РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

Факультет физико-математических и естественных наук

Кафедра прикладной информатики и теории вероятностей

Утверждаю

Заведующий кафедрой   
прикладной информатики   
и теории вероятностей  
д.т.н., профессор  
\_\_\_\_\_\_\_\_\_\_\_\_ К.Е. Самуйлов

« » 20 г.

КУРСОВАЯ РАБОТА

на тему

«Статистический анализ выборок малого объема»

по дисциплине «Компьютерный практикум по математическому моделированию»

шифр и наименование учебной дисциплины

Выполнил

Студент группы НФИбд-01-17

Студенческий билет №:

И.О. Фамилия

« » 20 г.

Руководитель

доцент кафедры   
прикладной информатики и теории вероятностей, к.ф.-м.н.

И.О. Фамилия

**Оглавление**

[Список используемых сокращений 3](#_Toc91110755)

[Введение 4](#_Toc91110756)

[Глава 1 Кластеризация малой выборки 5](#_Toc91110757)

[1.1 Иерархическая кластеризация 6](#_Toc91110758)

[**1.1.1** **Практическая реализация** 7](#_Toc91110759)

[1.2 Кластеризация методом К-средних 11](#_Toc91110760)

[**1.2.1** **Практическая реализация** 12](#_Toc91110761)

[Глава 2 Классификация малой выборки 14](#_Toc91110762)

[2.1 Практическая реализация сравнения классификаторов 14](#_Toc91110763)

[2.2 Практическая реализация классификации новых значений лучшим методом 18](#_Toc91110764)

[Глава 3 Экстраполяция на основе малой выборки 19](#_Toc91110765)

[3.1 Экстраполяция методом наименьших квадратов 20](#_Toc91110766)

[**3.1.1** **Практическая реализация прогнозирования МНК** 21](#_Toc91110767)

[3.2 Экстраполяция методом скользящей средней 24](#_Toc91110768)

[**3.2.1** **Практическая реализация прогнозирования МСС** 24](#_Toc91110769)

[Заключение 26](#_Toc91110770)

# **Список используемых сокращений**

ГС – генеральная совокупность

МНК – метод наименьших квадратов

МСС – метод скользящей средней

PUS – абсцесс печени

PIM – детский мультисистемный воспалительный синдром

PWS – синдром Прадера-Вилли

# **Введение**

Актуальность данной работы обусловлена необходимостью исследования малых выборок в различных сферах человеческой деятельности. Далеко не всегда у исследователя имеется достаточно большой или полный набор данных. Порой получение выборки достаточно большого объема представляет собой трудозатратный, дорогой и долгий процесс, который может себе позволить далеко не каждый исследователь или компания, поэтому единственным подходящим вариантом становится малая выборка данных. По этой причине необходимо изучать методы анализа и исследования выборок малого объема, уметь работать с такими выборками и определять наилучшие алгоритмы для анализа этих выборок. Изучение методов анализа малых выборок, определение лучших алгоритмов и дальнейшее применение полученных знаний позволит исследователям оптимизировать трудовую деятельность и сократить возможные временные и финансовые расходы.

Целью моей курсовой работы является обзор и изучение методов статистического анализа выборок малого объема, их практическая реализация на языке python, а также выбор лучших алгоритмов для решения задач в данной работе. Под выборкой малого объема будем понимать выборку, содержащую не более 200 значений.

Основными задачами мой работы являются:

1. Кластеризация используемой малой выборки
2. Выбор лучшего метода классификации для используемой малой выборки и прогноз классов для новых данных на основе данного метода
3. Экстраполяция на основе данных используемой малой выборки

Методами исследования является кластерный анализ, корреляционный анализ, классификация объекта и экстраполяция. Все эти методы полезны на практике и реализуются с помощью алгоритмов, которые приведены в работе.

Курсовая работа состоит из введения, трех разделов, заключения и списка используемой литературы.

Во введении дается определение малой выборки, формируется проблема статистического анализа малой выборки и ставится задача на дальнейшую работу.

В первом разделе в роли метода анализа выступает кластеризация малой выборки данных о клиентах торговой сети.

Во втором разделе производится определение лучшего алгоритма классификации малой выборки данных о пациентах с COVID-19.

В третьем разделе малая выборка данных о стоимости жилья в России исследуется методом экстраполяции с получением статистически-обоснованного прогноза на следующий год.

В заключении подведены общие итоги курсовой работы, изложены основные выводы.

# 

# **Глава 1 Кластеризация малой выборки**

Кластеризация (или кластерный анализ) — это задача разбиения множества объектов на группы, называемые кластерами. Внутри каждой группы должны оказаться «похожие» объекты, а объекты разных группы должны быть как можно более отличны. Главное отличие кластеризации от классификации состоит в том, что перечень групп четко не задан и определяется в процессе работы алгоритма.

Одним из первых пунктов статистического анализа выборки является кластеризация, также известная как кластерный анализ. В процессе кластеризации решается задача разбиения наблюдений или элементов из выборки на группы, которые называют кластерами. В каждом кластере должны находиться максимальное похожие элементы, а элементы из других групп должны максимально отличаться. Принципиально важным моментом является то, что до начала процедуры кластеризации неизвестно точное количество кластеров, как неизвестна и принадлежность элементов к какому-либо кластеру. Все это определяется в процессе выполнения кластеризации.  
  
