МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ государственное БЮДЖЕТНОЕ

образовательное учреждение

высшего образования

«НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Кафедра вычислительной техники

**РАСЧЁТНО-ГРАФИЧЕСКАЯ РАБОТА**  
«Метод опорных векторов»  
по дисциплине «Методы анализа данных»

**Вариант 31**

Преподаватель: Пешков А.В.

Факультет: АВТФ

Группа: АВТ-918

Студенты: Ванин К.Е.

Новосибирск

2022

# Теоретическая часть

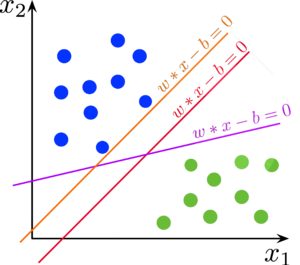
# Общие сведения

Метод опорных векторов (англ. support vector machine, SVM) — один из наиболее популярных методов, который применяется для решения задач классификации и регрессии. Основная идея метода заключается в построении гиперплоскости, разделяющей объекты выборки оптимальным способом. Алгоритм работает в предположении, что чем больше расстояние (зазор) между разделяющей гиперплоскостью и объектами разделяемых классов, тем меньше будет средняя ошибка классификатора.

* 1. **Описание алгоритма**

Рассмотрим задачу бинарной классификации, в которой объектам из  соответствует один из двух классов . Пусть задана обучающая выборка пар "объект-ответ": . Необходимо построить алгоритм классификации .

### **Разделяющая гиперплоскость**

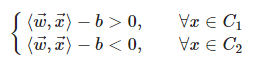


*Рис.1 Примеры разделяющих гиперплоскостей*

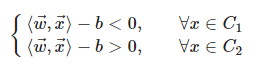
В пространстве  уравнение  при заданных  и определяет гиперплоскость — множество векторов  , принадлежащих пространству меньшей размерности

Гиперплоскость делит   на два полупространства:  и

Говорят, что гиперплоскость разделяет два класса C1 и C2, если объекты этих классов лежат по разные стороны от гиперплоскости, то есть выполнено либо



либо



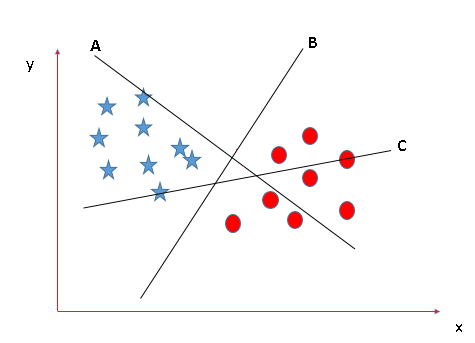
В этом алгоритме каждый элемент данных изображается как точка в n-мерном пространстве (где n — количество признаков), при этом значение каждого признака является значением конкретной координаты. Затем выполняется классификация путем нахождения гиперплоскости, которая лучше всего различает два класса.

В дополнение к выполнению линейной классификации SVM могут эффективно выполнять нелинейную классификацию, неявно отображая свои входные данные в многомерные пространства признаков.

SVM — это дискриминативный классификатор, формально определяемый разделяющей гиперплоскостью. Другими словами, учитывая помеченные обучающие данные (обучение с учителем), алгоритм выводит оптимальную гиперплоскость, которая классифицирует новые примеры.

Самый важный вопрос, возникающий при использовании SVM, заключается в том, как определить правильную гиперплоскость. Рассмотрим следующие сценарии:

* В этом сценарии есть три гиперплоскости, называемые A, B, C. Теперь проблема состоит в том, чтобы определить правильную гиперплоскость, которая лучше всего различает звезды и круги.

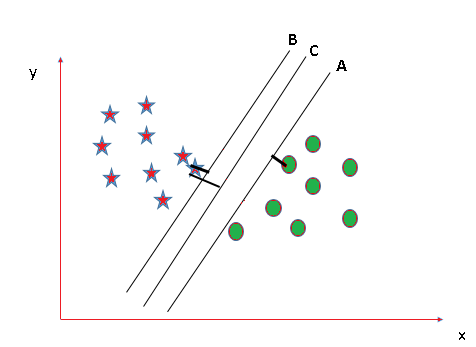


*Рис.2 Гиперплоскости без однозначного разделения*

Эмпирическое правило, которое необходимо знать, прежде чем искать правильную гиперплоскость для классификации звезды и круга: следует выбрать гиперплоскость, которая лучше разделяет два класса.

