

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

Курс «Технологии машинного обучения»

Отчёт по лабораторной работе №6

Выполнил:
Дувакин А.В.
группа ИУ5-63Б

Проверил:
Гапанюк Ю.Е.

Дата: 25.04.25

Дата:

Подпись:

Подпись:

Москва, 2025 г.

Цель лабораторной работы: изучение ансамблей моделей машинного обучения.

Задание:

1. Выберите набор данных (датасет) для решения задачи классификации или регрессии.
2. В случае необходимости проведите удаление или заполнение пропусков и кодирование категориальных признаков.
3. С использованием метода `train_test_split` разделите выборку на обучающую и тестовую.
4. Обучите следующие ансамблевые модели:
 - a. одну из моделей группы стекинга.
 - b. модель многослойного перцептрона. По желанию, вместо библиотеки `scikit-learn` возможно использование библиотек TensorFlow, PyTorch или других аналогичных библиотек.
 - c. двумя методами на выбор из семейства МГУА (один из линейных методов COMBI / MULTI + один из нелинейных методов MIA / RIA) с использованием библиотеки `gmdh`.

Ход выполнения:

The screenshot shows a Jupyter Notebook interface. On the left is a sidebar with a table of contents. The main area displays the task description and the installation of the 'heamy' library.

Содержание

МГТУ им. Н.Э.Баумана ИУ5 6 семестр ТМО ЛР№6
Загрузка датасета
Подготовка датасета
Разделение выборки
Обучение и оценка моделей
Оценка качества моделей
Модель группы стекинга
Модель многослойного перцептрона
МГУА
COMBI
MIA
+ Раздел

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

https://github.com/vgapanyuk/courses_current/wiki/LAB_TMO_ENSEMBLES_2

Задание

1. Выберите набор данных (датасет) для решения задачи классификации или регрессии.
2. В случае необходимости проведите удаление или заполнение пропусков и кодирование категориальных признаков.
3. С использованием метода `train_test_split` разделите выборку на обучающую и тестовую.
4. Обучите следующие ансамблевые модели:
 - одну из моделей группы стекинга.
 - модель многослойного перцептрона. По желанию, вместо библиотеки `scikit-learn` возможно использование библиотек TensorFlow, PyTorch или других аналогичных библиотек.
 - двумя методами на выбор из семейства МГУА (один из линейных методов COMBI / MULTI + один из нелинейных методов MIA / RIA) с использованием библиотеки `gmdh`.
 - В настоящее время библиотека МГУА не позволяет решать задачу классификации !!!
5. Оцените качество моделей с помощью одной из подходящих для задачи метрик. Сравните качество полученных моделей.
6. В телеграмм-канале потока ИУ5 в теме ТМО_МГУА напишите обратную связь по использованию библиотеки `gmdh`:
 - обнаруженные баги с приложением скриншотов ошибок, за каждый найденный баг +1 балл на экзамене;
 - опечатки в документации или учебном пособии МГУА;
 - возникшие вопросы или трудности при установке и использовании библиотек;
 - любая другая информация (критика, предложения по улучшению и тд).

```
[ ] !pip3 install heamy
```

```
Requirement already satisfied: heamy in /usr/local/lib/python3.11/dist-packages (0.0.7)
Requirement already satisfied: scikit-learn==0.17.0 in /usr/local/lib/python3.11/dist-packages (from heamy) (1.6.1)
Requirement already satisfied: pandas==0.17.0 in /usr/local/lib/python3.11/dist-packages (from heamy) (2.2.2)
Requirement already satisfied: six==1.10.0 in /usr/local/lib/python3.11/dist-packages (from heamy) (1.17.0)
Requirement already satisfied: scipy==0.16.0 in /usr/local/lib/python3.11/dist-packages (from heamy) (1.14.1)
Requirement already satisfied: numpy==1.7.0 in /usr/local/lib/python3.11/dist-packages (from heamy) (1.24.4)
Requirement already satisfied: python-dateutil==2.8.2 in /usr/local/lib/python3.11/dist-packages (from pandas==0.17.0->heamy) (2.8.2)
Requirement already satisfied: pytz==2020.1 in /usr/local/lib/python3.11/dist-packages (from pandas==0.17.0->heamy) (2025.2)
Requirement already satisfied: tzdata==2022.7 in /usr/local/lib/python3.11/dist-packages (from pandas==0.17.0->heamy) (2025.2)
Requirement already satisfied: joblib==1.2.0 in /usr/local/lib/python3.11/dist-packages (from scikit-learn==0.17.0->heamy) (1.4.2)
Requirement already satisfied: threadpoolctl==3.1.0 in /usr/local/lib/python3.11/dist-packages (from scikit-learn==0.17.0->heamy) (3.6.0)
```

```
import numpy as np
```

