

Рубежный контроль №2

Группа: ИУ5Ц-83Б

Студент: Соловьева Александра

Тема: Методы построения моделей машинного обучения.

Задание

Для заданного набора данных (по Вашему варианту) постройте модели классификации или регрессии (в зависимости от конкретной задачи, рассматриваемой в наборе данных). Для построения моделей используйте методы 1 и 2 (по варианту для Вашей группы). Оцените качество моделей на основе подходящих метрик качества (не менее двух метрик). Какие метрики качества Вы использовали и почему? Какие выводы Вы можете сделать о качестве построенных моделей? Для построения моделей необходимо выполнить требуемую предобработку данных: заполнение пропусков, кодирование категориальных признаков, и т.д.

* Номер варианта: 27

* Метод №1: Дерево решений

* Метод №2: Случайный лес

Описание исходного набора

Этот набор данных содержит данные о банкротстве из Тайваньского экономического журнала за 1999–2009 гг.

Данные были получены из Тайваньского экономического журнала за период с 1999 по 2009 год. Банкротство компании было определено на основании правил ведения бизнеса Тайваньской фондовой биржи.

Столбцы: Y = выходной объект, X = входной объект).

- Y - Банкрот?: Ярлык класса
- X1 - ROA(C) до вычета процентов и амортизации до вычета процентов: Рентабельность общих активов(C)
- X2 - ROA(A) до уплаты процентов и % после налогообложения: Рентабельность общих активов(A)
- X3 - ROA(B) до вычета процентов и амортизации после налогообложения: Рентабельность всех активов(B)
- X4 – Валовая прибыль от операционной деятельности: Валовая прибыль/чистая выручка от продаж
- X5 – Валовая прибыль от реализованных продаж: Реализованная валовая прибыль/чистая выручка от продаж
- X6 – Норма операционной прибыли: Операционная прибыль/ Чистая выручка
- X7 – Чистая процентная ставка до налогообложения: Доход/чистая выручка

- X8 – Чистая процентная ставка после уплаты налогов: Чистая прибыль/Чистая выручка
- X9 – Непромышленные доходы и расходы/выручка: Коэффициент чистой внереализационной прибыли
- X10 — Непрерывная процентная ставка (после налогообложения): Чистая прибыль — исключая прибыль или убыток от выбытия / Чистая выручка
- X11 — Ставка операционных расходов: Операционные расходы / Чистая выручка
- X12 — Ставка расходов на исследования и разработки: (Расходы на исследования и разработки) / Чистая выручка
- X13 - Скорость движения денежных средств: Денежный поток от операционных/текущих обязательств
- X14 - Процентная ставка по процентным долгам: Процентные долги/капитал
- X15 - Ставка налога (A): Эффективная налоговая ставка
- X16 - Чистая стоимость одной акции (B): Балансовая стоимость На акцию(B)
- X17 - Чистая стоимость на акцию (A): Балансовая стоимость на акцию(A)
- X18 - Чистая стоимость на акцию (C): Балансовая стоимость на акцию(C)
- X19 - Постоянная прибыль на акцию за последние четыре сезона: прибыль на акцию -Чистый доход
- X20 - Денежный поток на акцию
- X21 — Доход на акцию (в юанях): Продажи на акцию
- X22 — Операционная прибыль на акцию (в юанях): Операционный доход на акцию
- X23 — Чистая прибыль на акцию до налогообложения (в юанях): Доход на акцию до налогообложения
- X24 — Валовой реализованный доход Темп роста прибыли
- X25 - Темп роста операционной прибыли: Рост операционной прибыли
- X26 - Темп роста чистой прибыли после налогообложения: Рост чистой прибыли
- X27 - Темп роста обычной чистой прибыли: Рост текущей операционной прибыли после налогообложения
- X28 - Темп роста постоянной чистой прибыли: Чистая прибыль - Исключая рост прибыли или убытков от выбытия
- X29 - Темп роста общих активов: Рост совокупных активов
- X30 - Темп роста чистой стоимости: Рост совокупных активов
- X31 - Коэффициент темпов роста общей доходности активов: Прибыль к совокупным приростам активов
- X32 - Реинвестирование денежных средств, %: Коэффициент реинвестирования денежных средств
- X33 - Коэффициент текущей ликвидности
- X34 - Коэффициент быстрой ликвидности: тест на кислотность
- X35 - Коэффициент процентных расходов: процентные расходы/общий доход
- X36 - Общий долг/Общий собственный капитал: Коэффициент общих обязательств/капитала
- X37 - Коэффициент долга, % : Обязательства/Общие активы
- X38 - Чистая стоимость/Активы: Собственный капитал/Общие активы
- X39 - Коэффициент пригодности долгосрочного фонда (A): (Долгосрочные обязательства + Собственный капитал)/Основные активы
- X40 - Зависимость от заимствования: Стоимость процентного долга
- X41 – Условные обязательства/Собственный капитал: Условные обязательства/Капитал
- X42 – Операционная прибыль/Оплаченный капитал: Операционный доход/Капитал
- X43 – Чистая прибыль до налогообложения/Оплаченный капитал: Доход/капитал до налогообложения
- X44 — Товарно-материальные запасы и дебиторская задолженность/Чистая стоимость: (Запасы+Дебиторская задолженность)/Собственный капитал
- X45 — Общий оборот активов
- X46 — Оборачиваемость дебиторской задолженности

