- **Постановка задачи:** Произвести оценку частоты покупки каждого из 5ти товаров, чьи статистические данные были представлены, и оценить частоту покупки пары товаров и предложить наиболее рациональный вариант из размещения для повышения спроса на них.
- **Метод решения:** Обработка представленных статистических данных встроенными методами Excel

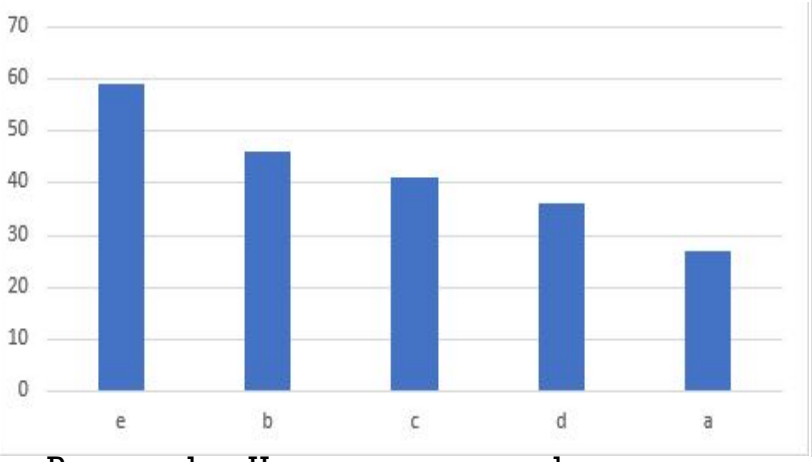

Рисунок 1 — Частота появления 1го товара

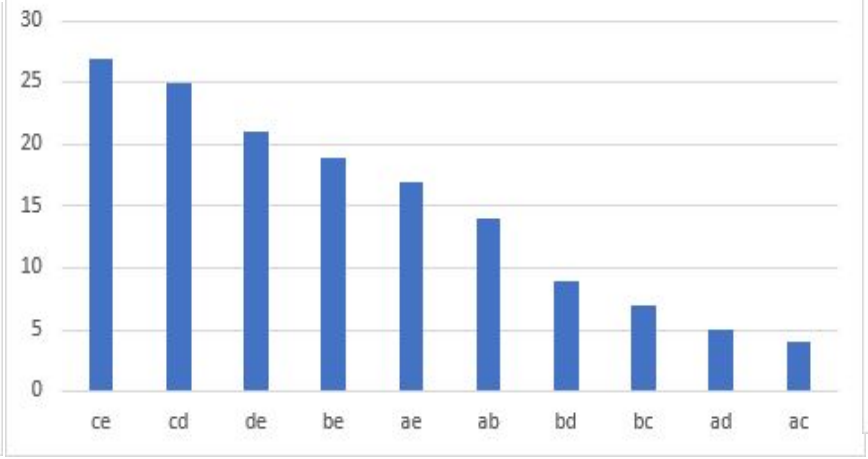

Рисунок 2 — Частота покупки двух товаров одновременно

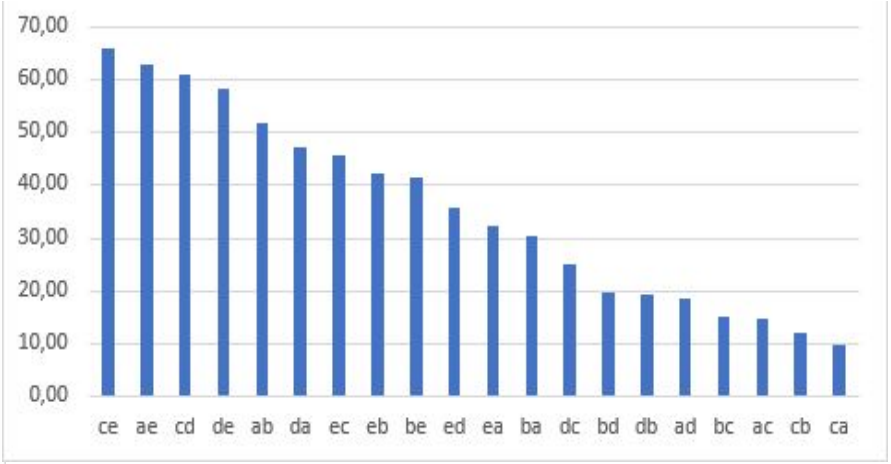

Рисунок 3 — Частота покупки второго товара при покупке первого

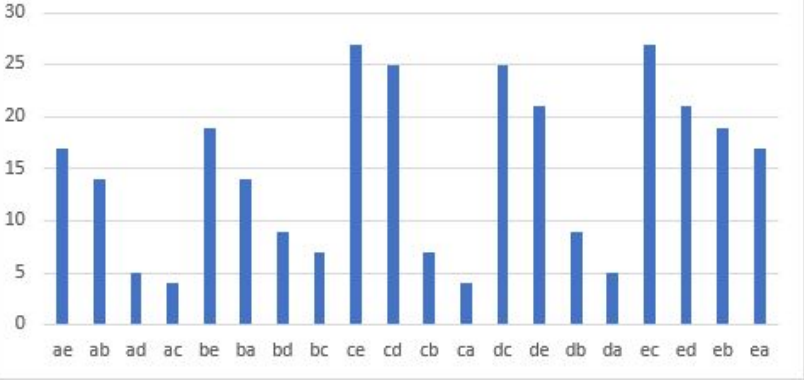

Рисунок 4 — Частота покупки двух товаров одновременно сгупированная

- **Результат:** Самый большой спрос имеет товар e, причём при покупке товара e чаще всего покупают товар c, а при покупке других товаров чаще всего покупают товар e