Содержание

МГТУ им. Н.Э.Баумана | ИУ5 | 6 семестр | ТМО | ЛРН-6

Загрузка датасета

Подготовка датасета

Разделение выборки

Обучение и оценка моделей

Оценка качества моделей

Модель группы стекинга

Модель многослойного перцептрона

МГУА

СОМБИ

МИА

Раздел

from inspect import getargspec, getargvaluespec

Загрузка датасета

Признак	Описание
Square_Footage	The size of the house in square feet. Larger homes typically have higher prices.
Num_Bedrooms	The number of bedrooms in the house. More bedrooms generally increase the value of a home.
Num_Bathrooms	The number of bathrooms in the house. Houses with more bathrooms are typically priced higher.
Year_Built	The year the house was built. Older houses may be priced lower due to wear and tear.
Lot_Size	The size of the lot the house is built on, measured in acres. Larger lots tend to add value to a property.
Garage_Size	The number of cars that can fit in the garage. Houses with larger garages are usually more expensive.
Neighborhood_Quality	A rating of the neighborhood's quality on a scale of 1-10, where 10 indicates a high-quality neighborhood. Better neighborhoods usually command higher prices.
House_Price (Target Variable)	The price of the house, which is the dependent variable you aim to predict.

```
[ ] data = pd.read_csv("/house_price_dataset.csv")
```

```
<frozen importlib._bootstrap>:1047: ImportWarning: _PyDriveImportHook.find_spec() not found; falling back to find_module()
<frozen importlib._bootstrap>:1047: ImportWarning: _BokehImportHook.find_spec() not found; falling back to find_module()
<frozen importlib._bootstrap>:1047: ImportWarning: _PyDriveImportHook.find_spec() not found; falling back to find_module()
<frozen importlib._bootstrap>:1047: ImportWarning: _BokehImportHook.find_spec() not found; falling back to find_module()
```

```
data.head()
```

```
<frozen importlib._bootstrap>:1047: ImportWarning: _PyDriveImportHook.find_spec() not found; falling back to find_module()
<frozen importlib._bootstrap>:1047: ImportWarning: _BokehImportHook.find_spec() not found; falling back to find_module()
<frozen importlib._bootstrap>:1047: ImportWarning: _PyDriveImportHook.find_spec() not found; falling back to find_module()
<frozen importlib._bootstrap>:1047: ImportWarning: _BokehImportHook.find_spec() not found; falling back to find_module()
<frozen importlib._bootstrap>:1047: ImportWarning: _PyDriveImportHook.find_spec() not found; falling back to find_module()
<frozen importlib._bootstrap>:1047: ImportWarning: _BokehImportHook.find_spec() not found; falling back to find_module()
```

	Square_Footage	Num_Bedrooms	Num_Bathrooms	Year_Built	Lot_Size	Garage_Size	Neighborhood_Quality	House_Price
0	1360	2	1	1981	0.599637	0	5	2.623829e+05
1	4272	3	3	2016	4.753014	1	6	9.852609e+05
2	3592	1	2	2016	3.634823	0	9	7.779774e+05
3	966	1	2	1977	2.730667	1	8	2.296989e+05
4	4926	2	1	1993	4.699073	0	8	1.041741e+06

```
[ ] data.isnull().sum()
```

```
0
```

