# Подготовка и оценка набора многомерных данных

**Описание набора многомерных данных.**

Набор данных содержит в себе информацию о характеристиках заёмщиков и статусах их заявок на кредиты, такие как:

1. Возраст.
2. Профессия.
3. Семейное положение.
4. Образование.
5. Наличие дефолтов в прошлом.
6. Статус наличия ипотеки.
7. Статус наличия счета в банке
8. Тип связи с клиентом
9. Длительность последнего контакта
10. Количество контактов с клиентом
11. Статус предыдущей заявки на кредит
12. Статус текущей заявки на кредит

Для решения задачи анализа кредитных заявок были определены наиболее влияющие на решение о выдаче кредита характеристики заёмщиков. К ним относятся:

1. Наличие дефолтов в прошлом.
2. Возраст.
3. Статус о наличии непогашенного кредита.
4. Профессия.

В ходе написания главы был проанализирован вышеописанный датасет на предмет наиболее встречающихся значений у заёмщиков по характеристикам профессия и возраст.

На рисунке 1 приведена статистика заёмщиков по профессиям.

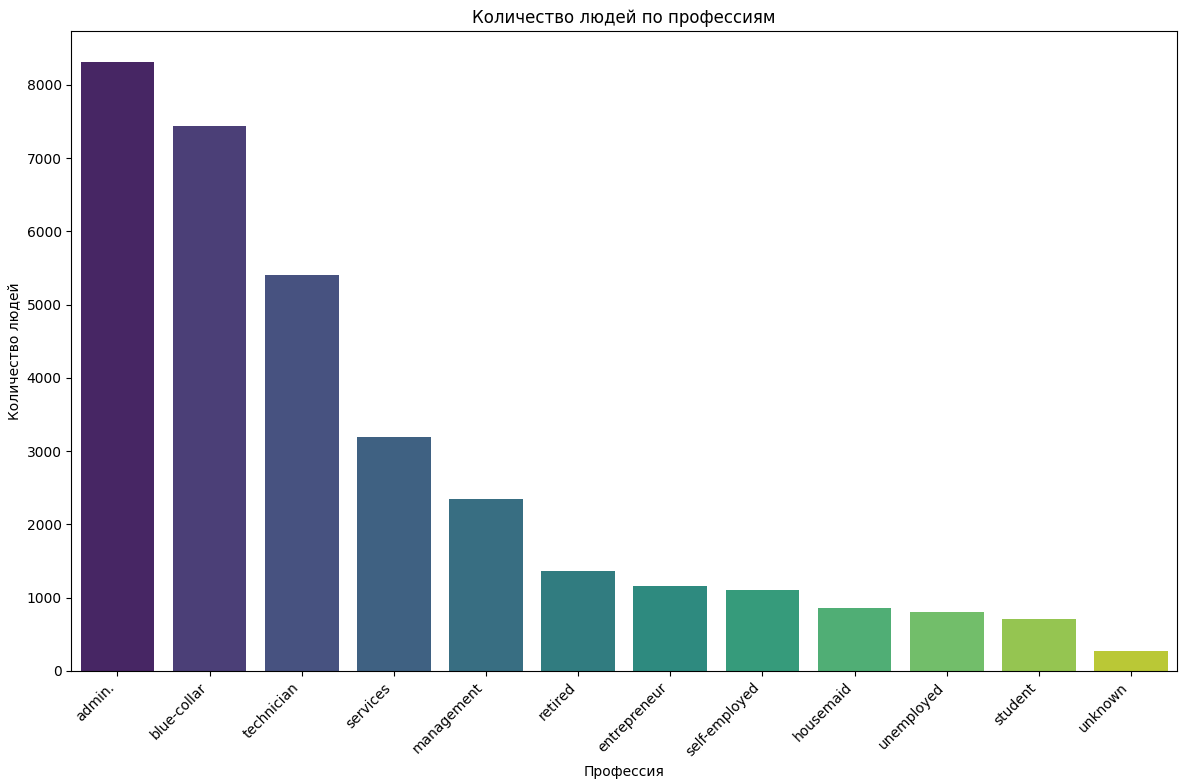


Рисунок 1. Статистика заёмщиков по профессиям

На рисунке 3 представлена статистика разбиения заёмщиков на группы по возрастам, где возраст человека принадлежит конкретной возрастной группе:

1. Группа «Молодежь» - возраст от 18 до 29 лет
2. Группа «Средний возраст» - возраст от 30 до 44 лет
3. Группа «Высокий возраст» - возраст от 45 до 59 лет
4. Группа «Пожилые» - возраст от 60 лет и выше.

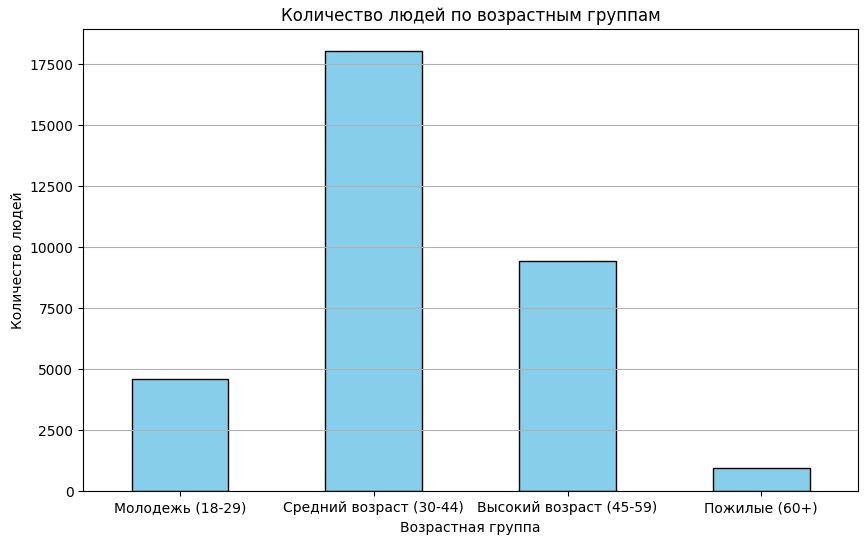


Рисунок 3. Статистика разбиения заёмщиков на группы по возрастам.

# Кластеризация многомерных данных

В ходе написания главы был проанализирован массив данных, содержащих в себе данные о заявках на кредит, характеристики заявителей и статус заявок.

В ходе написания главы был выполнен сравнительный анализ следующих методов кластеризации, представленных в python-библиотеке sklearn.cluster:

1. KMeans.
2. Birch.
3. BisectingKMeans.

В качестве характеристик заявителей, представленных для кластеризации, были выбраны такие как возраст и баланс банковской карты. Данный датасет был разбит на четыре кластера. Изображения кластеризации ниже представлены в осях возраст-баланс карты.

Задачей, решаемой путем кластеризации, является уменьшение размера, исследуемого датасета путем определения наиболее незначимых групп экземпляров.