 Общий алгоритм кластерного анализа:

1. Получить выборку из ГС
2. Проверить значения переменных (признаков), при необходимости произвести нормализацию.
3. Вычислить значение меры сходства между переменными (признаками)
4. Применить один из методов кластерного анализа для разбиения исходной выборки на кластеры

Теперь, когда определены понятие кластеризации и алгоритм данной процедуры, можно перейти к разбору методов кластеризации и практической реализации.

## **1.1 Иерархическая кластеризация**

Основной идеей иерархической кластеризации является создание дерева вложенных кластеров, которое называют дендрограммой. Данное дерево строится по матрице мер близости и отражает связи между объектами. В основном при использовании иерархической кластеризации изначально каждый элемент выборки рассматривается как отдельный кластер, а в дальнейшем производится объединение кластеров на основе схожести. Такой вид иерархической кластеризации называется агломеративным.

Более подробно данный метод описан в книге Б. Дюрана «Кластерный анализ». Алгоритм иерархической кластеризации выглядит следующим образом:

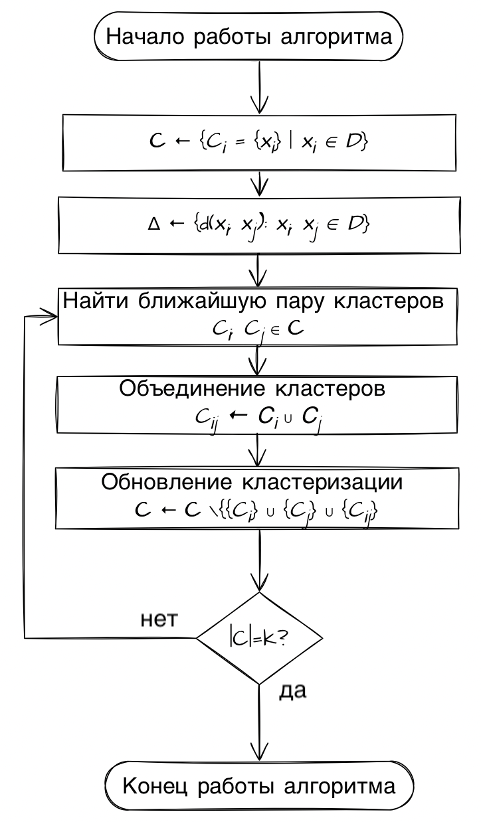


Рисунок 1 Алгоритм иерархической кластеризации

Применение данного метода для выборки малого объема должно быть эффективным. Данные в любом случае разбиваются на единичные кластеры и после этого объединяются, размер выборки не имеет определяющего значения в данном случае.

### **1.1.1 Практическая реализация иерархической кластеризации**

Для начала необходимо определить нашу выборку. Для реализации кластеризации выборки малого объема был выбран Customer Clustering dataset с сайта kaggle.com.

В этом наборе данных содержатся сведения о покупателях торгового центра:

ID, возраст, пол, семейное положение, уровень образования (неизвестно/другое, старшая школа, университет, выпускник школы), годовая зарплата, категория занятости (неквалифицированный сотрудник, квалифицированный сотрудник / должностное лицо, руководство / самозанятый / высококвалифицированный сотрудник / должностное лицо), место жительства (большой, средний, маленький город).

Все значения изначально переведены в числовые для удобства вычислений.

Проведем подготовку рабочей области, подключим необходимые библиотеки и функции и отключим отображение предупреждений:

**import** **numpy** **as** **np**

**import** **matplotlib.pyplot** **as** **plt**

**import** **pandas** **as** **pd**

**import** **seaborn** **as** **sns**

**from** **sklearn.cluster** **import** KMeans

**import** **scipy.cluster.hierarchy** **as** **sch**

**from** **sklearn.preprocessing** **import** StandardScaler

**import** **warnings**

warnings.filterwarnings("ignore")

Далее считаем 25 значений из нашего набора данных, который находится в той же директории, удалим столбец ID (он не несет никакой ценности для анализа), просмотрим наш набор данных:

X = pd.read\_csv("customer\_clust.csv", sep = ',', skipfooter = **1974**)

X = X.drop(["ID"], axis = **1**)

X

Наша выборка выглядит следующим образом:

Изображение выглядит как стол

Автоматически созданное описание

Рисунок 2 Customer Clustering dataset

Построим матрицу корреляции между признаками

sns.heatmap(X.corr(), cmap = "RdBu", annot = True);