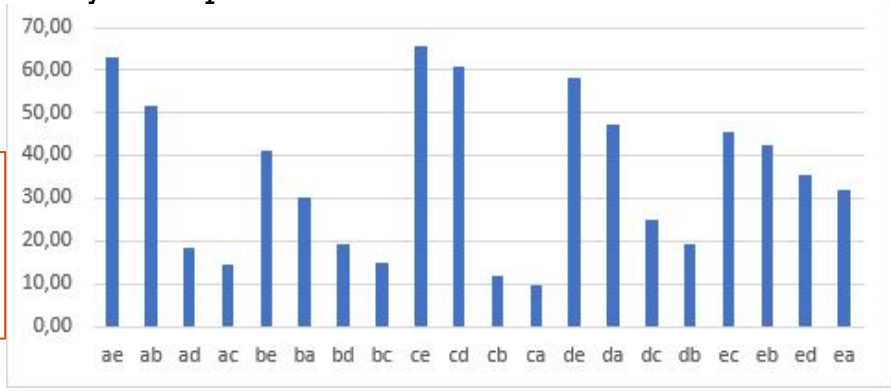

Рисунок 5 — Частота покупки второго товара при покупке первого сгупированная

- **Вывод:** Для получения наибольшего спроса требуется разместить товары категории *e* в центре зала так, чтобы все потребители проходили около него и в случае покупки товаров других категорий наиболее часто сталкивались с товарами категории *e*, в свою очередь, ближе всего к товарам категории *e* требуется разместить товары категории *c* в силу высокого спроса на товары данной категории, при покупке товаров категории *e*

Постановка задачи: Провести оценку классификации данных ионосферы методом KNeighborsClassifier и определить число соседей для получения наиболее эффективной оценки, оценить эффективность классификации при нормализации данных

Метод решения: Обработка данных их классификация с использованием возможностей Jupyter lab