В этом случае B лучше классифицирует звезду и круг, следовательно, это правая гиперплоскость.

* Теперь возьмем другой сценарий, в котором все три плоскости хорошо разделяют классы. Теперь возникает вопрос, как определить нужную плоскость в этой ситуации.



*Рис.3 Гиперплоскости с однозначным разделением*

В таких сценариях вычислите margin, который представляет собой расстояние между ближайшей точкой данных и гиперплоскостью. Плоскость с максимальным расстоянием будет считаться правильной гиперплоскостью для лучшей классификации классов.

Здесь C имеет максимальный margin и, следовательно, будет рассматриваться как правая гиперплоскость.

**1.3 Реализация в R**

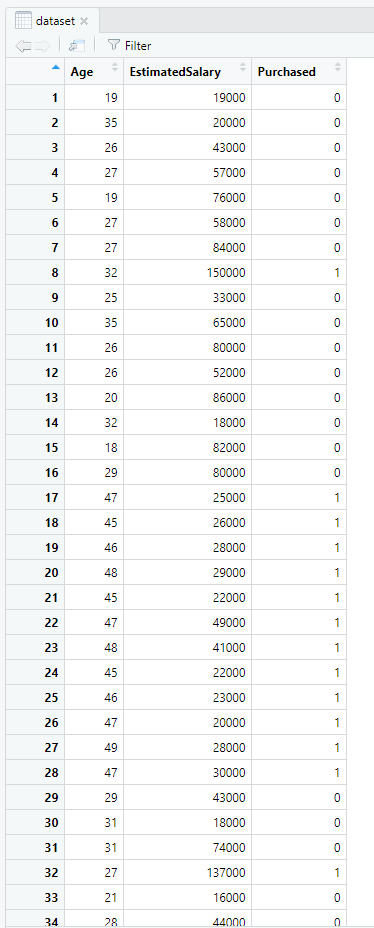
Для реализации будем использовать пакет e1071. В нём функции для анализа скрытых классов, кратковременное преобразование Фурье, нечеткая кластеризация, машины опорных векторов, вычисление кратчайшего пути, групповая кластеризация, и многие другие полезные статистические функции.

Нужные нам имеют окончание .svm

**Практическая часть**

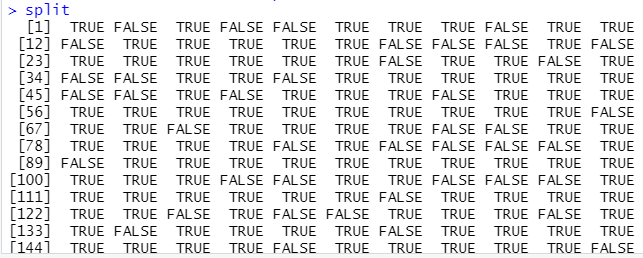
Решим задачу классификации на данных о возрасте и зарплате и покупках американцев, попытаемся предсказать на данных о возрасте и историях покупок, какая у человека будет зарплата.

В наборе данных оставим только нужные столбцы, и как фактор принятия решения выберем данные о покупках.



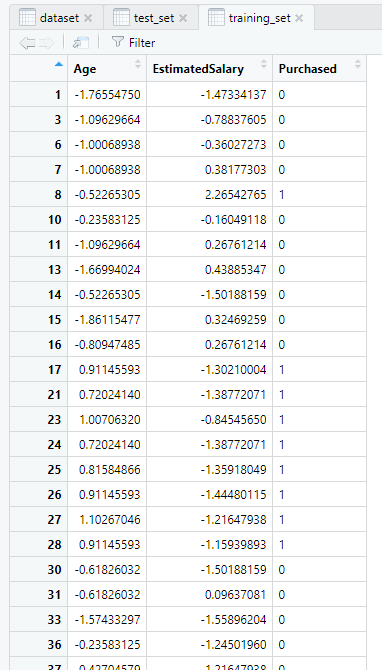
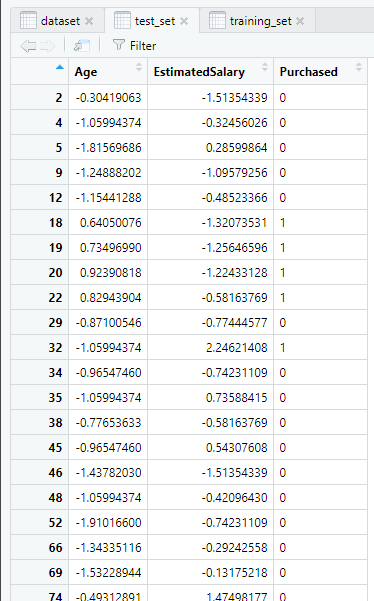
*Рис.4 Выбраны столбцы 3-5*