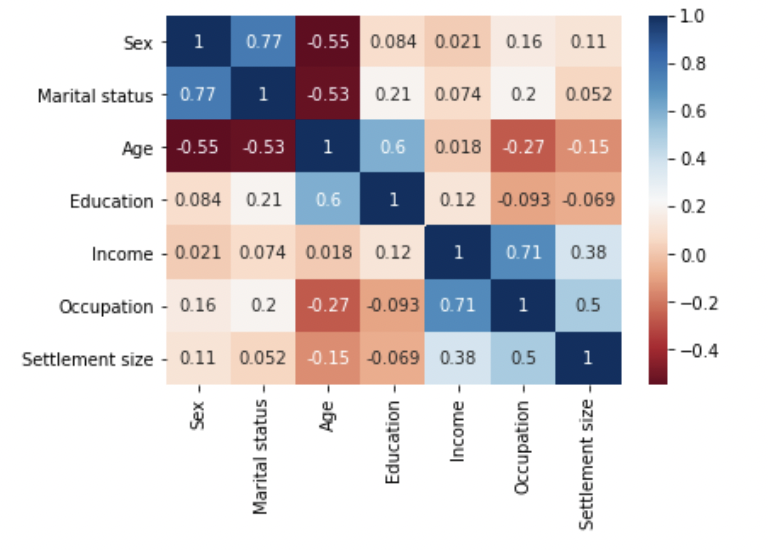


Рисунок 3 Матрица корреляций

Просматривается связь в паре годовой доход-категория занятости. Связь между показателями пола и семейного положения и образования и возраста не считаем полезной в данном случае.

Далее необходимо произвести стандартизацию данных, так как в выборке имеются аномально большие значения ежегодного дохода. После этого производим построение дендрограммы для стандартизированного набора данных с помощью соответствующей функции библиотеки scipy.cluster.hierarchy

scaler = StandardScaler()

X\_std = scaler.fit\_transform(X)

dendogram = sch.dendrogram(sch.linkage(X\_std,method = 'ward'))

plt.title('Dendrogram')

plt.xlabel('Customers')

plt.ylabel('Euclidean Distances')

plt.show()

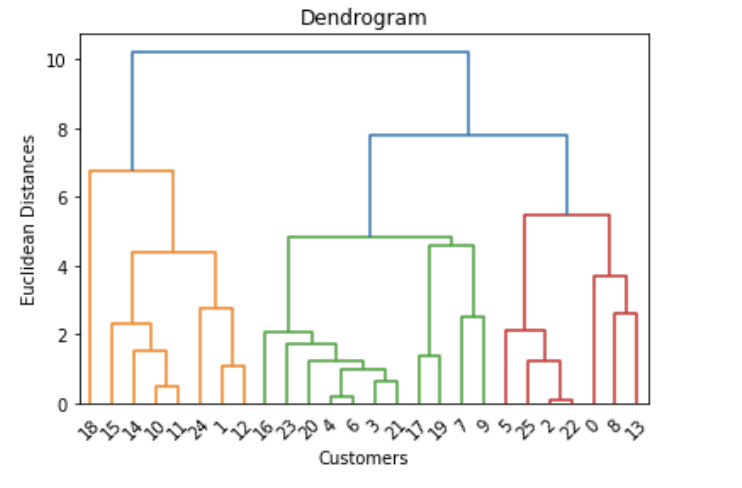


Рисунок 4 Дендрограмма набора данных

В итоге имеем нашу выборку, разбитую на три кластера. Плюсом данного метода кластеризации является то, что число кластеров указывать не нужно, оно определяется в процессе вычислений автоматически.

## **1.2 Кластеризация методом К-средних**

Далее рассмотрим кластеризацию методом k-means. Данный метод относится к группе методов квадратичной ошибки. В данном случае можно говорить, что в процессе кластеризации производится оптимальное разбиение на кластеры. Оптимальность характеризуется как минимизация среднеквадратичной ошибки разбиения:

где cj — «центр масс» кластера j (точка со средними значениями характеристик для данного кластера). Алгоритм k-means в процессе выполнения разбивает выборку на заданное количество кластеров, элементы которых максимально удалены друг от друга. Подробнее алгоритм описан в [книге «Основы статистического анализа» автора Вуколова Э. А.](#_Список_литературы)

Алгоритм метода k-means:

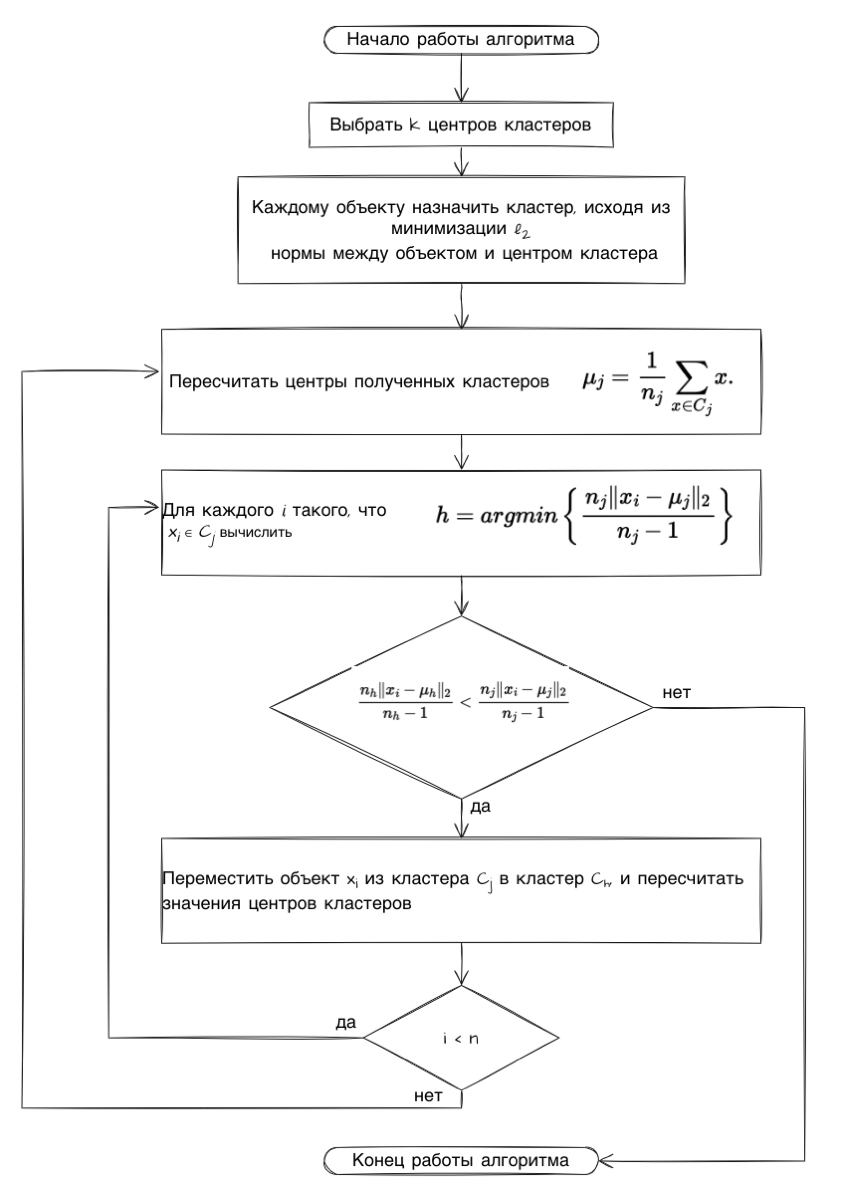


Рисунок 5 Алгоритм k-means

### **1.2.1 Практическая реализация метода К-средних**

Для реализации используем функцию KMeans библиотеки sklearn.cluster для нахождения трех кластеров. Далее добавим результаты разбиения в отдельный столбец нашего набора данных

kmeans=KMeans(n\_clusters=**3**,random\_state=**0**)

kmeans.fit(dataset\_std)

prediction=kmeans.predict(X\_std)

prediction

X['Cluster'] = prediction

X

Итоговый набор данных выглядит так:

Изображение выглядит как стол

Автоматически созданное описание

Рисунок 6 Набор данных после кластеризации

Кластеризация проведена. Проверим ранее отмеченную нами зависимость из матрицы корреляции