	Square_Footage
0	0

Команды + Код + Текст

Подключиться повторно

Содержание

МГТУ им. Н.Э.Баумана | ИУ5 | 6 семестр | ТМО | ЛРН-6

Загрузка датасета

Подготовка датасета

Разделение выборки

Обучение и оценка моделей

Оценка качества моделей

Модель группы стекинга

Модель многослойного перцептрона

МГУА

СОМБИ

МИА

Раздел

```
fig, ax = plt.subplots(figsize=(12,5))
sns.heatmap(data.corr(method='pearson'), ax=ax, annot=True, cbar=False, fmt='.2f')
```

```
<Axes: >
<frozen importlib._bootstrap>:1047: ImportWarning: _PyDriveImportHook.find_spec() not found; falling back to find_module()
<frozen importlib._bootstrap>:1047: ImportWarning: _BokehImportHook.find_spec() not found; falling back to find_module()
<frozen importlib._bootstrap>:1047: ImportWarning: _PyDriveImportHook.find_spec() not found; falling back to find_module()
<frozen importlib._bootstrap>:1047: ImportWarning: _BokehImportHook.find_spec() not found; falling back to find_module()
<frozen importlib._bootstrap>:1047: ImportWarning: _PyDriveImportHook.find_spec() not found; falling back to find_module()
<frozen importlib._bootstrap>:1047: ImportWarning: _BokehImportHook.find_spec() not found; falling back to find_module()
<frozen importlib._bootstrap>:1047: ImportWarning: _PyDriveImportHook.find_spec() not found; falling back to find_module()
<frozen importlib._bootstrap>:1047: ImportWarning: _BokehImportHook.find_spec() not found; falling back to find_module()
<frozen importlib._bootstrap>:1047: ImportWarning: _PyDriveImportHook.find_spec() not found; falling back to find_module()
<frozen importlib._bootstrap>:1047: ImportWarning: _BokehImportHook.find_spec() not found; falling back to find_module()
<frozen importlib._bootstrap>:1047: ImportWarning: _PyDriveImportHook.find_spec() not found; falling back to find_module()
<frozen importlib._bootstrap>:1047: ImportWarning: _BokehImportHook.find_spec() not found; falling back to find_module()
<frozen importlib._bootstrap>:1047: ImportWarning: _PyDriveImportHook.find_spec() not found; falling back to find_module()
<frozen importlib._bootstrap>:1047: ImportWarning: _BokehImportHook.find_spec() not found; falling back to find_module()
<frozen importlib._bootstrap>:1047: ImportWarning: _PyDriveImportHook.find_spec() not found; falling back to find_module()
<frozen importlib._bootstrap>:1047: ImportWarning: _BokehImportHook.find_spec() not found; falling back to find_module()
```

	Square_Footage	Num_Bedrooms	Num_Bathrooms	Year_Built	Lot_Size	Garage_Size	Neighborhood_Quality	House_Price
Square_Footage	1.00	-0.04	-0.03	-0.02	0.09	0.03	-0.01	0.99
Num_Bedrooms	-0.04	1.00	0.02	-0.02	-0.01	0.11	-0.05	0.01
Num_Bathrooms	-0.03	0.02	1.00	-0.02	0.03	0.02	0.02	-0.00
Year_Built	-0.02	-0.02	-0.02	1.00	-0.06	-0.03	-0.01	0.05
Lot_Size	0.09	-0.01	0.03	-0.06	1.00	0.00	0.04	0.16
Garage_Size	0.03	0.11	0.02	-0.03	0.00	1.00	-0.01	0.05
Neighborhood_Quality	-0.01	-0.05	0.02	-0.01	0.04	-0.01	1.00	-0.01
House_Price	0.99	0.01	-0.00	0.05	0.16	0.05	-0.01	1.00

Команды | Код | Текст

Подключиться повторно

Содержание

МГТУ им. Н.Э.Баумана | ИУС | 6 семестр | ТМО | ЛР№6

Загрузка датасета

Подготовка датасета

Разделение выборки

Обучение и оценка моделей

Оценка качества моделей

Модель группы стекинга

Модель многослойного перцептрона

МГУА

COMBI

MIA

Раздел

```
print('R²={}'.format(result))

# Точность на отдельных моделях
for model in [
    LinearRegression(),
    DecisionTreeRegressor(),
    RandomForestRegressor(n_estimators=50)
]:
    print(model)
    val_mae(model)
    val_mse(model)
    val_MedAE(model)
    val_r2(model)
    print("=====")
    print()

<frozen importlib._bootstrap>:1047: ImportWarning: _PyDriveImportHook.find_spec() not found; falling back to find_module()
<frozen importlib._bootstrap>:1047: ImportWarning: _BokehImportHook.find_spec() not found; falling back to find_module()
LinearRegression()
MAE=8111.297882965999
MSE=108162688.0895218
MedAE=6482.716831298565
R²=0.9982859969268084
=====
DecisionTreeRegressor()
MAE=25389.97571486141
MSE=1146136859.9739945
MedAE=21365.939845343635
R²=0.9822274832791457
=====
RandomForestRegressor(n_estimators=50)
MAE=18977.338746209814
MSE=597280114.8457341
MedAE=15287.529811232341
R²=0.989922117526043
=====

[ ] # Импортируем библиотеку heapy
# набор данных
dataset = Dataset(X_train, y_train, X_test)

# модели первого уровня
model_tree = Regressor(dataset=dataset, estimator=DecisionTreeRegressor, name='tree')
model_lr = Regressor(dataset=dataset, estimator=LinearRegression, parameters={'normalize': True}, name='lr')
model_rf = Regressor(dataset=dataset, estimator=RandomForestRegressor, parameters={'n_estimators': 50}, name='rf')

[ ] # Первый уровень: дерево и случайный лес
```

