Московский государственный технический университет им. Н. Э. Баумана

Курс «Технологии машинного обучения»

Отчёт по рубежному контролю №2

«Методы построения моделей машинного обучения.»

Вариант № 9

Выполнил:
Дувакин А.В.
группа ИУ5-63Б

Проверил:
Гапанюк Ю.Е.

Дата: 15.04.25

Подпись:

Дата:

Подпись:

Москва, 2025 г.

Задание:

Номер варианта: **9**

Номер набора данных, указанного в задаче: **9** (https://www.kaggle.com/arindam235/startup-investments-crunchbase)

Метод №1: **Дерево решений**

Метод №2: **Случайный лес**

Ход выполнения:

МГТУ им. Н.Э.Баумана | ИУ5 | 6 семестр | ТМО | РК№2

ИУ5-63Б | Ювенский Лев | Вариант № 19

Задание: https://github.com/ugapanyuk/courses_current/wiki/TMO_RK_2

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.

Метод №1: Дерево решений

Метод №2: Случайный лес

Датасет: https://www.kaggle.com/datasets/arindam235/startup-investments-crunchbase

Загрузка и первичный анализ

```
[152] import numpy as np
      import pandas as pd
      import matplotlib.pyplot as plt
      import seaborn as sns
      from tabulate import tabulate

      from sklearn.compose import ColumnTransformer
      from sklearn.model_selection import train_test_split, GridSearchCV
      from sklearn.tree import DecisionTreeClassifier
      from sklearn.ensemble import RandomForestClassifier
      from sklearn.preprocessing import OneHotEncoder, StandardScaler, LabelEncoder

      from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score
      from sklearn.metrics import confusion_matrix, ConfusionMatrixDisplay
```

```
[153] df = pd.read_csv("/dataset.csv", encoding="ISO-8859-1")
```

```
[154] df.head()
```

	permalink	name	homepage_url	category_list	market	funding_total_usd	status	country_code	state_code	region	...	seco

Файлы

Подготовка датасета

Удаление колонок

- `permalink, name, homepage_url`, поскольку они носят информационных характер и не влияют на определение статуса компании
- `region`, поскольку она избыточна для определения метоположения компании
- `founded_at`, 'founded_quarter', 'founded_year'. Будем использовать значения из колонок `founded_month`
- `funding_total_usd`, поскольку она содержит данные в некорректном формате
- `category_list`, поскольку она содержит список категорий, который тяжело обработать в рамках данной задачи

```
[156] df = df.drop(['permalink', 'name', 'homepage_url', 'category_list', 'region', 'founded_at', 'founded_quarter', 'founded_year', 'funding_total_usd'], axis=1)
```

Удаление пропусков

Удалим все строки, содержащие null хотя бы в одной колонке

```
[157] df = df.dropna()
```

```
[158] df.info()
      df.head()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 23217 entries, 0 to 49437
Data columns (total 30 columns):
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   market                23217 non-null  object
 1   status                23217 non-null  object
 2   country_code          23217 non-null  object
 3   state_code            23217 non-null  object
 4   city                  23217 non-null  object
 5   funding_rounds        23217 non-null  float64
 6   founded_month         23217 non-null  object
 7   first_funding_at      23217 non-null  object
 8   last_funding_at       23217 non-null  object
 9   seed                  23217 non-null  float64
 10  venture               23217 non-null  float64
 11  equity_crowdfunding   23217 non-null  float64
 12  undisclosed           23217 non-null  float64
 13  convertible_note      23217 non-null  float64
 14  debt_financing        23217 non-null  float64
 15  angel                 23217 non-null  float64
 16  grant                 23217 non-null  float64
```

```
print(Y_test.shape)
```

```
(18572, 29)
(4644, 29)
(18572,)
(4644,)
```

Обучение моделей

Дерево решений

```
[171] clf = GridSearchCV(DecisionTreeClassifier(random_state=10, class_weight='balanced'), {'max_depth':range(3,40)})

      clf.fit(X_train, Y_train)

      dt_clf = clf.best_estimator_

      print(clf.best_score_, clf.best_params_)
```

```
0.7813913380022918 {'max_depth': 35}
```

Случайный лес

```
[172] rf_clf = RandomForestClassifier(random_state=10,  class_weight='balanced', n_jobs=-1)

      rf_clf.fit(X_train, Y_train)
```

```
RandomForestClassifier(class_weight='balanced', n_jobs=-1, random_state=10)
```

Оценка качества моделей

Метрики

Accuracy (Точность) = (число правильно предсказанных классов) / (общее число объектов)

Precision для каждого класса — это отношение числа правильно классифицированных объектов данного класса к числу всех объектов, которые были предсказаны как принадлежащие этому классу.

Recall для каждого класса — это отношение числа правильно классифицированных объектов данного класса к числу всех реально принадлежат этому классу.

```
print(tabulate(data, headers=headers, tablefmt="grid"))
```

```
+--------------------+--------------------+--------------------+
| Метрика \ модель   |   Дерево решений   |   Случайный лес    |
+====================+====================+====================+
| accuracy           |              0.776 |              0.847 |
+--------------------+--------------------+--------------------+
| precision          |              0.779 |              0.791 |
+--------------------+--------------------+--------------------+
| recall             |              0.776 |              0.847 |
+--------------------+--------------------+--------------------+
| f1                 |              0.777 |              0.803 |
+--------------------+--------------------+--------------------+
```

```
cm_dt = confusion_matrix(Y_test, y_pred_dt)
cm_rf = confusion_matrix(Y_test, y_pred_rf)

fig, axes = plt.subplots(1, 2, figsize=(12, 4))

disp1 = ConfusionMatrixDisplay(confusion_matrix=cm_dt)
disp1.plot(ax=axes[0], cmap=plt.cm.Blues, colorbar=False)
axes[0].set_title('Дерево решений')

disp2 = ConfusionMatrixDisplay(confusion_matrix=cm_rf)
disp2.plot(ax=axes[1], cmap=plt.cm.Blues, colorbar=False)
axes[1].set_title('Случайный лес')
```

```
Text(0.5, 1.0, 'Случайный лес')
```