На рисунке 4 представлен результаты работы методы кластеризации Kmeans. Действие алгоритма заключается в том, что он стремится минимизировать суммарное квадратичное отклонение точек кластеров от центров этих кластеров:

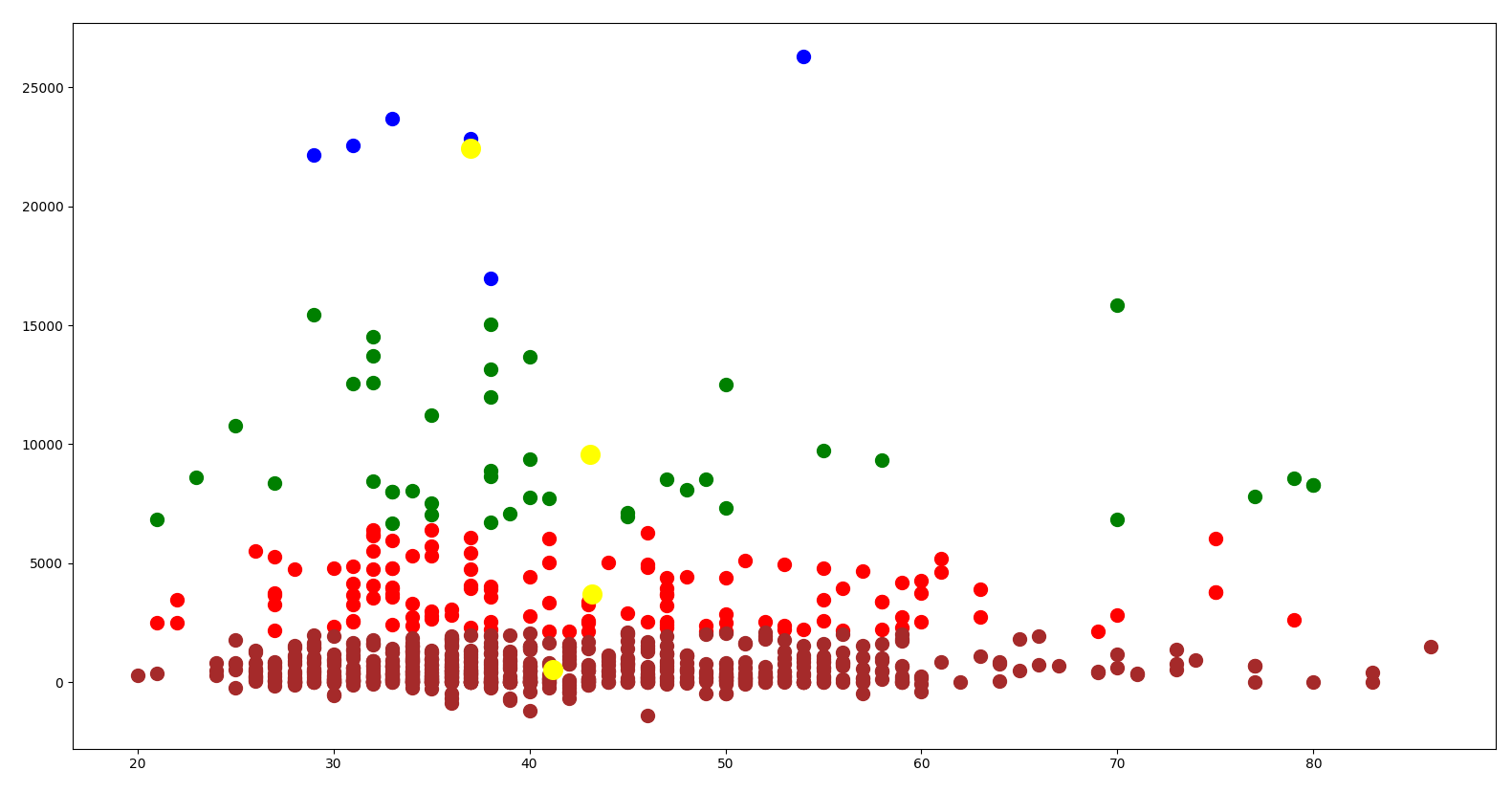


Рисунок 4. Результаты работы методы кластеризации Kmeans.

Как видно из рисунка – алгоритм KMeans достаточно четко разделил границы кластеров и определил центроиды. Явно видно, что для уменьшения размера датасета можно отбросить кластер синего цвета как самый незначимый.

На рисунке 5 показан результат работы методы кластеризации Birch. BIRCH (Balanced Iterative Reducing and Clustering hierarchies) - это иерархический алгоритм кластеризации, который предназначен для эффективной обработки больших наборов данных. Алгоритм строит древовидную структуру кластеров путем рекурсивного разделения данных на подкластеры до тех пор, пока не будет выполнен критерий остановки:

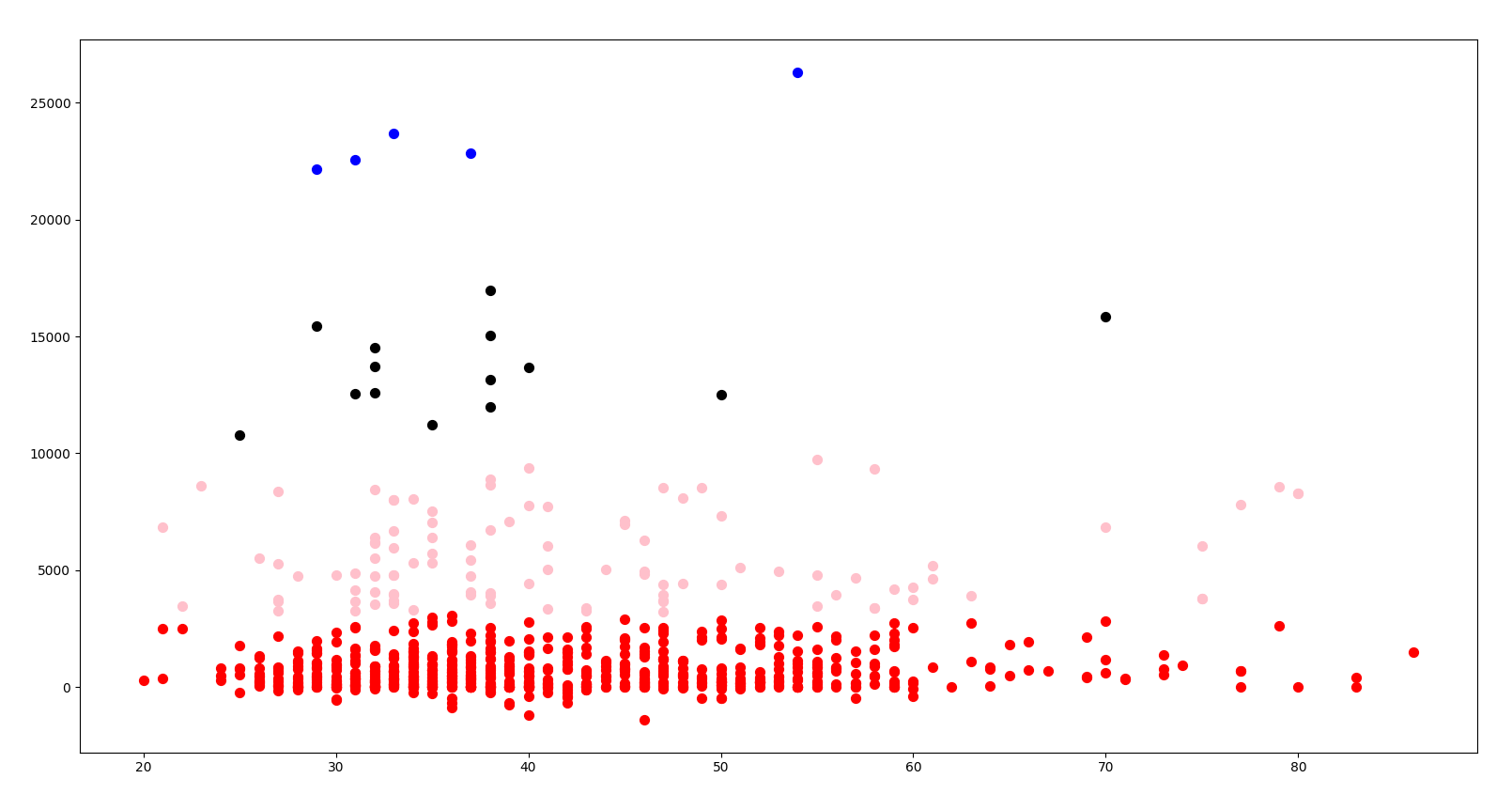


Рисунок 5. Результаты работы методы кластеризации Birch.