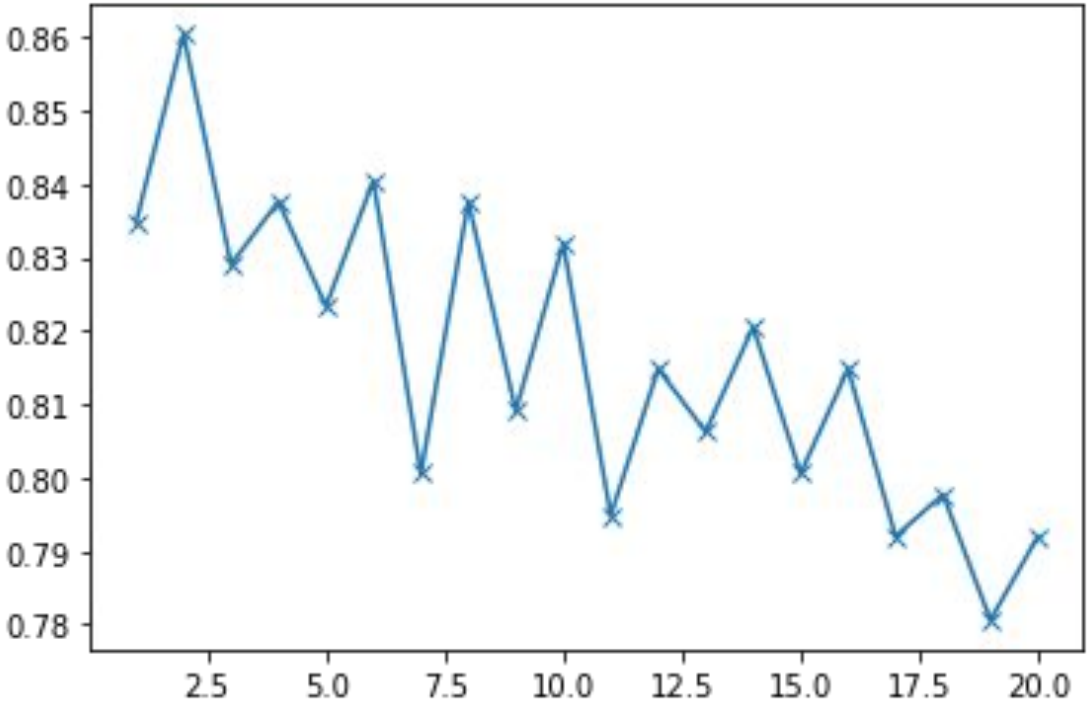

Рисунок 6 — зависимость точности модели от числа соседей

Размерность X:(351, 34)
Размерность Y:(351,)
Обучающая выборка (263,) примеров
Тестовая выборка (88,) примеров

Результат

- Наивысшая точность 0.9% при числе соседей = 2
- Точность cross val на тестовой выборке: 84.18%
- Точность cross val на "перекошенной" тестовой выборке: 80.73%
- Точность cross val на нормализованной тестовой выборке: 85.33%

- **Вывод:** Наибольшая эффективность классификатора достигается при количестве равном 2м, теряя свою эффективность в иных случаях. Также было выявлено, что нормализация данных влияет на качество классификации

Постановка задачи: Провести оценку классификации предоставленных данных различными методами и произвести оценку эффективности

Метод решения: Обработка данных их классификация с использованием возможностей Jupyter lab