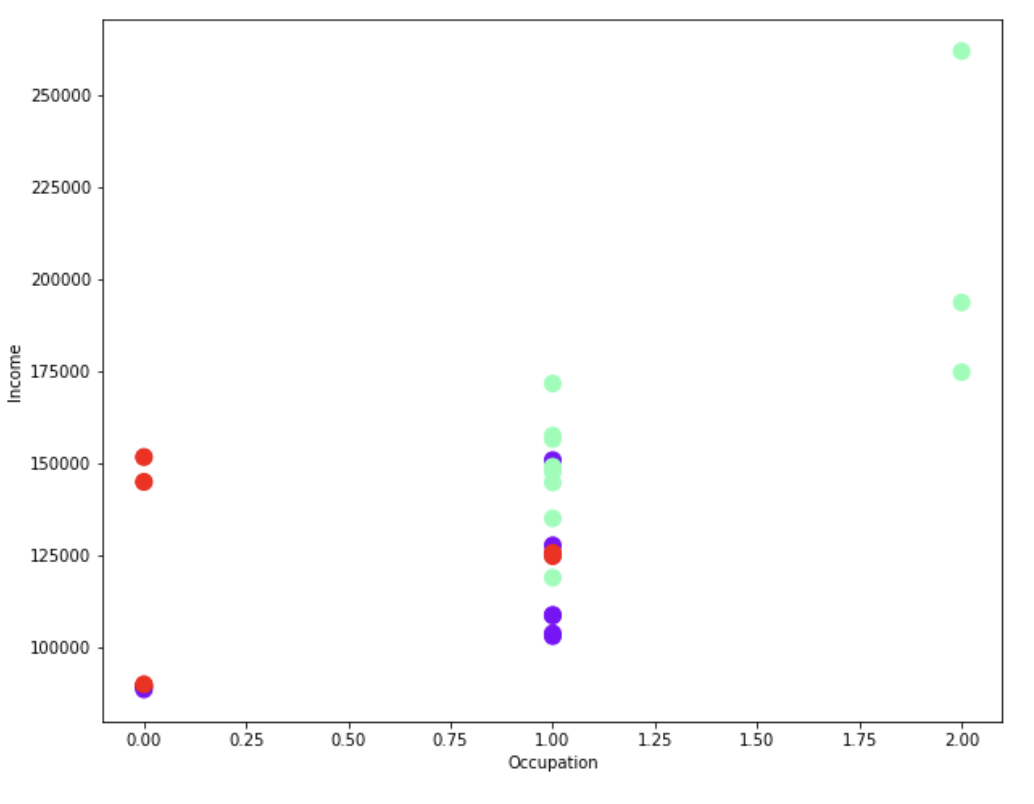


Рисунок 7 Отношение среднего годового дохода к должности

Кластер 1 (светло-зеленый цвет) собрал в себе клиентов с лучшим уровнем категории занятости и высшим средним ежегодным доходом. Кластер 0 (фиолетовый цвет) – «золотая середина», в кластере 2 (красный цвет) находятся клиенты с худшей категорией занятости и меньшим ежегодным доходом. Кластеризация проведена успешно, результаты объяснимы.

# **Глава 2 Классификация малой выборки**

Классификация – метод анализа данных, задачей которого является определение принадлежности различных элементов выборки к классам.

В общем виде реализация классификации заключается в получении выборки, в которой имеется конечное число объектов с известным классом и некоторое число объектов, классовая принадлежность которых неизвестна. Формируется обучающая выборка, которая состоит только из объектов с известным классом. Далее необходимо разработать или выбрать алгоритм, который позволит классифицировать, то есть предсказать номер или название класса, объект с неизвестной классовой принадлежностью.

Для выборок малого объема задача классификации также актуальна. Не всегда имеется большое количество наблюдений и данных, а также не всегда имеется возможность или время собрать или получить достаточно большой набор данных для исследования. Но эти проблемы не означают, что классификация невозможна или не нужна.

Алгоритмы классификации основаны на различных методах и измерениях, поэтому использование различных алгоритмов при анализе одной и той же выборки даст не одинаковый результат и точность. Нам необходимо выяснить, какой из алгоритмов классификации даст лучший результат для имеющейся у нас малой выборки данных.

## **2.1 Практическая реализация сравнения классификаторов**

На данном этапе работы мы будем использовать COVID-19 Surveillance Data Set, который был загружен с известного репозитория данных для машинного обучения UCI Machine Learning Repository. Размер набора данных 14х8, что подходит под наше определение малой выборки.

Этот набор данных представляет собой наблюдения за развитием болезни COVID-19 и возможными последствиями переноса болезни у 13 пациентов. Имеется семь симптомов болезни вида А01-А07 и три класса последствий болезни: PUS, PIM, PWS.

* PUS – абсцесс печени
* PIM – детский мультисистемный воспалительный синдром
* PWS – синдром Прадера-Вилли

Требуется обучить классификатор на основе имеющихся данных для возможности дальнейшей классификации возможных последствий у новых пациентов.

Но для начала необходимо будет выбрать лучший классификатор для нашей задачи. Проведем сравнение точности классификации следующих алгоритмов:

|  |  |  |  |
| --- | --- | --- | --- |
| № | Название алгоритма | Перевод | Документация python |
| 1 | LogisticRegression | Логистическая регрессия | [LogisticRegression](https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html) |
| 2 | SVC | Метод векторов поддержки | [SVC](https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html) |
| 3 | KNeighbours | Метод к-ближайших соседей | [KNeighbours](https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html) |
| 4 | DecisionTree | Дерево принятия решений | [DecisionTree](https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html) |
| 5 | RandomForest | Случайный лес | [RandomForest](https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html) |
| 6 | GaussianNB | Наивный байесовский классификатор | [GaussianNB](https://scikit-learn.org/stable/modules/generated/sklearn.naive_bayes.GaussianNB.html) |

Таблица 1 Алгоритмы классификации и источники информации

Источники информации для всех алгоритмов находятся в списке используемой литературы.

* Для начала подключим необходимые библиотеки и импортируем нужные функции
* Также отключим отображение предупреждений:

**from** **sklearn.linear\_model** **import** LogisticRegression

**from** **sklearn.svm** **import** SVC

**from** **sklearn.neighbors** **import** KNeighborsClassifier

**from** **sklearn.tree** **import** DecisionTreeClassifier

**from** **sklearn.ensemble** **import** RandomForestClassifier

**from** **sklearn.naive\_bayes** **import** GaussianNB

**from** **sklearn.metrics** **import** classification\_report,confusion\_matrix

**from** **sklearn.model\_selection** **import** train\_test\_split

**import** **numpy** **as** **np**

**import** **pandas** **as** **pd**

**import** **warnings**

warnings.filterwarnings("ignore")