Результат несколько отличается от результата работы KMeans. Так определенные пограничные объекты были отнесены к иным кластеры.

В целом, кластеры были определены в схожих границах. Самым незначимым кластером оказался кластер синего цвета.

На рисунке 6 изображен результат работы методы кластеризации BisectingKMeans. Принцип работы основан на том, что создается один кластер, затем он делится на два кластера, используя KMeans. Полученные кластеры в свою очередь делятся, пока количество кластеров не достигнет заданного значения или становится невозможно их больше разделить.

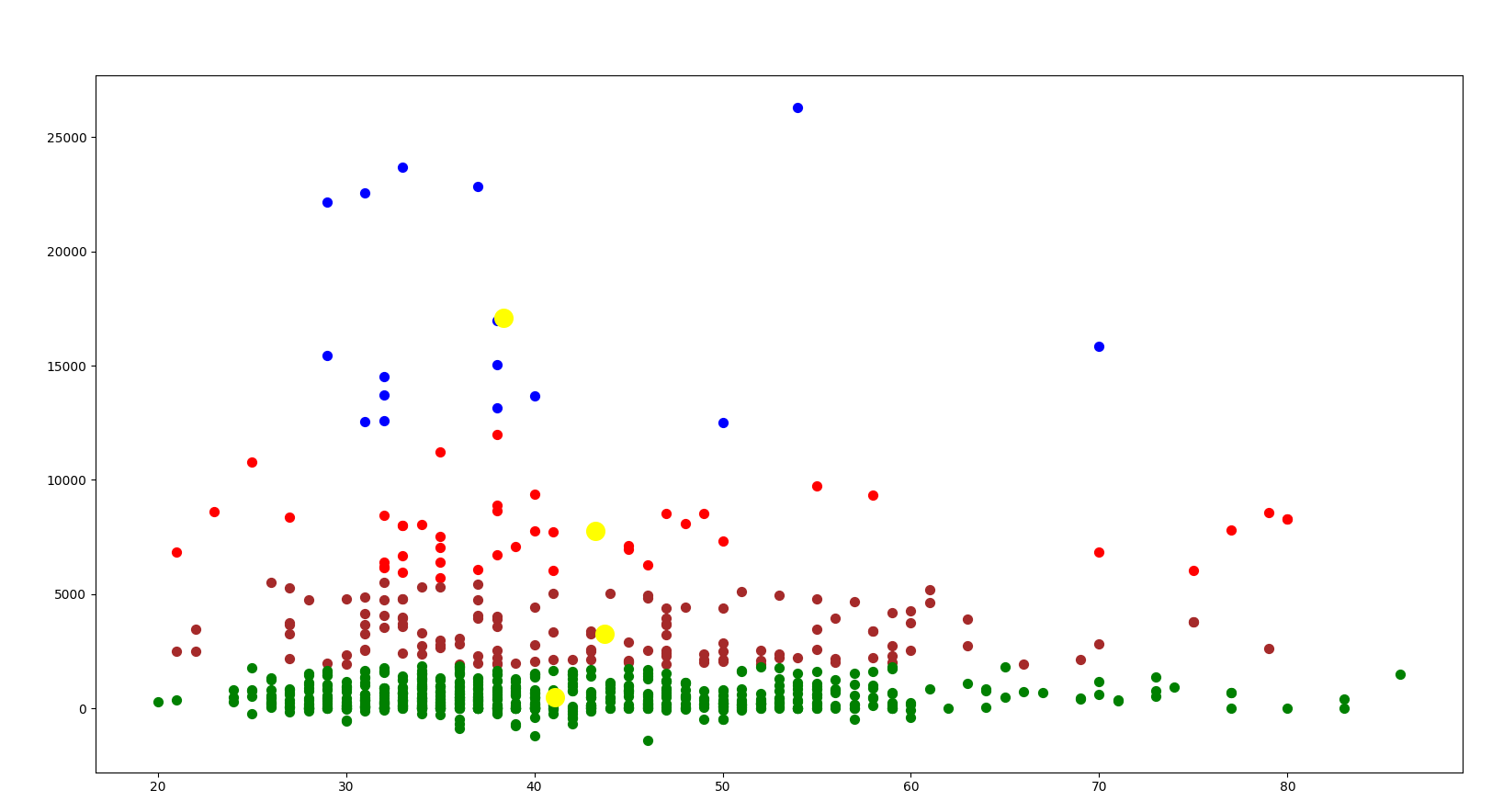


Рисунок 6. Результаты работы методы кластеризации BisectingKMeans.

По сравнению с двумя предыдущими алгоритмами, данный метод определил кластера более сбалансированными по оси Y (баланс карты). Хоть в верхнем синем кластере не так много объектов, но нижняя граница баланса понизилась, за счет чего синий кластер стал более многочисленным, по сравнению с предыдущими алгоритмами. В данном случае уже нельзя однозначно сказать, какой кластер можно считать самым незначительным.

Кластеризация датасета методом Kmeans

На рисунке 7 можно выделить 4 кластера:

1. Клиенты с небольшим балансом
2. Клиенты со средним балансом
3. Клиенты с высоким балансом
4. Клиенты с очень высоким балансом