- X47 — Среднее количество дней сбора: Дни непогашенной дебиторской задолженности X48 — Коэффициент оборачиваемости запасов (раз)
- X49 — Оборачиваемость основных средств Частота
- X50 - Коэффициент оборота чистой стоимости (раз): Оборачиваемость собственного капитала
- X51 - Выручка на человека: Продажи на одного работника
- X52 - Операционная прибыль на человека: Операционный доход на одного работника
- X53 - Коэффициент распределения на человека: Основные средства на одного работника
- X54 - Общий оборотный капитал Активы
- X55 - Быстрые активы/Итого активы
- X56 - Оборотные активы/Итого активы
- X57 - Денежные средства/Итого активы
- X58 – Оборотные активы/Текущие обязательства
- X59 – Денежные средства/Текущие обязательства
- X60 – Текущие обязательства к активам
- X61 – Оборотные средства к обязательствам
- X62 – Запасы/оборотный капитал
- X64 – Текущие обязательства/обязательства
- X65 – Оборотный капитал/капитал
- X66 - Текущие обязательства/капитал
- X67 - Долгосрочные обязательства по текущим активам
- X68 - Нераспределенная прибыль к общей сумме активов
- X69 - Общая прибыль/общая сумма расходов
- X70 - Общая сумма расходов/активов
- X71 - Коэффициент оборачиваемости текущих активов: Текущие активы к продажам
- X72 - Быстрая оборачиваемость активов Отношение оборотных средств к продажам
- X73 - Коэффициент оборачиваемости оборотного капитала: оборотный капитал к продажам
- X74 - Норма денежного оборота: Денежные средства к продажам
- X75 - Денежные потоки к продажам
- X76 - Основные средства к активам
- X77 - Текущие обязательства к обязательствам
- X78 - Текущие обязательства к капиталу
- X79 - Капитал к долгосрочным обязательствам
- X80 - Денежные потоки к общим активам
- X81 - Денежный поток к обязательствам
- X82 - Финансовый директор к активам
- X83 - Денежный поток к собственному капиталу
- X84 - Текущие обязательства к оборотным активам
- X85 - Флаг пассивов-активов: 1, если общая сумма обязательств превышает общую сумму активов, 0 в противном случае
- X86 - чистая прибыль к общей сумме активов
- X87 - общая сумма активов к цене ВВП
- X88 - Интервал без кредита
- X89 - Валовая прибыль к продажам
- X90 - Чистая прибыль к акционерному капиталу
- X91 - Обязательства к капиталу
- X92 — Степень финансового рычага (DFL)
- X93 — Коэффициент покрытия процентов (процентные расходы к EBIT)
- X94 — Флаг чистой прибыли: 1, если чистая прибыль отрицательная за последние два года, 0 в противном случае
- X95 — отношение капитала к обязательствам

In [21]:

```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib
import matplotlib.pyplot as plt
import warnings
warnings.filterwarnings("ignore")
from IPython.display import Image
from sklearn.metrics import mean_absolute_error, mean_squared_error
from sklearn.preprocessing import MinMaxScaler
from sklearn.neighbors import KNeighborsRegressor, KNeighborsClassifier
from sklearn.model_selection import train_test_split, GridSearchCV
from sklearn.impute import SimpleImputer, MissingIndicator
from sklearn.preprocessing import LabelEncoder, OneHotEncoder, MinMaxScaler
from sklearn.model_selection import StratifiedKFold
from sklearn.model_selection import cross_val_score
```