* Далее загрузим наш набор данных, заменим символы «+» на 1, «-» на 0 для признаков и «PUS» на 0, «PIM» на 1 и «PWS» на 2 для классов:

data = pd.read\_csv('covid\_class.csv')

data = data.replace({'+': **1**, '-': **0**})

data["Categories"] = data["Categories"].replace({'PUS' : **0**, 'PIM': **1**, 'PWS': **2**})

data

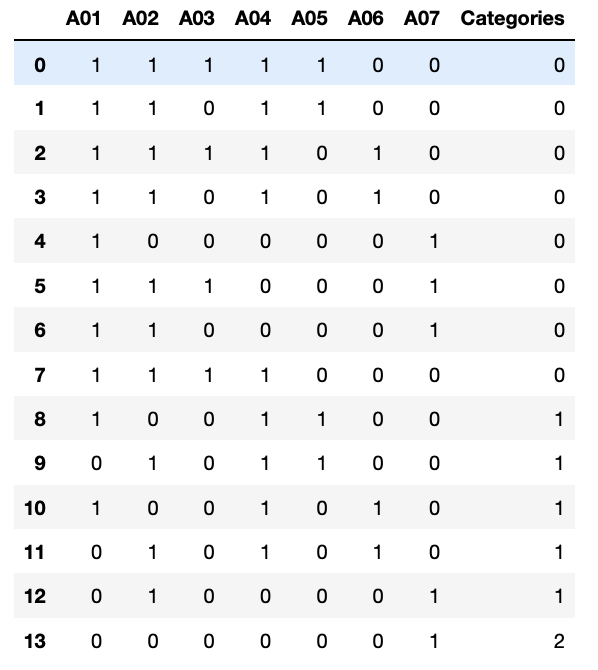


Рисунок 8 COVID-19 Surveillance Data Set

data1 = data.drop(["Categories"], axis = **1**)

target1 = data["Categories"]

* Наш набор данных содержит только одно наблюдение с классом 2, поэтому будет лучше, если оно точно попадет в обучающую выборку, для этого сначала удалим его, а потом добавим вручную.
* Выделим столбец классов в отдельный набор данных, из исходного набора этот столбец удалим.
* Создадим словарь для записи показателей точности классификации:

data = data.drop([**13**], axis = **0**)

target = data["Categories"]

data = data.drop(["Categories"], axis = **1**)

values = {}

* Инициализируем классификаторы.
* Далее проводим 1000 проверок точности классификации нашими алгоритмами
* На каждой итерации производим разделение набора данных на обучающую и тестовую выборку в соотношении 0,15. Также вручную добавляем в обучающую выборку элемент с классом 2, которое удаляли ранее.
* На каждой итерации производится обучение каждого классификатора и предсказание классов для тестовой выборки. Результаты точности классификации записываются в словарь:

log\_clf = LogisticRegression()

svc\_clf = SVC()

knn\_clf = KNeighborsClassifier()

dt\_clf = DecisionTreeClassifier()

rf\_clf = RandomForestClassifier()

nbc\_clf = GaussianNB()

**for** i **in** range(**1**,**1000**):

X\_train, X\_test, y\_train, y\_test = train\_test\_split(data, target, test\_size=**0.15**)

X\_train.loc[**13**] = [**0**, **0**, **0**, **0**, **0**, **0**, **1**]

y\_train.loc[**13**] = **2**

**if** i == **1**:

**for** clf **in** [log\_clf, svc\_clf, knn\_clf, dt\_clf, rf\_clf, nbc\_clf]:

clf.fit(X\_train, y\_train)

pred = clf.predict(X\_test)

values[clf] = accuracy\_score(y\_test,pred)

**else**:

**for** clf **in** [log\_clf, svc\_clf, knn\_clf, dt\_clf, rf\_clf, nbc\_clf]:

clf.fit(X\_train, y\_train)

pred = clf.predict(X\_test)

values[clf] += accuracy\_score(y\_test,pred)

* Найдем среднюю точность:

**for** i **in** values:

values[i] = values[i] / **1000**

values

Имеем следующие результаты:

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 9 Сравнение точности классификации разными алгоритмами

Лучший результат показан алгоритмом DecisionTree, для дальнейших операций будем использовать его.

## **2.2 Практическая реализация классификации новых значений лучшим методом**

Далее перейдем к задаче классификации нового значения. Проверим, сможет ли классификатор предсказать классы для наблюдений, аналогичных имеющимся и для одного нового.

* Обучим классификатор на всем наборе данных
* Создадим четыре переменные и построим прогноз класса:

dt\_clf.fit(data1,target1)

test1 = np.array([**1**, **1**, **1**, **1**, **0**, **0**, **0**]) # test for 0 class

test2 = np.array([**0**, **1**, **0**, **1**, **0**, **1**, **0**]) # test for 1 class

test3 = np.array([**0**, **0**, **0**, **0**, **0**, **0**, **1**]) # test for 2 class

test4 = np.array([**1**, **1**, **1**, **0**, **0**, **0**, **0**]) # test for new value

y1 = dt\_clf.predict(test1.reshape(**1**, -**1**))[**0**]

y2 = dt\_clf.predict(test2.reshape(**1**, -**1**))[**0**]

y3 = dt\_clf.predict(test3.reshape(**1**, -**1**))[**0**]

y4 = dt\_clf.predict(test4.reshape(**1**, -**1**))[**0**]

**print**('Предсказанные классы: **\n**',y1,'для точки нулевого класса**\n**',

y2, 'для точки первого класса **\n**',y3,'для точки второго класса**\n**',

y4, 'для случайной новой точки')

Имеем прогноз:

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 10 Прогноз методом DecisionTree для 4 разных наблюдений

Как видим, наш классификатор справился с задачей классификации наблюдений, дублирующих наблюдение каждого класса из нашего набора данных. Также он классифицировал новое наблюдение как 0 класс.