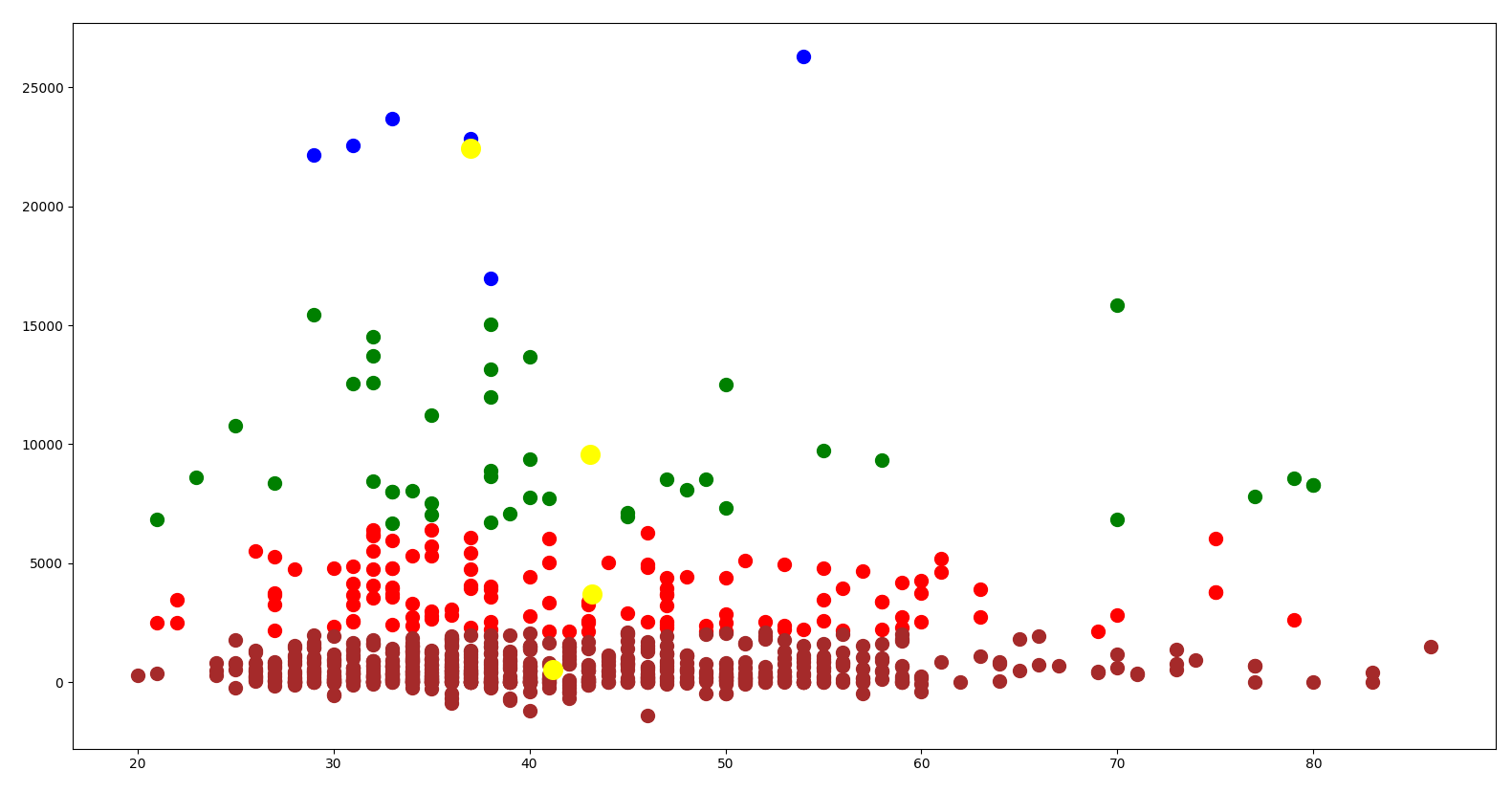


Рисунок 7. Результаты работы методы кластеризации Kmeans.

Можно убрать из датасета элементы, принадлежащие к кластеру «клиенты с очень высоким доходом»

На рисунке 8 можно выделить следующие кластеры:

1. Молодые клиенты
2. Клиенты среднего возраста
3. Клиенты высокого возраста
4. Клиенты пожилого возраста

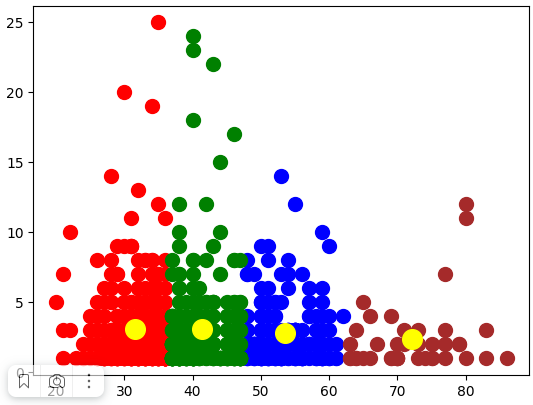


Рисунок 8. Результаты работы методы кластеризации Kmeans по возрасту заемщиков.

В результате мы видим, что для уменьшения объема датасета можно отбросить кластер с пожилыми людьми как самый незначительный.

# Корреляционный анализ многомерных рядов

**Описание набора временных рядов**

Для выполнения анализа использованы датасеты с сайта экономических исследований [https://research.stlouisfed.org](https://research.stlouisfed.org/) по стране «Малайзия» Датасеты представлены различными государственными источниками Малайзии.

Далее из различных наборов данных по экономическим показателям выбраны три датасета.

*Рентабельность собственного капитала для Банка Малайзии [1].* Рентабельность собственного капитала банка Малайзии изменяется с течением времени. Описываемый период – 1 января 2000 – 1 января 2024 гг.

Датасет представляет собой таблицу из двух полей: «DATE» и «VALUE». Графически представлен на рисунке 9.

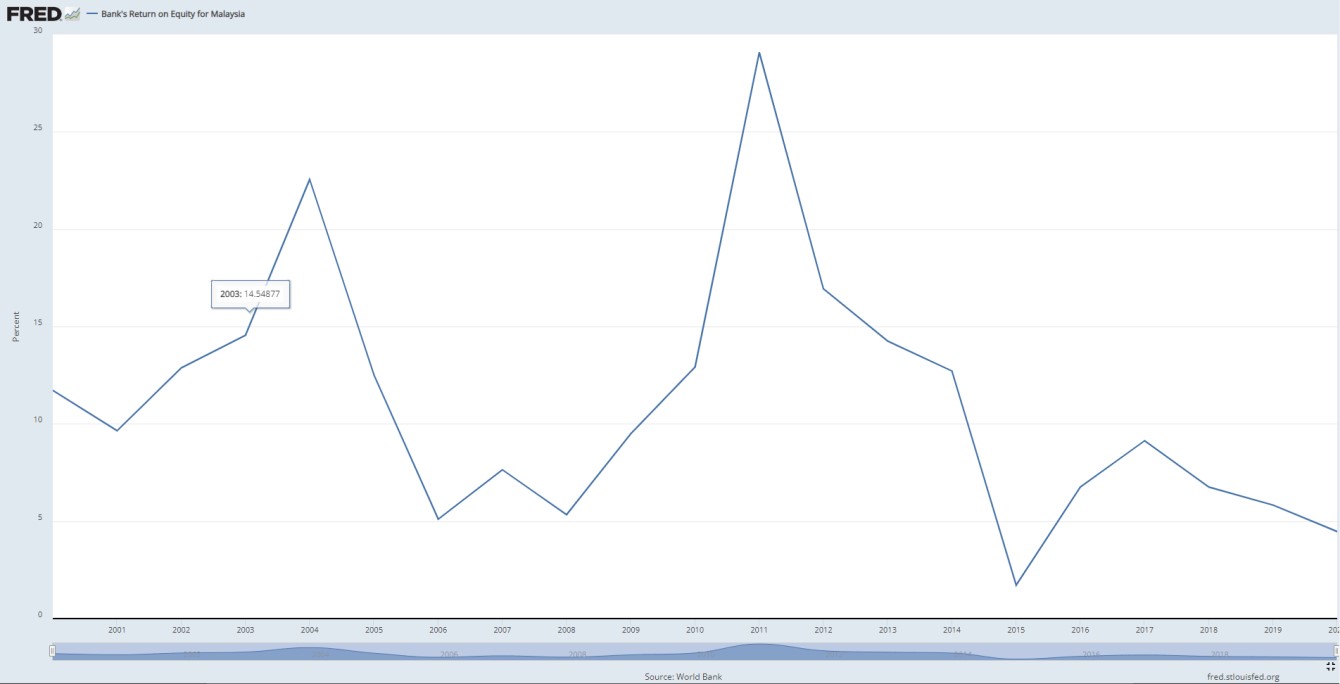


Рисунок 9 – временной ряд рентабельности собственного капитала для

Банка Малайзии

*Инфляция потребительских цен в Майлазии [2].* Датасет представляет собой таблицу из двух полей: «DATE» и «VALUE». Описываемый период – 1 января 1960 – 1 января 2022 гг. График представлен на рисунке 10.