In [22]:

```
#Загрузка датасета
df = pd.read_csv('data.csv', sep="," )
```

In [23]:

```
df.head()
```

Out[23]:

| | Bankrupt? | ROA(C) before interest and depreciation before interest | ROA(A) before interest and % after tax | ROA(B) before interest and depreciation after tax | Operating Gross Margin | Realized Sales Gross Margin | Operating Profit Rate | Pre-tax net Interest Rate |
|---|-----------|--|--|---|------------------------------|--------------------------------------|-----------------------------|------------------------------------|
| 0 | 1 | 0.370594 | 0.424389 | 0.405750 | 0.601457 | 0.601457 | 0.998969 | 0.796887 |
| 1 | 1 | 0.464291 | 0.538214 | 0.516730 | 0.610235 | 0.610235 | 0.998946 | 0.797380 |
| 2 | 1 | 0.426071 | 0.499019 | 0.472295 | 0.601450 | 0.601364 | 0.998857 | 0.796403 |
| 3 | 1 | 0.399844 | 0.451265 | 0.457733 | 0.583541 | 0.583541 | 0.998700 | 0.796967 |
| 4 | 1 | 0.465022 | 0.538432 | 0.522298 | 0.598783 | 0.598783 | 0.998973 | 0.797366 |

5 rows × 96 columns



In [24]:

```
# Размер набора данных
df.shape
```

Out[24]:

```
(6819, 96)
```

In [25]:

```
# Типы колонок
df.dtypes
```

Out[25]:

```
Bankrupt?                                int64
ROA(C) before interest and depreciation before interest  float64
ROA(A) before interest and % after tax                    float64
ROA(B) before interest and depreciation after tax         float64
Operating Gross Margin                                   float64
...
Liability to Equity                                       float64
Degree of Financial Leverage (DFL)                       float64
Interest Coverage Ratio (Interest expense to EBIT)       float64
Net Income Flag                                           int64
Equity to Liability                                       float64
Length: 96, dtype: object
```

In [26]:

```
# Проверим есть ли пропущенные значения
df.isnull().sum()
```

Out[26]:

```
Bankrupt?                                0
ROA(C) before interest and depreciation before interest  0
ROA(A) before interest and % after tax                    0
ROA(B) before interest and depreciation after tax         0
Operating Gross Margin                                   0
..
Liability to Equity                                       0
Degree of Financial Leverage (DFL)                       0
Interest Coverage Ratio (Interest expense to EBIT)       0
Net Income Flag                                           0
Equity to Liability                                       0
Length: 96, dtype: int64
```

В наборе нет пропусков, следовательно не нужно их обрабатывать

Обучающая и тестовая выборка

In [27]:

```
#data_scaled.dtypes
X = df.drop(columns=' ROA(C) before interest and depreciation before interest')
Y = df[' ROA(C) before interest and depreciation before interest']
```

In [28]:

```
# Входные данные
X.head()
```

Out[28]:

| | Bankrupt? | ROA(A) before interest and % after tax | ROA(B) before interest and depreciation after tax | Operating Gross Margin | Realized Sales Gross Margin | Operating Profit Rate | Pre-tax net Interest Rate | After-tax net Interest Rate | e |
|---|-----------|--|---|------------------------------|--------------------------------------|-----------------------------|------------------------------------|--------------------------------------|---|
| 0 | 1 | 0.424389 | 0.405750 | 0.601457 | 0.601457 | 0.998969 | 0.796887 | 0.808809 | |
| 1 | 1 | 0.538214 | 0.516730 | 0.610235 | 0.610235 | 0.998946 | 0.797380 | 0.809301 | |
| 2 | 1 | 0.499019 | 0.472295 | 0.601450 | 0.601364 | 0.998857 | 0.796403 | 0.808388 | |
| 3 | 1 | 0.451265 | 0.457733 | 0.583541 | 0.583541 | 0.998700 | 0.796967 | 0.808966 | |
| 4 | 1 | 0.538432 | 0.522298 | 0.598783 | 0.598783 | 0.998973 | 0.797366 | 0.809304 | |