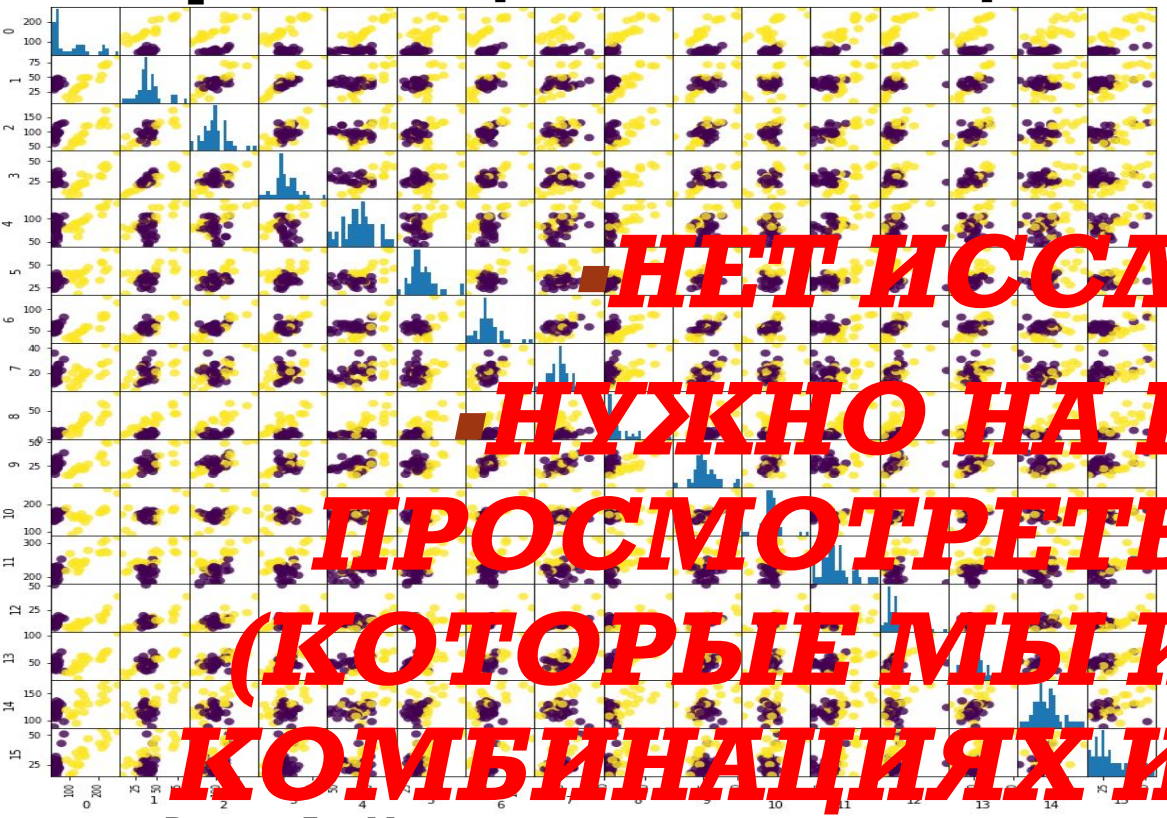

Рисунок 7 — Матрица диаграммы рассеивания

Метод	Train	Test	Train_pca	Test_pca
KNeighbors	100%	100%	100%	100%
SVC	100%	31.25%	100%	31.25%
KNeighborsRegressor	100%	100%	100%	100%
LinearRegression	98.44%	95.01%	94.02%	95.51%
Ridge	98.44%	95.02%	94.02%	95.51%
Lasso	94.24%	94.95%	93.61%	94.97%
LogisticRegression	100%	100%	100%	100%
DecisionTree	100%	100%	100%	100%
RandomForest	100%	100%	100%	100%
GradientBoosting	100%	100%	100%	100%
MLPClassifier	100%	100%	100%	100%

Таблица 1 — Сравнение методов классификации с PCA и без

НЕТ ИССЛЕДОВАНИЯ!!!

НУЖНО НА ВСЕХ МЕТОДАХ ПРОСМОТРЕТЬ ВСЕ ПАРАМЕТРЫ (КОТОРЫЕ МЫ ИЗУЧАЛИ) В РАЗНЫХ КОМБИНАЦИЯХ И ЭТО ПРЕДСТАВИТЬ!

СКАЗАТЬ С КАКИМИ ПАРАМЕТРАМИ ЛУЧШЕ!

- PCA(n_components=2)
- Массив train: (48, 16)
- Массив test: (16, 16)
- Массив train_PCA: (48, 2)
- Массива test_PCA: (16, 2)

Результат

Многие классификаторы показали 100% разделение, что показало линейное разделение данных.

- **Вывод:** Данные, являясь линейно разделимыми, без ошибок классифицируются большинством методов классификации, однако, понимая это, лучшими методами классификации в данном случае будут являться линейные, т.к. они являются наиболее простыми и быстрыми и наименее трудоёмкими

Постановка задачи: Провести оценку классификации данных ionosphere различными методами и произвести оценку эффективности

Метод решения: Обработка данных их классификация с использованием возможностей Jupyter lab

Рисунок 8 — Матрица диаграммы рассеивания

Метод	Train	Test	Train_pca	Test_pca
KNeighbors	90.11%	88.64%	91.63%	90.91%
SVC	99.62%	94.32%	99.24%	90.91%
KNeighborsRegressor	58.53%	59.75%	65.73%	65.75%
LinearRegression	53.48%	46.77%	42.77%	53.04%
Ridge	65.22%	48.98%	42.76%	52.99%
Lasso	0.00%	-0.04%	0.00%	-0.04%
LogisticRegression	91.34%	87.50%	86.69%	87.50%
DecisionTree	100%	89.77%	100%	92.05%
RandomForest	100%	89.77%	100%	92.05%
GradientBoosting	100%	89.77%	100%	92.05%
MLPClassifier	98.48%	97.32%	91.58%	94.32%

Таблица 2 — Сравнение методов классификации с PCA и без

- PCA(n_components=11)
- Массив train: (48, 11)
- Массив test: (16, 16)
- Массив train_PCA: (48, 2)
- Массива test_PCA: (16, 2)

- **Результаты:**
  - Наилучший результат показал классификатор MLP с данным на тесте 97.32%

- **Вывод:** Представленные данные не являются визуально разделимыми. В данном случае большую роль играют параметры, которые будут использоваться в классификаторах, а также выбранное количество компонент в модели PCA. В рассмотренном примере наилучший результат в обоих случаях показал MLPClassifier.

НЕ ИССЛЕДОВАНИЯ!!! НУЖНО НА ВСЕХ МЕТОДАХ ПРОСМОТРЕТЬ ВСЕ ПАРАМЕТРЫ (КОТОРЫЕ МЫ ИЗУЧАЛИ) В РАЗНЫХ КОМБИНАЦИЯХ И ЭТО ПРЕДСТАВИТЬ! СКАЗАТЬ С КАКИМИ ПАРАМЕТРАМИ ЛУЧШЕ!