Разделим данные на тренировочный и тестовый датасеты (алгоритм предполагает обучение).



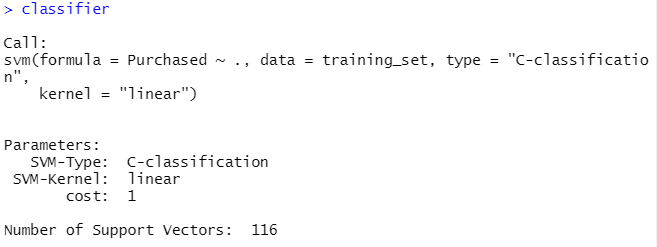
*Рис.5 Разделённый датасет*

Скейлируем датасеты по фактору принятия решения.

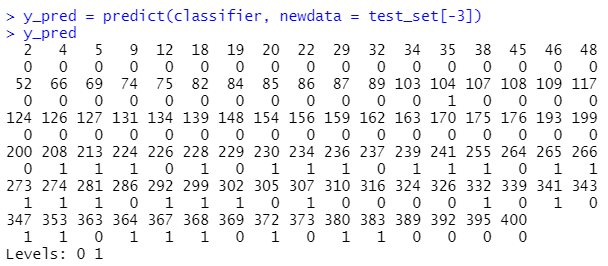
*Рис.6 Скейлированные датасеты*

Создадим функцию-классификатор (код в приложении).



*Рис.7 Детали классификатора*

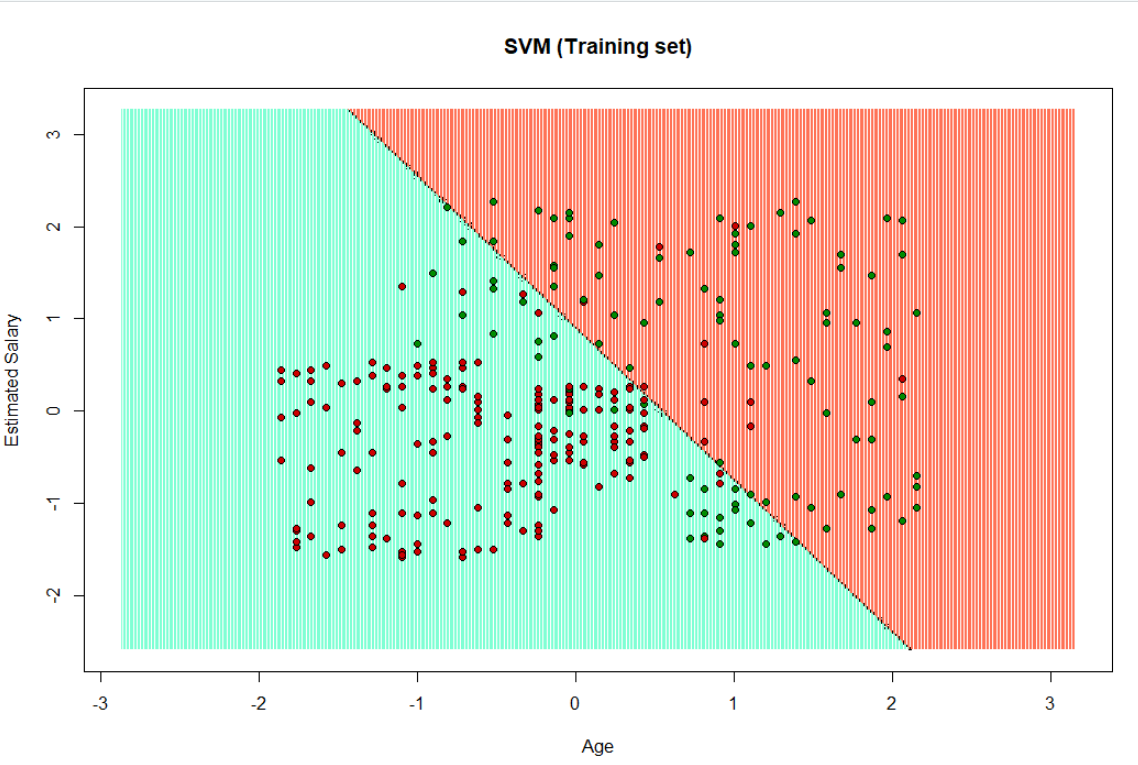
Создаём предиктор – функцию предсказания результата.



