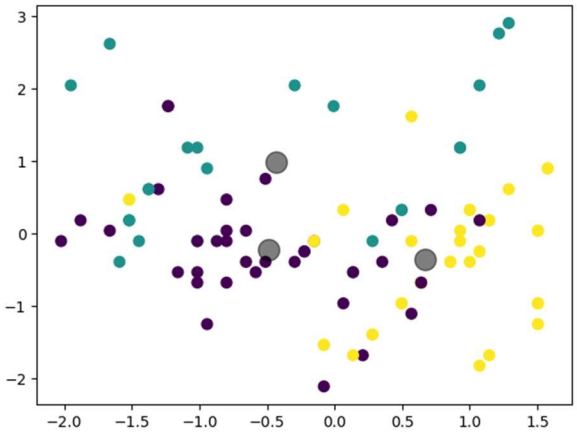
```
sns.lineplot(x=range(12), y=cluster_model.cluster_centers_[1], marker='o',
markersize=5, label="cluster2")
sns.lineplot(x=range(12), y=cluster_model.cluster_centers_[2], marker='o',
markersize=5, label="cluster3")
plt.xticks(range(12), labels=data.columns)
#plt.yscale('log')
plt.xlabel("parameters")
plt.ylabel("cluster center")
plt.title("Clusterization results for 3 clusters")
plt.show()
```



Визуализируем кластеры на двумерном графике:

```
plt.scatter(X[:, 0], X[:, 1], c=y_kmeans, s=50, cmap='viridis')

centers = kmeans.cluster_centers_
plt.scatter(centers[:, 0], centers[:, 1], c='black', s=200, alpha=0.5);
```



Визуализация является не очень наглядной, поскольку учитываются только первые 2 координаты.

Факторный анализ

Факторный анализ (FA) – это исследовательский метод анализа данных, используемый для поиска важных основополагающих факторов или скрытых переменных из набора наблюдаемых переменных. Он помогает в интерпретации данных за счет уменьшения количества переменных. Он извлекает максимальную общую дисперсию из всех переменных и помещает их в общую оценку.

Факторный анализ широко используется в маркетинговых исследованиях, рекламе, психологии, финансах и исследованиях операционной деятельности. Исследователи рынка используют факторный анализ для выявления чувствительных к цене клиентов, выявления особенностей бренда, влияющих на выбор потребителя, и помогает понять критерии выбора канала сбыта.

Импортируем модуль для факторного анализа:

```
import pandas as pd
from statsmodels.multivariate.factor import Factor
df = pd.read_csv("Factor-Hair-Revised.csv")
df = df.drop(columns=['ID'])
df.columns
```

```
Index(['ProdQual', 'Ecom', 'TechSup', 'CompRes', 'Advertising', 'ProdLine',
       'SalesFImage', 'ComPricing', 'WartyClaim', 'OrdBilling', 'DelSpeed',
       'Satisfaction'],
      dtype='object')
```

Посмотрим сведения о данных:

```
df.shape
```

```
(100, 12)
```

Проведем масштабирование данных:

```
scaler = StandardScaler()
df = scaler.fit_transform(df)
```

Создадим модель факторного анализа и запустим её. В качестве аргументов мы указываем наш набор данных, желаемое число факторов и метод, который мы будем использовать. В данном случае “pa” соответствует методу главных компонент.

Само преобразование делается в два шага. Сначала мы «создаём» модель с желаемыми параметрами и сохраняем её как переменную, а затем используем метод fit() и вычисляем результат преобразований.

```
fa = Factor(df, n_factor=3, method='pa')
res = fa.fit()
```

Мы получили таблицу факторных нагрузок. Она отражает корреляцию между фактором и переменной. Подсветим все значения больше 0.6 и меньше -0.6. Их мы будем считать достаточными.

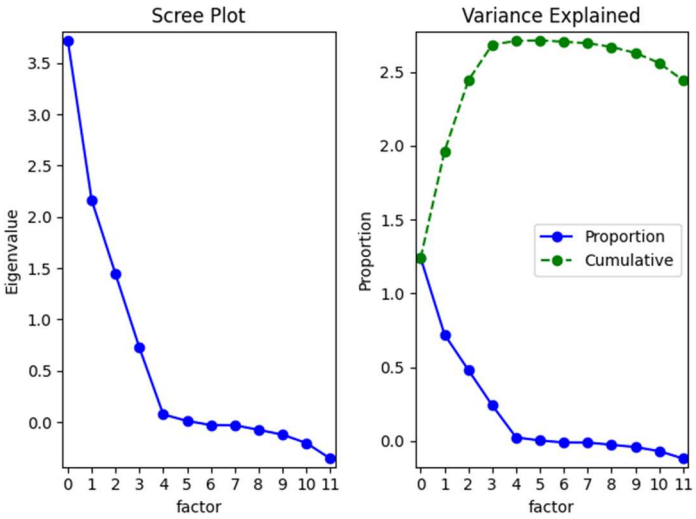
```
res.get_loadings_frame(threshold=0.6)
```

	factor 0	factor 1	factor 2
DelSpeed	0.843772		
CompRes	0.835953		
Satisfaction	0.779325		
OrdBilling	0.727681		
ProdLine	0.672546		
SalesFImage		0.792460	
Ecom		0.640586	
ComPricing		0.539798	
Advertising		0.451964	
ProdQual		-0.370817	
TechSup			-0.732779
WartyClaim			-0.720709

Факторы можно интерпретировать следующим образом:
Factor 0 связан с оценкой компании со стороны покупателей – то, насколько хорошо происходит взаимодействие клиентов с компанией с различных сторон
Factor 1 связан с менеджерской стороной компании – сюда входят показатели продаж, показатель Интернет-торговли и рекламы.
Factor 2 связан с клиентской поддержкой.

Выберем другое число факторов с помощью метода локтя

```
import matplotlib.pyplot as plt
res.plot_scree()
plt.show()
```



Можно сказать, что важными являются 4 фактора – на них приходится набольшая доля объяснённой дисперсии. Объясненная дисперсия используется для измерения доли изменчивости прогнозов модели машинного обучения. Проще говоря, это разница между ожидаемым значением и прогнозируемым значением. Очень важно понимать, сколько информации мы можем потерять, сверяя набор данных.

Рассмотрим 4 фактора:

```
fa = Factor(df, n_factor=4, method='pa')
```

```
res = fa.fit()
res.get_loadings_frame(threshold=0.6)
```

	factor 0	factor 1	factor 2	factor 3
DelSpeed	0.864802			
CompRes	0.842932			
Satisfaction	0.812538			
OrdBilling	0.728887			
ProdLine	0.666167			
SalesFImage		0.819548		
Ecom		0.627880		
ComPricing		0.544637		
Advertising		0.443034		
WartyClaim			-0.755322	
TechSup			-0.732550	
ProdQual				-0.606818

Factor 0 остался аналогичным, Factor 1 все так же связан с менеджментом, Factor 2 – с клиентской поддержкой, а Factor 3 – с качеством продукции.

Заключение

В ходе выполнения курсовой работы нами исследовался набор данных Factor-Hair-Revised.csv с помощью различных аналитических методов. Факторный анализ показал, что ключевыми факторами являются: фактор оценки компании со стороны её клиентов, фактор менеджмента внутри компании, фактор клиентской поддержки и фактор качества продукции.

Список литературы

1. https://www.reg.ru/blog/sreda-razrabotki-jupyter-notebook/
2. https://habr.com/ru/articles/690414/
3. https://www.kaggle.com/code/ipravin/factor-analysis-and-linear-programming/notebook