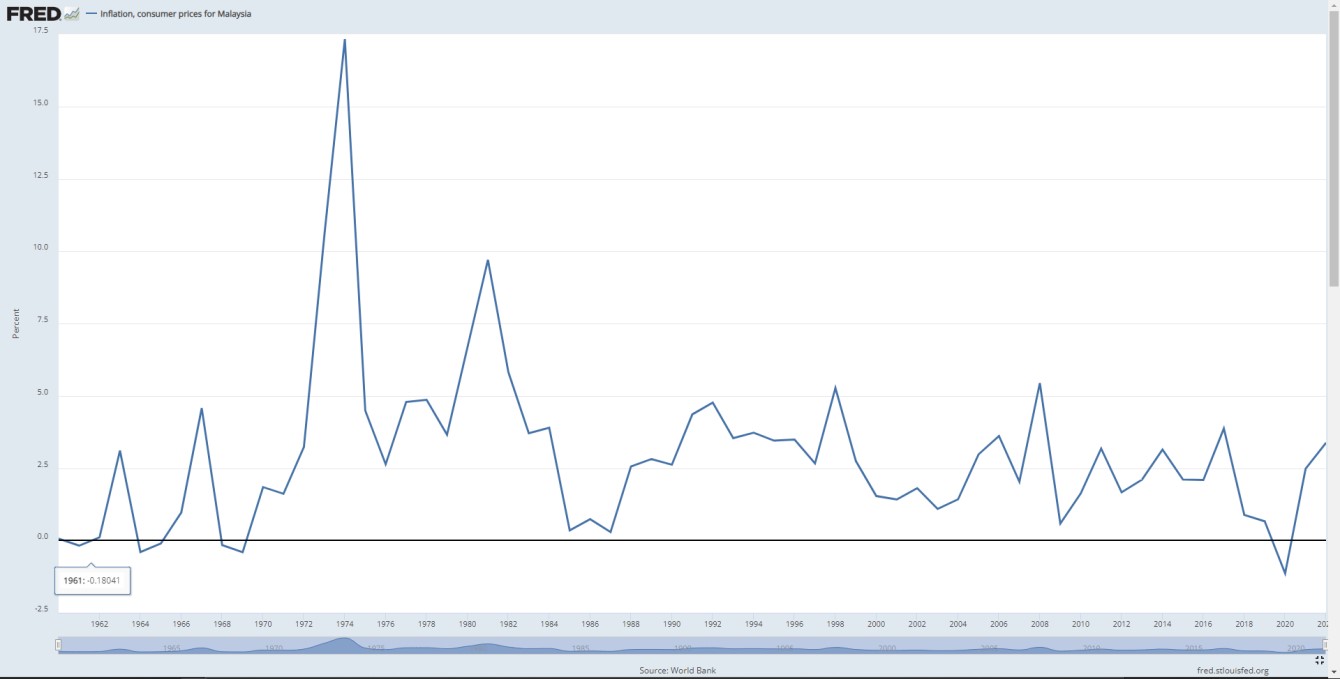


Рисунок 10 – инфляция потребительских цен в Майлазии

*Цены на жилую недвижимость в Малайзии [3].* Датасет содержит столбцы “Date” и “Value”. Описываемый период - 1 января 1988 - 1 июля 2023 гг.

Графически представлен на рисунке 11.



Рисунок 11 – временной ряд цен на жилую недвижимость в Малайзии

**Описание метода корреляционного анализа**

Временные ряды являются одним из важных компонентов современной аналитики данных и имеет большие практические применения в различных областях.

Анализ временных рядов является важным инструментом для многих областей, в том числе бизнеса, науки, технологий и экономики. Методы анализа временных рядов помогают понять поведение и изменение временных данных и выявить скрытые тенденции и закономерности. Существует три основных метода анализа временных рядов: стационарность, автокорреляцию и спектральный анализ.

Корреляция - это степень взаимосвязи переменных можно измерить. Для двух количественных переменных можно воспользоваться коэффициентом Пирсона. Диапазон значений этого коэффициента находится в пределах от минус единицы (отрицательная корреляция, если один показатель увеличивается, другой уменьшается и наоборот) до плюс единицы (положительная корреляция, если один увеличивается, другой тоже увеличивается и наоборот). Ноль означает отсутствие корреляции.

Корреляция показывает силу взаимосвязи двух переменных и позволяет строить модель.

Автокорреляция также показывает степень взаимосвязи в диапазоне от – 1 до 1, но только не двух переменных, а одной и той же переменной в разные моменты времени.

Помимо этого, в лабораторной работе будет рассмотрен и применен на практике вариант прогнозирования временных рядов по выборке максимального подобия. Согласно этого метода подобие выборок, то есть свойство, заключенное в том, что одна выборка может быть выражена через другую, может быть выражена с помощью линейной регрессии.

Линейность корреляции проявляется в том, что точки расположены вдоль прямой линии. Положительный или отрицательный наклон такой линии определяется направлением взаимосвязи.

**Описание результатов корреляционного анализа.**

Анализ проведем в блокноте Google Colab. Для решения с датасетом [1] создадим код программы, представленный ниже.

Сначала импортируем все библиотеки.

|  |
| --- |
| import numpy as np # linear algebra  import pandas as pd # data processing, CSV file I/O (e.g. pd.read\_csv) from sklearn.linear\_model import LinearRegression from matplotlib import pyplot as plt from pandas.plotting import register\_matplotlib\_converters register\_matplotlib\_converters()  from google.colab import drive drive.mount('/content/drive') |

Теперь зададим параметры для прогнозирования временного ряда. В данной лабораторной будем прогнозировать диапазон с 2020.01.01 длиной в

30 месяцев, чтобы наглядно были видны значения ошибок.

|  |
| --- |
| forecast\_moment = pd.datetime(2020, 1, 1)  forecast\_horizon = 30 column\_name = 'CPALTT01ZAM659N' time\_index\_name = 'DATE'  M = 20 |

Далее загружаем датасет

|  |
| --- |
| dataset = r'Consumer Price Index.csv' data\_source = pd.read\_csv(f'drive/MyDrive/Colab  Notebooks/datasets/{dataset}', delimiter=',', low\_memory=False) data\_source.index = pd.to\_datetime(data\_source[time\_index\_name], format='%Y-%m-%d') data\_source.index.name = time\_index\_name data = data\_source[column\_name] |

Определи крайний доступный паттерн временного ряда:

|  |
| --- |
| ### pattern\_latest\_available\_range = pd.date\_range(forecast\_moment - pd.DateOffset(months=M),  forecast\_moment -  pd.DateOffset(months=1), freq='MS') pattern\_latest\_available = data.loc[pattern\_latest\_available\_range] |

В цикле определим наиболее подобный отрезок временного ряда. Результатом работы цикла являются максимальные значения.

