# Лабораторная работа №5

Робертс Даниил Александрович

Москва 2022

## 1 Лабораторная работа №5. Ансамбли моделей машинного обучения.

### 1.1 Задание

- Выберите набор данных (датасет) для решения задачи классификации или регресии.
- В случае необходимости проведите удаление или заполнение пропусков и кодирование категориальных признаков.
- С использованием метода train_test_split разделите выборку на обучающую и тестовую.
- Обучите следующие ансамблевые модели:
  - одну из моделей группы бэггинга (бэггинг или случайный лес или сверхслучайные деревья);
  - одну из моделей группы бустинга;
  - одну из моделей группы стекинга.
- Оцените качество моделей с помощью одной из подходящих для задачи метрик. Сравните качество полученных моделей.

[1]:
```
#Импорт библиотек:
from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.ensemble import GradientBoostingClassifier
from sklearn.ensemble import BaggingClassifier
from sklearn.linear_model import RidgeClassifier
import seaborn as sns
%matplotlib inline
sns.set(style="ticks")
```

Данные доступны были взяты из стандартного набора данных библиотеки sklearn. Данные представляют из себя информацию о видах ириса. Модель для решения задачи классификации.

[2]:
```
iris = load_iris()

iris_X_train, iris_X_test, iris_y_train, iris_y_test = train_test_split(
    iris.data, iris.target, test_size=0.5, random_state=1)
```

Бэггинг

[3]:
```
bg = BaggingClassifier(n_estimators=15, oob_score=True, random_state=1)
bg.fit(iris_X_train, iris_y_train)
```

[3]:
```
BaggingClassifier(n_estimators=15, oob_score=True, random_state=1)
```

[4]:
```
# Оценка качества модели
accuracy_score(iris_y_test, bg.predict(iris_X_test))
```

[4]:
```
0.9466666666666667
```

Бустинг (градиентный спуск)

[5]:
```
gb = GradientBoostingClassifier(n_estimators=15, random_state=1)
gb.fit(iris_X_train, iris_y_train)
```

[5]:
```
GradientBoostingClassifier(n_estimators=15, random_state=1)
```

[6]:
```
# Оценка качества модели
accuracy_score(iris_y_test, gb.predict(iris_X_test))
```

[6]:
```
0.96
```

Стекинг

[7]:
```
from heamy.estimator import Regressor
from heamy.pipeline import ModelsPipeline
from heamy.dataset import Dataset

dataset = Dataset(iris_X_train, iris_y_train, iris_X_test, iris_y_test)

model_tree = Regressor(dataset=dataset, estimator=DecisionTreeClassifier,␣
 ↪name='tree')
model_lr = Regressor(dataset=dataset, estimator=RidgeClassifier,name='lr')
model_rf = Regressor(dataset=dataset, estimator=RandomForestClassifier,␣
 ↪parameters={'n_estimators': 50},name='rf')

pipeline = ModelsPipeline(model_tree, model_lr, model_rf)
stack_ds = pipeline.stack(k=10, seed=1)
# модель второго уровня
stacker = Regressor(dataset=stack_ds, estimator=DecisionTreeClassifier)
results = stacker.validate(k=10,scorer=accuracy_score)
```

```
Metric: accuracy_score
Folds accuracy: [1.0, 1.0, 0.875, 0.875, 1.0, 1.0, 0.8571428571428571,
0.8571428571428571, 0.8571428571428571, 0.8571428571428571]
Mean accuracy: 0.9178571428571429
Standard Deviation: 0.06738557951469006
Variance: 0.0045408163265306155
```