Постановка задачи: Провести оценку классификации данных ionosphere с помощью деревьев решений и нейронных сетей при random_state=4

Метод решения: Обработка данных их классификация с использованием возможностей Jupyter lab

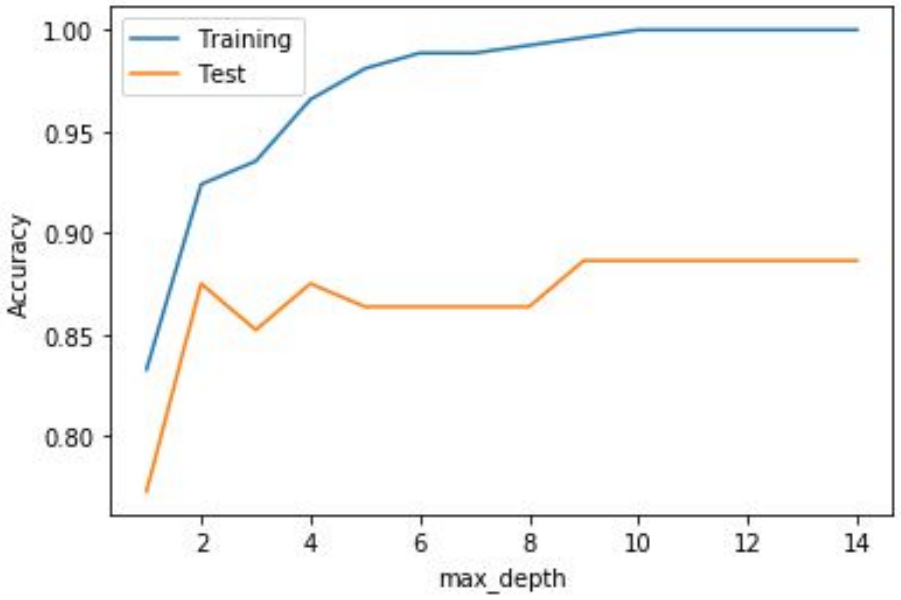

Рисунок 9 — График деревьев решений при разной глубине

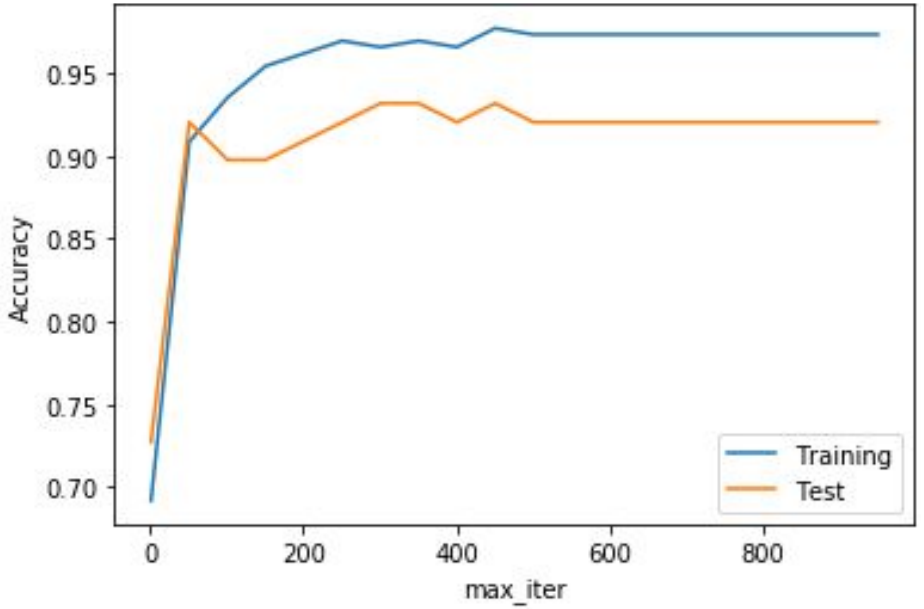

Рисунок 10 — График MLP при разном числе итераций

Лучшие результаты max_iters:

mlp.score train 301: 96.58%
mlp.score test 301: 93.18%

mlp.score train 351: 96.96%
mlp.score test 351: 93.18%

mlp.score train 401: 96.58%
mlp.score test 401: 92.05%

Результат

- Деревья решений с random_state только: test 100%, train 88.64%
- Деревья решений с max_depth=2: train 92.40%, train: 87.50%
- MLP с ACTIVATION tanh: train 100%, train: 89.77%
- MLP без ACTIVATION: train 99.62%, train: 85.23%
- MLP с random_state только: train 98.10%, train: 94.32%
- MLP (solver='adam',hidden=[3,4],a='tahn'): train 91.25%, train: 93.18%

- **Вывод:** На небольших выборках MLP является гораздо менее эффективной, чем деревья решений, однако, достигая достаточно высокой эффективности при больших объёмах данных, в случае большого количества итераций, MLP быстро переобучается, что является её основной проблемой с подбором необходимых входных данных для эффективной работы. Также, деревья решений являются менее трудоёмкими в сравнении с MLP-моделью