###

looping\_dates\_range

=

pd.date\_range

(

data.index

[

0

]

,

forecast\_moment

-

pd.DateOffset

(

months=M

+

forecast\_horizon

)

,

freq=

'MS'

)

similarity\_measure =

[]

time\_delay

=

[]

#

Looping

through

timeseries

history

for

d

in

looping\_dates\_range

:

pattern\_temp\_range = pd.date\_range

(

d

,

d + pd.DateOffset

(

months=M

-

1

)

,

freq=

'MS'

)

ds = data.loc

[

pattern\_temp\_range

]

.values

time\_delay.append

(

d

)

if

np.

sum

(

ds

)

==

0

:

similarity\_measure.append

(

0

)

else

:

#

Similarity

measure

=

abs

of

linear

correlation

c

=

np.

abs

(

np.corrcoef

(

ds

,

pattern\_latest\_available.values

))

similarity\_measure.append

(

c

[

0

,

1

])

Находим паттерн, с максимальным подобием:

|  |
| --- |
| ### similarity = pd.DataFrame(similarity\_measure, index=time\_delay, columns=['similarity']) max\_similarity = np.max(similarity.values) max\_time\_delay = similarity[similarity.values == max\_similarity].index max\_similarity\_pattern\_range = pd.date\_range(max\_time\_delay[0],  max\_time\_delay[0] +  pd.DateOffset(months=M - 1), freq='MS') max\_similarity\_pattern = data.loc[max\_similarity\_pattern\_range] |

Определим базовый паттерн:

|  |
| --- |
| ### base\_pattern\_range = pd.date\_range(max\_similarity\_pattern.index[-1] + pd.DateOffset(months=1), max\_similarity\_pattern.index[-1] +  pd.DateOffset(months=forecast\_horizon),  freq='MS')  base\_pattern = data.loc[base\_pattern\_range] |

Определим предсказанные значения:

### forecast\_range = pd.date\_range(pattern\_latest\_available.index[-1] + pd.DateOffset(months=1), pattern\_latest\_available.index[-1] +

pd.DateOffset(months=forecast\_horizon),

freq='MS')

x = np.column\_stack((base\_pattern.values, np.ones(len(base\_pattern)))) y = regress.predict(x)

forecast = pd.DataFrame(y, index=forecast\_range, columns=[column\_name])

Дополнительно просчитаем значения средней абсолютной ошибки (MAE) и средней абсолютной ошибки в процентах (MAPE):

|  |
| --- |
| ###  actual = data.loc[forecast\_range]  mae = np.mean(np.abs(actual.values.ravel() - forecast.values.ravel())) mape = np.mean(np.abs((actual.values.ravel() - forecast.values.ravel()) / actual.values.ravel())) \* 100 error\_line = 'MAE = %2.2f, MAPE = %2.2f %% ' % (mae, mape) |

Выведем результаты на графиках:

|  |
| --- |
| ### fig, (ax1, ax2) = plt.subplots(nrows=2, figsize=(20, 20)) ax1.plot(actual.index, actual.values, label='Actuals') ax1.plot(forecast.index, forecast.values, label='Forecast') ax1.set\_title('Forecast for dataset: ' + dataset + ': ' + error\_line, fontsize=20) ax1.legend()    ax2.plot(max\_similarity\_pattern.index, pattern\_latest\_available.values, label='Latest available pattern (x-shifted)') ax2.plot(max\_similarity\_pattern.index, max\_similarity\_pattern.values, label='Max similarity pattern') ax2.plot(max\_similarity\_pattern.index, max\_similarity\_pattern\_model, label='Model of max similarity pattern') ax2.plot(base\_pattern.index, base\_pattern.values, label='Base pattern values') ax2.plot(base\_pattern.index, forecast.values, label='Forecast (x- shifted)')  ax2.legend()  plt.subplots\_adjust(hspace=0.3) plt.show() |

Результат прогнозирования представлен на рисунке 12. Как видно из графика и заголовка – значение MAPE довольно велико и составляет 48,32%. Высокое значение так же может быть связано с малой волатильность данного показателя.

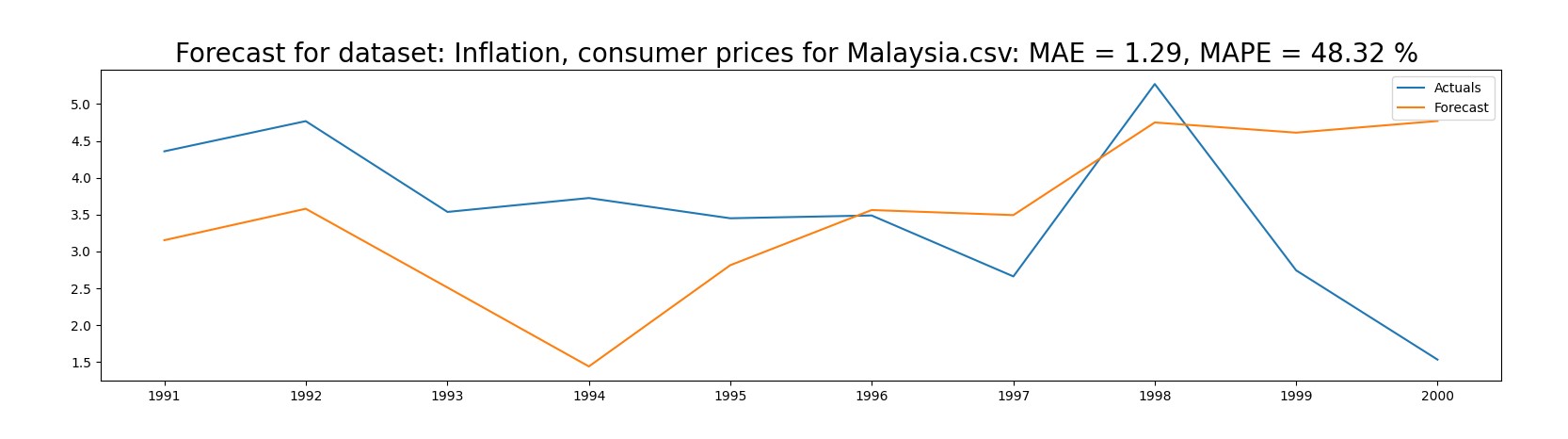


Рисунок 12 – результат прогнозирования датасета 1.

На рисунке 13 представлены результаты поиска наиболее подобного отрезка (по максимальному значению коэффициента линейной корреляции).