Подводя итоги, можно сказать, что задача классификации малой выборки выполнена. Мы определили наилучший классификатор для имеющейся выборки малого объема COVID-19 Surveillance Data Set. Точность классификации алгоритмом DecisionTree оказалась значительно выше, чем точность классификации всеми остальными, поэтому можно сделать вывод, что данный алгоритм наилучшим образом подходит для классификации данной малой выборки.

# **Глава 3 Экстраполяция на основе малой выборки**

Экстраполяция – это особый вид аппроксимации, при котором функция аппроксимируется не между заданными значениями, а вне заданного интервала.

Данный метод основывается на анализе тенденций (прошлых и настоящих), выводе, основанном на этом анализе и переносе вывода на весь набор данных (наблюдений). Экстраполяция относится к так называемым «прогнозирующим» методам анализа данных.

В данной работе мы рассмотрим два статистических метода экстраполяции: метод наименьших квадратов и метод скользящей средней.

Разработка прогноза – важная задача для анализа любой выборки. Возможно, с нашими малыми выборками появятся проблемы, ведь данных для качественного прогноза не так много. В любо случае мы построим прогноз двумя методами и определим лучший из них.

На данном этапе работы будем использовать данные о средней рыночной стоимости 1 квадратного метра квартиры на вторичном рынке в России. Данные для анализа взяты с сайта росстата.

Данную задачу считаю актуальной, особенно на фоне последних изменений цен на недвижимость. Будет полезно рассмотреть методы прогнозирования будущей цены на основе малого набора имеющихся данных, а именно с 2004 года, когда цена за 1 квадратный метр стала превышать 15000 рублей. Более ранние показатели сочтем слабо-информативными и уберем их из нашего набора данных. Предсказывать будем стоимость для 2020 года по причине того, что в дальнейшем сможем вычислить точность предсказания на основе уже имеющегося значения.

## **3.1 Экстраполяция методом наименьших квадратов**

Одним из наиболее распространенных методов статистического оценивания и прогнозирования является метод наименьших квадратов (далее МНК). Суть метода заключается в нахождении таких оптимальных параметров линейной регрессии, что сумма квадратов ошибок минимальна. МНК минимизирует евклидово расстояние между вектором предсказанных значений анализируемой переменной и фактическими значениями данной переменной:

Аппроксимировать наши данные будем линейной зависимостью *y = ax + b.* Необходимо найти параметры *a* и *b.* Преобразуем основную формулу для данной задачи:

Необходимо найти такие значения *a* и *b*, при которых значение функции ***F*** будет наименьшим. Найдем частные производные нашей функции по двум переменным и приравняем их к нулю:

Решим данную систему методом подстановки и получим:

Для коэффициентов *a* и *b,* найденных по данным формулам, функция ошибки будет принимать наименьшее значение. Более подробное описание метода приведено в [книге Сухорученкова Б. И. «Анализ малой выборки. Прикладные статистические методы»](#_Список_литературы)

### **3.1.1 Практическая реализация прогнозирования МНК**

* Для начала подключим необходимые библиотеки и загрузим набор данных:

**import** **numpy** **as** **np**

**import** **pandas** **as** **pd**

data = pd.read\_csv('appartment\_ave\_prices.csv', sep = ';')

data

Наш набор данных выглядит следующим образом:

Изображение выглядит как стол

Автоматически созданное описание

Рисунок 11 Набор данных о стоимости квадратного метра жилья в России

* Сохраним тестовое значение
* Удалим данные до 2004 года, а также данные за 2020 год
* Создадим два numpy-массива для хранения значений года(X) и цены за квадратный метр для всех типов квартир(Y)
* Чтобы избежать проблем с индексами заново индексируем наши массивы, а также отразим значение года на интервал [1;16]:

test\_arr = data["Все типы квартир"]

test\_value = test\_arr[**22**]

data = data.drop([**0**, **1**, **2**, **3**, **4**, **5**, **22**], axis = **0**)

X = data["Год"]

X.index = np.arange(**1**, len(X) + **1**)

**for** i **in** range(**1**,**17**):

X = X.replace({X[i]: i})

Y = data["Все типы квартир"]

Y.index = np.arange(**1**, len(Y) + **1**)

Имеем следующее:

Изображение выглядит как стол

Автоматически созданное описаниеИзображение выглядит как стол

Автоматически созданное описание

Рисунок 12 Преобразованная переменная времени и переменная стоимости

* Далее зададим n и найдем необходимые для вычисления коэффициентов произведения
* Вычислим коэффициенты и сделаем прогноз на 2020 год с вычислением погрешности предсказания:

**if** X.size == Y.size:

n = X.size

Yf\_X = Y \* X

X2 = X \*\* **2**

a = (sum(Yf\_X) - (sum(X) \* sum(Y)) / n) / (sum(X2) - (sum(X)\*\***2**)/n)

b = sum(Y) / n - a \*sum(X) / n

predict = a \* (n+**1**) + b

error\_mnk = (**1** - predict / test\_value)\***100**

**print**('Прогноз на 2020 год по МНК: ',predict)

**print**('Фактическое значение за 2020 год: ',test\_value)

**print**('Погрешность: ',error\_mnk, '%')

Вывод:

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 13 МНК прогноз

После выполнения алгоритма получили достаточно точное предсказание, что можно считать хорошим результатом в условиях анализа настолько малой выборки.