5 rows × 95 columns

In [29]:

```
# Выходные данные
Y.head()
```

Out[29]:

```
0    0.370594
1    0.464291
2    0.426071
3    0.399844
4    0.465022
Name: ROA(C) before interest and depreciation before interest, dtype: float64
```

In [30]:

```
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, random_state = 2022, test_
```

In [31]:

```
# Входные параметры обучающей выборки
X_train.head()
```

Out[31]:

| | Bankrupt? | ROA(A) before interest and % after tax | ROA(B) before interest and depreciation after tax | Operating Gross Margin | Realized Sales Gross Margin | Operating Profit Rate | Pre-tax net Interest Rate | After-tax net Interest Rate |
|------|-----------|--|---|------------------------------|--------------------------------------|-----------------------------|------------------------------------|--------------------------------------|
| 6108 | 0 | 0.485172 | 0.495101 | 0.594683 | 0.594683 | 0.998955 | 0.797321 | 0.809252 |
| 6226 | 0 | 0.566289 | 0.548477 | 0.601940 | 0.601940 | 0.999015 | 0.797451 | 0.809358 |
| 3901 | 0 | 0.508995 | 0.536271 | 0.595346 | 0.595346 | 0.998899 | 0.797276 | 0.809213 |
| 873 | 0 | 0.522623 | 0.514107 | 0.605760 | 0.605803 | 0.998937 | 0.797314 | 0.809215 |
| 1091 | 0 | 0.547427 | 0.543337 | 0.607424 | 0.607417 | 0.999008 | 0.797445 | 0.809360 |

5 rows × 95 columns

In [32]:

```
# Входные параметры тестовой выборки
X_test.head()
```

Out[32]:

| | Bankrupt? | ROA(A) before interest and % after tax | ROA(B) before interest and depreciation after tax | Operating Gross Margin | Realized Sales Gross Margin | Operating Profit Rate | Pre-tax net Interest Rate | After-tax net Interest Rate |
|------|-----------|--|---|------------------------------|--------------------------------------|-----------------------------|------------------------------------|--------------------------------------|
| 4796 | 0 | 0.588585 | 0.620965 | 0.605277 | 0.605277 | 0.999108 | 0.797548 | 0.809460 |
| 5460 | 0 | 0.601614 | 0.587772 | 0.608167 | 0.608138 | 0.999078 | 0.797652 | 0.809506 |
| 6306 | 0 | 0.563290 | 0.553670 | 0.602978 | 0.602978 | 0.999034 | 0.797468 | 0.809370 |
| 3339 | 0 | 0.558166 | 0.604101 | 0.609615 | 0.609615 | 0.999082 | 0.797530 | 0.809440 |
| 3666 | 0 | 0.603740 | 0.600996 | 0.606632 | 0.606632 | 0.999134 | 0.797583 | 0.809467 |

5 rows × 95 columns

In [33]:

```
# Выходные парметры обучающей выборки
Y_train.head()
```

Out[33]:

```
6108    0.448301
6226    0.504461
3901    0.491786
873     0.471213
1091    0.495296
Name: ROA(C) before interest and depreciation before interest, dtype: float64
```

In [34]:

```
# Выходные парметры обучающей выборки
Y_train.head()
```

Out[34]:

```
6108    0.448301
6226    0.504461
3901    0.491786
873     0.471213
1091    0.495296
Name: ROA(C) before interest and depreciation before interest, dtype: float64
```

Дерево решений

In [35]:

```
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, accuracy_score
from sklearn.svm import SVC, NuSVC, LinearSVC
from sklearn.tree import DecisionTreeClassifier, DecisionTreeRegressor
from sklearn.model_selection import GridSearchCV
from sklearn import tree
from IPython.core.display import HTML
from sklearn.tree import export_text
from operator import itemgetter
```

In [36]:

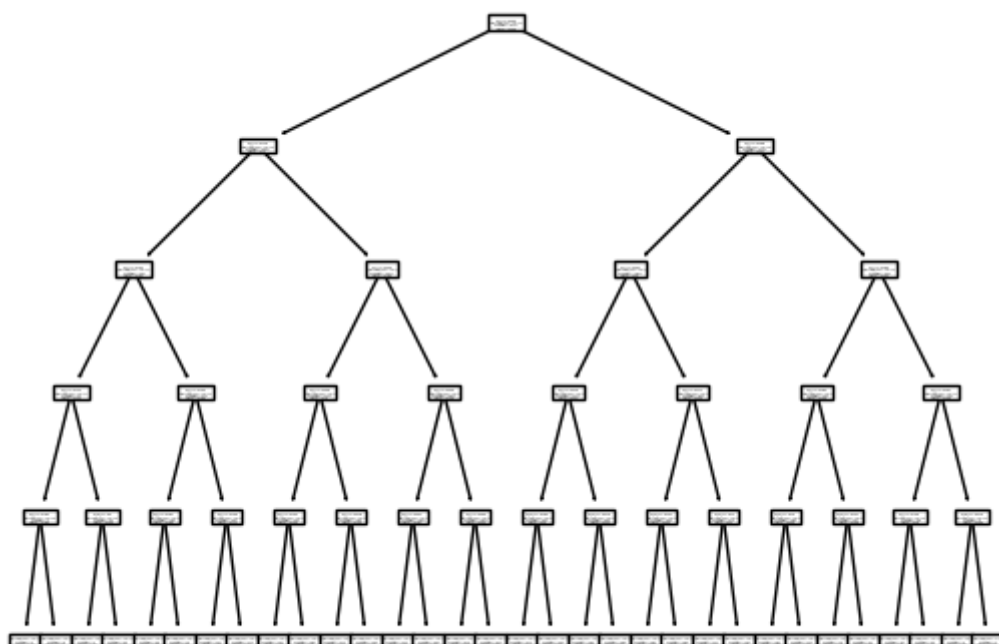
```
dt_none = DecisionTreeRegressor(max_depth=5)
dt_none.fit(X_train, Y_train)
```

Out[36]:

```
DecisionTreeRegressor
DecisionTreeRegressor(max_depth=5)
```


In [37]:

```
1 tree.plot_tree(dt_none);
```



In [38]:

```
tree_rules = export_text(dt_none, feature_names=list(X.columns))
HTML('<pre>' + tree_rules + '</pre>')
```

Out[38]:

```
|--- ROA(B) before interest and depreciation after tax <= 0.56
|   |--- ROA(B) before interest and depreciation after tax <= 0.48
|   |   |--- ROA(B) before interest and depreciation after tax <= 0.3
4
|   |   |   |--- ROA(B) before interest and depreciation after tax <=
0.14
|   |   |   |   |--- ROA(B) before interest and depreciation after ta
x <= 0.04
|   |   |   |   |   |--- value: [0.01]
|   |   |   |   |   |--- ROA(B) before interest and depreciation after ta
x > 0.04
|   |   |   |   |   |   |--- value: [0.09]
|   |   |   |   |   |   |--- ROA(B) before interest and depreciation after tax >
0.14
|   |   |   |   |   |   |--- Current Asset Turnover Rate <= 0.00
```

Случайный лес

In [39]:

```
from sklearn.ensemble import RandomForestRegressor
from sklearn.tree import DecisionTreeClassifier, DecisionTreeRegressor, export_graphviz
```

In [40]:

```
# Обучим регрессор на 4 деревьях
tree1 = RandomForestRegressor(n_estimators=4, oob_score=True, random_state=2022)
tree1.fit(X, Y)
```

Out[40]:

```
RandomForestRegressor
RandomForestRegressor(n_estimators=4, oob_score=True, random_state=2022)
```

In [41]:

```
# Out-of-bag error, возвращаемый регрессором
tree1.oob_score_, 1-tree1.oob_score_
```

Out[41]:

```
(-10.813179067920368, 11.813179067920368)
```

Результат

In [42]:

```
regressor = RandomForestRegressor(n_estimators=4, random_state=2022)
regressor.fit(X_train, Y_train)
y_pred = regressor.predict(X_test)
```

In [43]:

```
print('Mean Absolute Error:', mean_absolute_error(Y_test, y_pred))
print('Mean Squared Error:', mean_squared_error(Y_test, y_pred))
print('Root Mean Squared Error:', np.sqrt(mean_squared_error(Y_test, y_pred)))
```

```
Mean Absolute Error: 0.004380831810683535
Mean Squared Error: 5.0548138274317145e-05
Root Mean Squared Error: 0.0071097213921726314
```