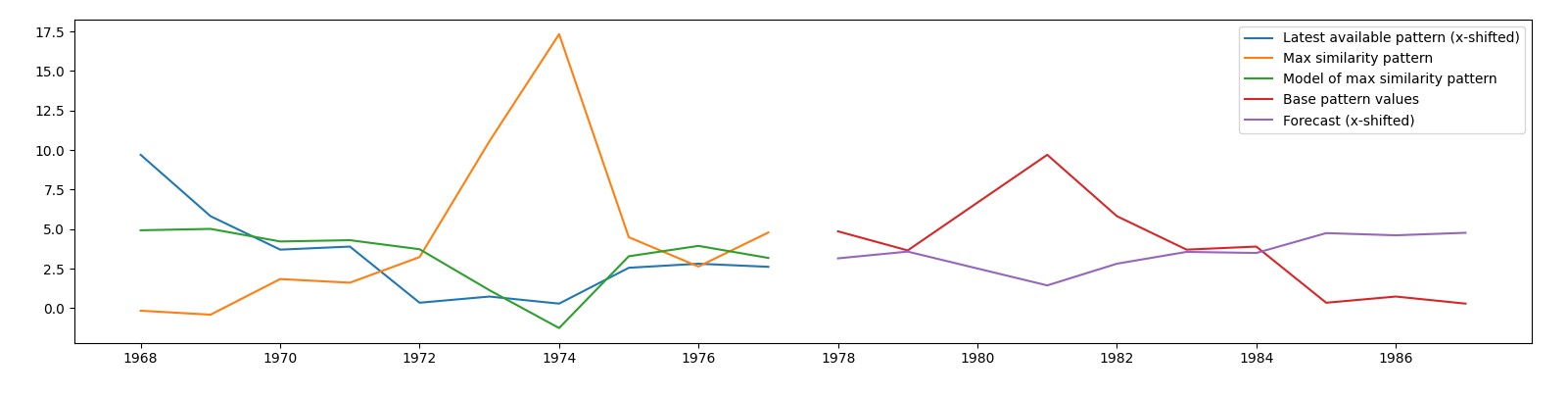


Рисунок 13 – результаты поиска подобного отрезка датасета 1 Повторим эксперимент на датасете 2.

На рисунке 14 виден графический результат предстаказанных значений по сравнению с реальными данными. Так как значения не сильно менялись за указанный период, предсказанные значения также не очень колеблятся.

На рисунке 15 представлены результаты поиска наиболее подобного отрезка (по максимальному значению коэффициента линейной корреляции).

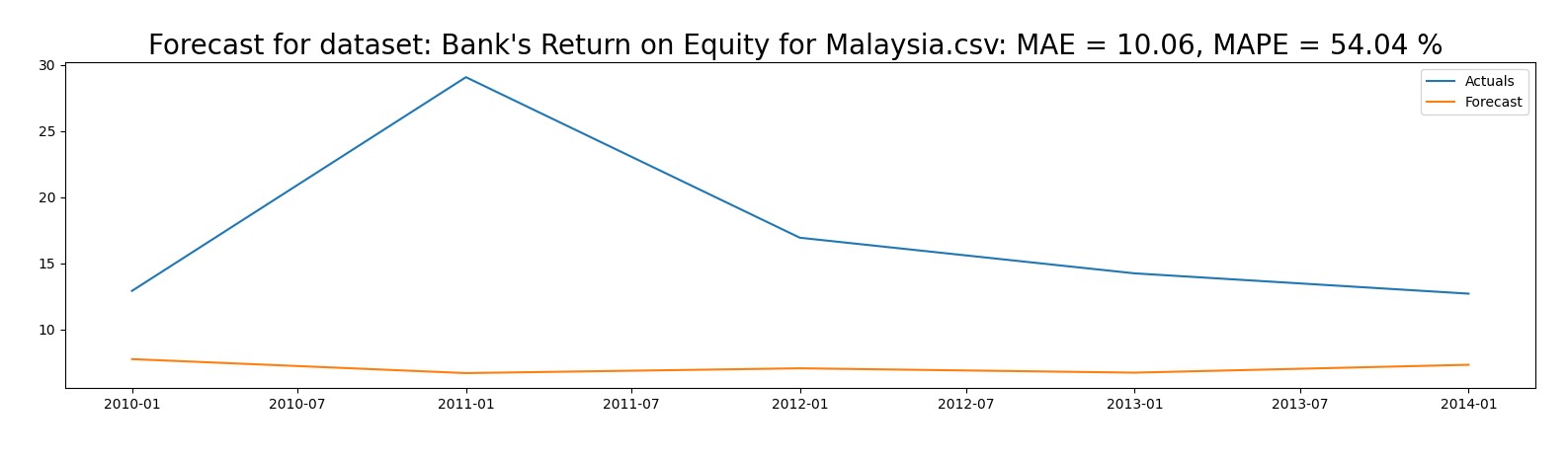


Рисунок 14 – результат прогнозирования датасета 2

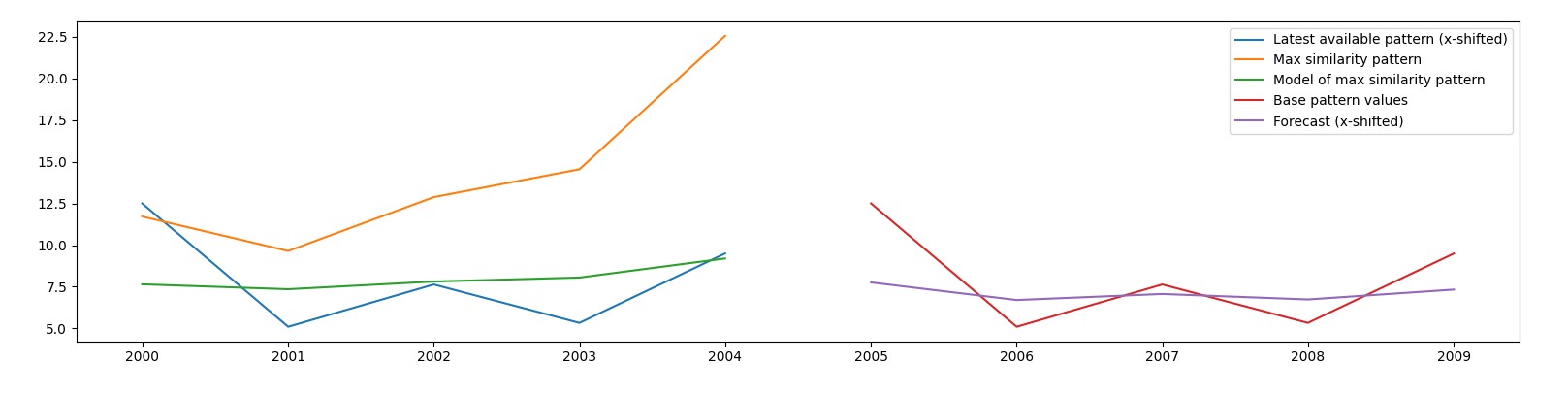


Рисунок 15 – результаты поиска подобного отрезка датасета 2

Повторим эксперимент на датасете 3.

В этот раз на рисунке 16 видно, что график предсказанных значений на этот раз имеет относительно небольшую абсолютную ошибку в 4,45%. Что интереснее на этот раз представлено на рисунке 17 – на этот раз удалось благодаря коэффициенту линейной корреляции найти максимально похожие подобные отрезки.