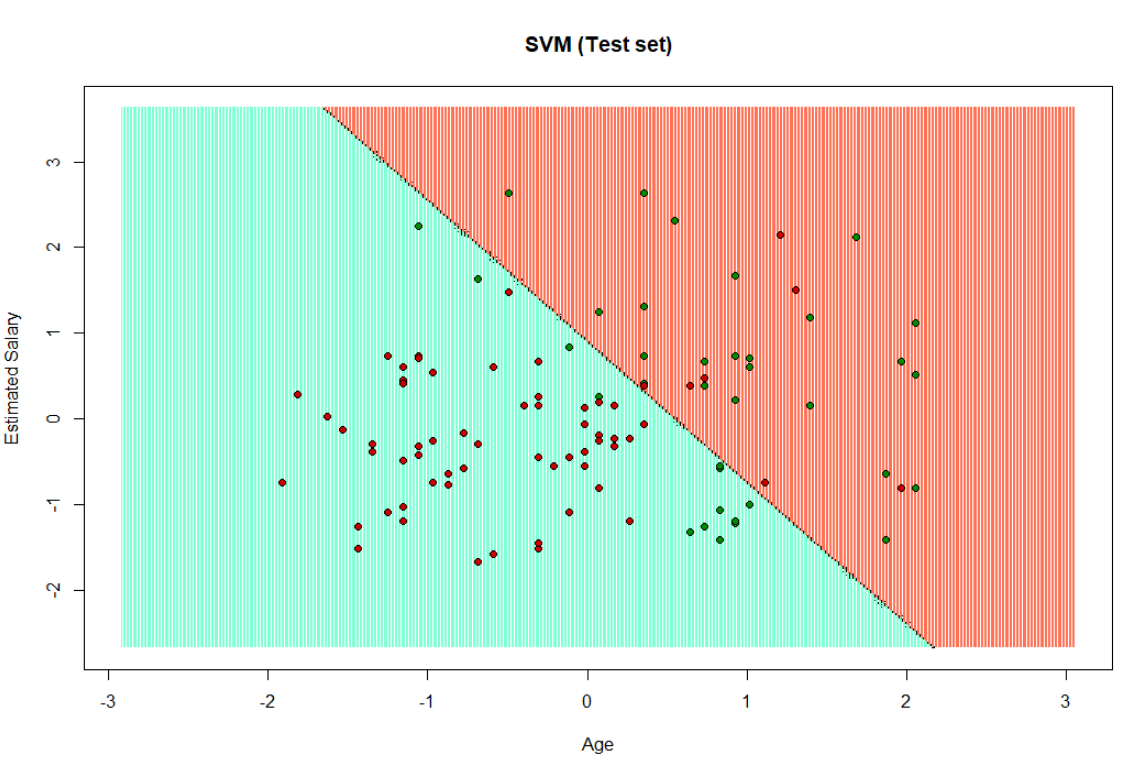
*Рис.8 Функция предсказания результата*

Далее создаётся матрица, чтобы «запутать» тренировочный набор: для более качественного обучения.

Визуализируем результаты обучения и тестирования. Проверим, совпадают ли гиперплоскости в тренировочном и тестовом варианте.



*Рис.9 Разбиваем и визуализируем тренировочный набор*



*Рис.10 Разбиваем и визуализируем тестовый набор*

1. **Выводы**

Поскольку в результате гиперплоскость была найдена в результате тренировочного набора и подтверждена как лучшая в результате тестового набора. Следовательно, SVM был успешно реализован в R.

