# Московский государственный технический университет им. Н. Э. Баумана

Курс «Технологии машинного обучения»

Отчёт по рубежному контролю №2

«Методы построения моделей машинного обучения.»

Вариант № 19

Выполнил:	Проверил:
Ювенский Л.А.	Гапанюк Ю.Е.
группа ИУ5-63Б	

Дата: 15.04.25 Дата:

Подпись:

### Задание:

Номер варианта: 19

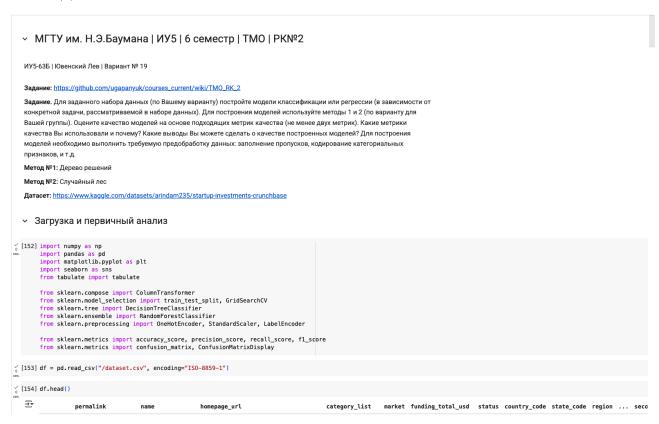
Номер набора данных, указанного в задаче: 19

(https://www.kaggle.com/arindam235/startup-investments-crunchbase)

Метод №1: Дерево решений

Метод №2: Случайный лес

## Ход выполнения:



# 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 23217 non-null object 23217 non-null float64 0 market 1 status status country\_code state\_code city funding\_rounds founded\_month first\_funding\_at last\_funding\_at seed venture 10 venture 11 equity\_crowdfunding 12 undisclosed 13 convertible\_note 14 debt\_financing 15 angel 23217 non-null 23217 non-null 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 RandomForestClassifier(class\_weight='balanced', n\_jobs=-1, random\_state=10) Оценка качества моделей

Метрики

**Ассигасу (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 SFAhjdv6izDi9Jc#scrollTo=\_ | реально принадлежат этому классу.

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



7	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))

$$\label{disp1} \begin{split} & \text{disp1} = \text{ConfusionMatrixDisplay(confusion\_matrix=cm\_dt)} \\ & \text{disp1.plot(ax=axes[0], cmap=plt.cm.Blues, colorbar=False)} \\ & \text{axes[0].set\_title('Дерево решений')} \end{split}$$

disp2 = ConfusionMatrixDisplay(confusion\_matrix=cm\_rf) disp2.plot(ax=axes[1], cmap=plt.cm.Blues, colorbar=False) axes[1].set\_title('Cnyчайный лес')

#### <u></u> Техt(0.5, 1.0, 'Случайный лес')