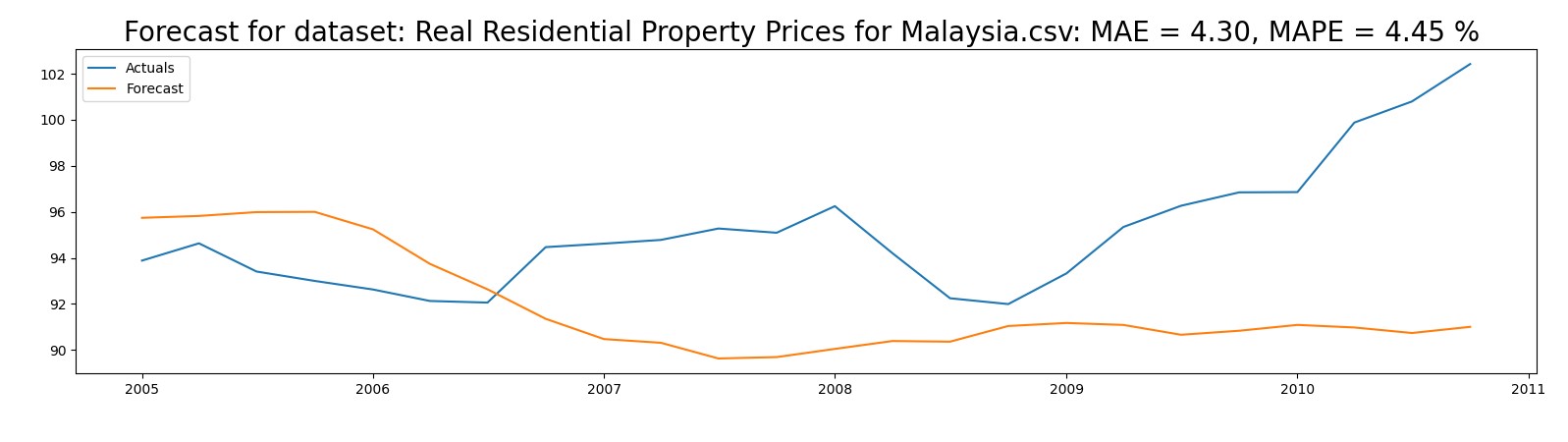


Рисунок 16 - результат прогнозирования датасета 3

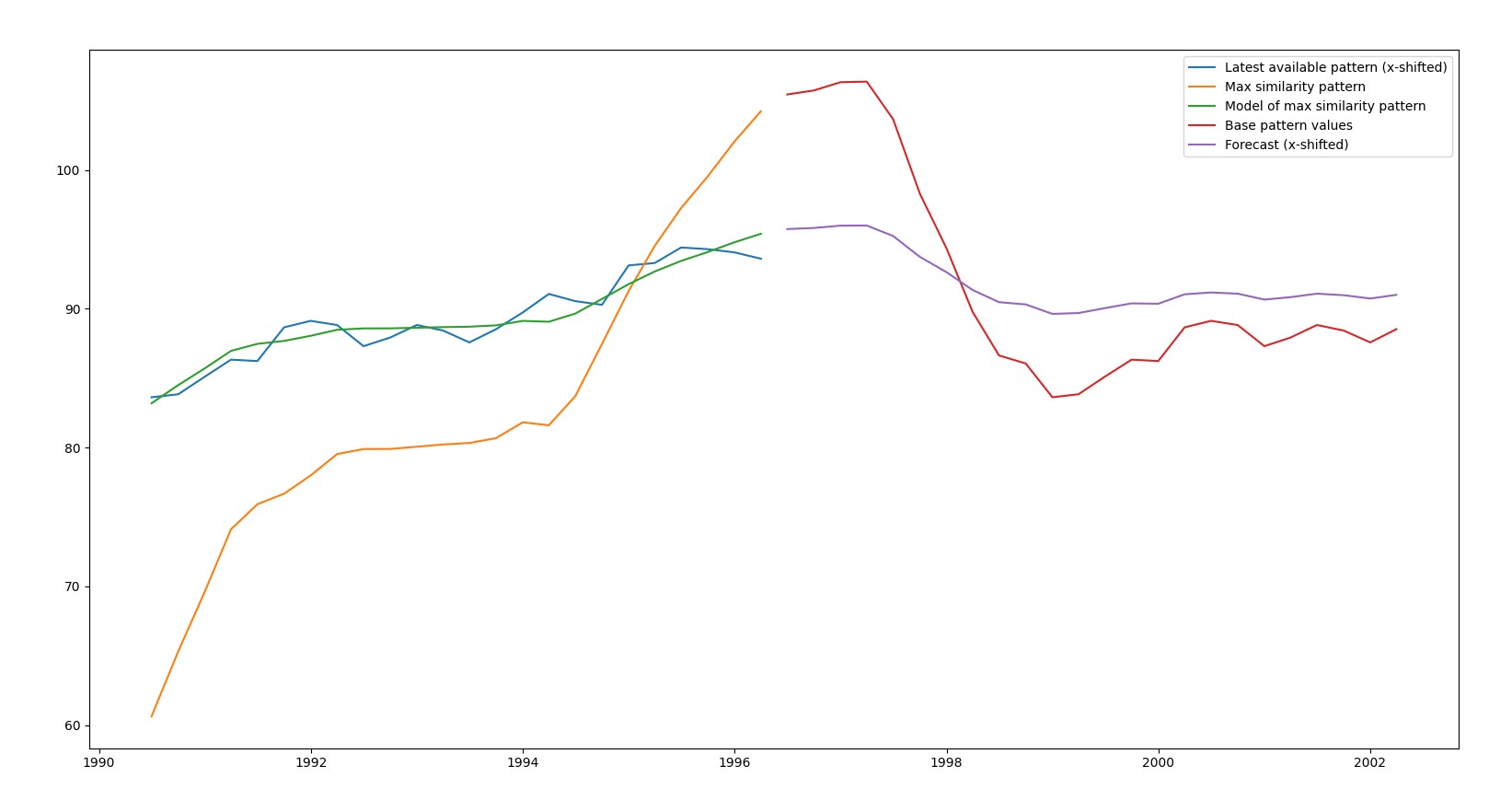


Рисунок 17 – результаты поиска подобного отрезка датасета 3

# Заключение

По результатам проведенной работы были проанализированы датасеты заявок клиентов на получение кредитов. В ходе анализа были определены наиболее многочисленные и малочисленные группы клиентов. Отбрасывание самых незначительных групп позволяет существенно сократить объем датасета при неизменном качестве обучения нейронной сети на указанном датасете. Также это дает возможность выделить наиболее типичные группы заёмщиков по различным характеристикам. По результатам проведенного анализа можно сделать вывод, что больше всего заёмщиков по кредитам относятся к средней возрастной группе. Так же можно сделать вывод, что большинство заёмщиков работают в сфере управления.

В ходе выполнения работы был произведен сравнительный анализ алгоритмов кластеризации. Все алгоритмы кластеризации оказались полезны и эффективны. Но их нужно применять в зависимости от поставленных целей. Для решения задачи уменьшения объема датасета предпочтителен алгоритм KMeans, т.к. он явно определяет наиболее незначимые кластера. В результате работы алгоритма датасет можно уменьшить путем удаления из него записи, которые относятся к классу «Клиенты с очень высоким балансом» как к самому малозначительному.

Также исследованию подвергся метод корреляции Пирсона для определения степени взаимосвязи между переменными. Из проведенной работы по исследованию трех датасетов и прогнозированию, используя модель выборки максимального подобия по коэффициенту линейной корреляции Пирсона можно сделать выводы, что метод неплохо подходит для работы с временными рядами. Элементарные функции прогнозирования он выполняет, пусть и не очень хорошо. Так же важны размер датасета и свойства данных (сезонность, монотонность).

# Список использованных источников

1. Датасет: Рентабельность собственного капитала для Банка Малайзии https://fred.stlouisfed.org/series/DDEI06MYA156NWDB
2. Датасет: Инфляция потребительских цен в Майлазии - https://fred.stlouisfed.org/series/FPCPITOTLZGMYS.
3. Датасет: Цены на жилую недвижимость в Малайзии -https://fred.stlouisfed.org/series/QMYR628BIS.