1. **Список использованных источников**
2. Classifying data using Support Vector Machines(SVMs) in R. — Текст : электронный // geeksforgeeks : [сайт]. — URL: https://www.geeksforgeeks.org/classifying-data-using-support-vector-machinessvms-in-r/ (дата обращения: 25.12.2022).Sam, Wilson The MICE Algorithm / Wilson Sam. — Текст : электронный // R-project : [сайт]. — URL: https://cran.r-project.org/web/packages/miceRanger/vignettes/miceAlgorithm.html (дата обращения: 15.12.2022).
3. Демешев, Б. Заметки по R: Метод опорных векторов (Support Vector Machine) / Б. Демешев. — Текст : электронный // bdemeshev.github.io : [сайт]. — URL: https://bdemeshev.github.io/r\_cycle/cycle\_files/11\_svm.html (дата обращения: 25.12.2022).
4. Метод опорных векторов (SVM). — Текст : электронный // neerc.ifmo : [сайт]. — URL: https://neerc.ifmo.ru/wiki/index.php?title=Метод\_опорных\_векторов\_(SVM)https://neerc.ifmo.ru/wiki/index.php?title=Метод\_опорных\_векторов\_(SVM) (дата обращения: 25.12.2022).
5. e1071. — Текст : электронный // rdocumentation.org : [сайт]. — URL: https://www.rdocumentation.org/packages/e1071/versions/1.7-3 (дата обращения: 25.12.2022).

# Приложение

**Листинг**

*> dataset = read.csv('social.csv')*

*> dataset = dataset[3:5]*

*> View(dataset)*

*> dataset$Purchased = factor(dataset$Purchased, levels = c(0, 1))*

*> set.seed(123)*

*> split = sample.split(dataset$Purchased, SplitRatio = 0.75)*

*> training\_set = subset(dataset, split == TRUE)*

*> test\_set = subset(dataset, split == FALSE)*

*> training\_set[-3] = scale(training\_set[-3])*

*> test\_set[-3] = scale(test\_set[-3])*

*> classifier = svm(formula = Purchased ~ .,*

*+ data = training\_set,*

*+ type = 'C-classification',*

*+ kernel = 'linear')*

*> y\_pred = predict(classifier, newdata = test\_set[-3])*

*> set = training\_set*

*> X1 = seq(min(set[, 1]) - 1, max(set[, 1]) + 1, by = 0.01)*

*> X2 = seq(min(set[, 2]) - 1, max(set[, 2]) + 1, by = 0.01)*

*> grid\_set = expand.grid(X1, X2)*

*> colnames(grid\_set) = c('Age', 'EstimatedSalary')*

*> y\_grid = predict(classifier, newdata = grid\_set)*

*> plot(set[, -3],*

*+ main = 'SVM (Training set)',*

*+ xlab = 'Age', ylab = 'Estimated Salary',*

*+ xlim = range(X1), ylim = range(X2))*

*> contour(X1, X2, matrix(as.numeric(y\_grid), length(X1), length(X2)), add = TRUE)*

*> points(grid\_set, pch = '.', col = ifelse(y\_grid == 1, 'coral1', 'aquamarine'))*

*> points(set, pch = 21, bg = ifelse(set[, 3] == 1, 'green4', 'red3'))*

*> set = test\_set*

*> X1 = seq(min(set[, 1]) - 1, max(set[, 1]) + 1, by = 0.01)*

*> X2 = seq(min(set[, 2]) - 1, max(set[, 2]) + 1, by = 0.01)*

*> grid\_set = expand.grid(X1, X2)*

*> colnames(grid\_set) = c('Age', 'EstimatedSalary')*

*> y\_grid = predict(classifier, newdata = grid\_set)*

*> plot(set[, -3], main = 'SVM (Test set)',*

*+ xlab = 'Age', ylab = 'Estimated Salary',*

*+ xlim = range(X1), ylim = range(X2))*

*> contour(X1, X2, matrix(as.numeric(y\_grid), length(X1), length(X2)), add = TRUE)*

*> points(grid\_set, pch = '.', col = ifelse(y\_grid == 1, 'coral1', 'aquamarine'))*

*> points(set, pch = 21, bg = ifelse(set[, 3] == 1, 'green4', 'red3'))*