## **3.2 Экстраполяция методом скользящей средней**

Метод скользящей средней (далее МСС) представляет собой функцию, значение которой в каждой новой точке равны некоторому усредненному значению в предыдущих точках.

Обычно используется для сглаживания временных рядов, но мы попробуем использовать его для формирования прогноза. Сглаживание скользящей средней основано на взаимно погашающихся отклонениях в случайных величинах. Рабочая формула:

n – количество наблюдений, m – значение скользящей средней для n наблюдений, y – значение переменной. Более подробно метод описан в книге «Математические методы построения прогнозов» Грешилова А. А.

### **3.2.1 Практическая реализация прогнозирования МСС**

* Напишем функцию, которая будет вычислять скользящую среднюю для наших данных.
* Количество наблюдений примем равным 3.
* Далее создадим список, содержащий эти значения. Первый и последний его элемент равен 0:

**def** **s**(x, w, z):

med = (Y[x] + Y[w] + Y[z])/**3**

**return** med

ave\_values = [**0**]

**for** i **in** range(**2**,**16**):

ave\_values.append(s(i-**1**, i, i+**1**))

ave\_values.append(**0**)

ave\_values

Наш список средних значений:

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 14 Массив значений скользящей средней

* Строим прогноз и анализируем результаты:

pred = ave\_values[**14**] + **1**/**3** \* (Y[**16**] - Y[**15**])

error\_mss = (**1** - pred / test\_value)\***100**

**print**('Прогноз на 2020 год по МСС:', pred)

**print**('Фактическое значение за 2020 год: ',test\_value)

**print**('Погрешность: ', error\_mss, '%')

**print**('МНК оказался эффективнее в ', error\_mss/error\_mnk, 'раз')

Вывод:

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 15 Прогноз МСС и сравнение результатов

В заключении нужно сказать, что для наших целей МНК оказался более полезным, чем МСС. Точность прогноза МСС нельзя назвать хотя бы удовлетворительной. Возможно, причиной плохого прогнозирования является большой разброс значений, в принципе большие значения и резкие изменения значений. В любом случае МНК для нашей выборки наиболее предпочтителен.

Важно отметить, что даже для такой малой выборки реально составить вполне точный прогноз, поэтому задачу экстраполяции данных малой выборки считаю полезной и выполненной.

# **Заключение**

# **Список литературы**

Книги и учебники:

Сухорученков Б. И. Анализ малой выборки. Прикладные статистические методы. – М.: Вузовская книга, 2010

Шорохова И. С., Кисляк Н. В., Мариев О. С. Статистические методы анализа. – Екатеринбург: Изд-во Урал. ун-та, 2015

Peter Bruce, Andrew Bruce, Peter Gedeck. Practical statistics for Data Scientists. Second edition. – USA: O’Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472, 2020

Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani. An introduction to statistical learning with applications in R. Second edition. – 2021 – <https://hastie.su.domains/ISLR2/ISLRv2_website.pdf>

Гаскаров Д. В., Шаповалов В. И. Малая выборка. – М: Статистика, 1978

Кендалл М., Стьюарт А. Теория распределений. М.: Наука, 1966

Вуколов Э. А. Основы статистического анализа. – М.: ФОРУМ, 2008

Шитиков В. К., Мастицкий С. Э. Классификация, регрессия и другие алгоритмы Data Mining с использованием R. – Тольятти, Лондон, 2017

Черезов Д. С., Тюкачев Н. А. Обзор основных методов классификации и кластеризации данных. Воронеж: Вестник ВГУ, серия: системный анализ и информационные технологии, 2009

Грешилов А. А., Стакун В. А., Стакун А. А. Математические методы построения прогнозов. – М: Радио и связь, 1997

Дюран Б., Оделл П. Кластерный анализ. М.: Статистика, 1977

Громова Н. М., Громова Н. И. Основы экономического прогнозирования. – М:, Академия Естествознания, 2007

Python-документация:

<https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html>

<https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html>

<https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html>

<https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html>

<https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html>

<https://scikit-learn.org/stable/modules/generated/sklearn.naive_bayes.GaussianNB.html>

Статьи:

<https://www.analyticsvidhya.com/blog/2019/08/comprehensive-guide-k-means-clustering/>

Интернет-энциклопедии:

<https://ru.wikipedia.org/wiki/Заглавная_страница>

<http://www.machinelearning.ru/wiki/index.php?title=Заглавная_страница>

Источник набора данных, используемых в первой главе – <https://www.kaggle.com/dev0914sharma/customer-clustering?select=segmentation+data.csv>

Источник набора данных, используемых во второй главе – <http://archive.ics.uci.edu/ml/datasets/COVID-19+Surveillance>

Источник набора данных, используемых в третьей главе – <https://rosstat.gov.ru/storage/mediabank/tab_sred_cen_s_1998.htm>