Команды | Код | Текст

Подключиться повторно

Содержание

МГТУ им. Н.Э.Баумана | ИУС | 6 семестр | ТМО | ЛР№6

Загрузка датасета

Подготовка датасета

Разделение выборки

Обучение и оценка моделей

Оценка качества моделей

Модель группы стекинга

Модель многослойного перцептрона

МГУА

COMBI

MIA

Раздел

```
[ ] print(f'R²: {r2_score(y_test, y_pred)}')
crit_name = "MSE" if squared else "RMSE"
print(f'{crit_name}: {mean_squared_error(y_test, y_pred)}')
print(f'MAE: {mean_absolute_error(y_test, y_pred)}')
```

COMBI

```
combi_model = gmdh.Combi()
combi_model.fit(X_train, y_train, verbose=1, n_jobs=-1, test_size=0.24, limit=0,
               criterion=gmdh.Criterion(gmdh.CriterionType.REGULARITY))

print()
print(combi_model.get_best_polynomial())
print()
y_pred = combi_model.predict(X_test)
print_metrics(y_test, y_pred)
```

LEVEL 1 [=====] 100% [00m:00s] (7 combinations) error=2.08163e+11
LEVEL 2 [=====] 100% [00m:00s] (21 combinations) error=1.4153e+11
LEVEL 3 [=====] 100% [00m:00s] (35 combinations) error=7.32315e+1
LEVEL 4 [=====] 100% [00m:00s] (35 combinations) error=2.77816e+1
LEVEL 5 [=====] 100% [00m:00s] (21 combinations) error=2.15683e+1
LEVEL 6 [=====] 100% [00m:00s] (7 combinations) error=1.86144e+1
LEVEL 7 [=====] 100% [00m:00s] (1 combinations) error=1.86181e+1

y = 199.7451*x1 + 10453.9621*x2 + 8274.5885*x3 + 15133.6826*x4 + 5133.7659*x5 - 988.5595*x7 - 195.9368

R²: 0.9985513697113126
RMSE: 88462238.99707367
MAE: 7548.291878568521

MIA

```
mia_model = gmdh.Mia()

mia_model.fit(X_train, y_train, verbose=1, n_jobs=-1, test_size=0.24, limit=0,
             criterion=gmdh.Criterion(gmdh.CriterionType.REGULARITY))

print()
print(mia_model.get_best_polynomial())
print()
y_pred = mia_model.predict(X_test)
print_metrics(y_test, y_pred)
```

LEVEL 1 [=====] 100% [00m:00s] (21 combinations) error=1.4577e+11
LEVEL 2 [=====] 100% [00m:00s] (3 combinations) error=1.09262e+11
LEVEL 3 [=====] 100% [00m:00s] (3 combinations) error=1.08636e+11
LEVEL 4 [=====] 100% [00m:00s] (3 combinations) error=1.0821e+11
LEVEL 5 [=====] 100% [00m:00s] (3 combinations) error=1.07984e+11
LEVEL 6 [=====] 100% [00m:00s] (3 combinations) error=1.07e+11

Содержание

МГТУ им. Н.Э.Баумана | ИУС | 6 семестр | ТМО | ЛР№6

Загрузка датасета

Подготовка датасета

Разделение выборки

Обучение и оценка моделей

Оценка качества моделей

Модель группы стекинга

Модель многослойного персептрона

МГУА

COMBI

MIA

+ Раздел

MIA

mia_model = gmdh.Mia()

```
mia_model.fit(X_train, y_train, verbose=1, n_jobs=-1, test_size=0.24, limit=0,
              criterion=gmdh.Criterion(gmdh.CriterionType.REGULARITY))
```

```
print()
print(mia_model.get_best_polynomial())
```

```
print()
y_pred = mia_model.predict(X_test)
```

```
print_metrics(y_test, y_pred)
```

```
LEVEL 1 [=====] 100% [00m:00s] (21 combinations) error=1.4577e+11
LEVEL 2 [=====] 100% [00m:00s] (3 combinations) error=1.09262e+11
LEVEL 3 [=====] 100% [00m:00s] (3 combinations) error=1.08636e+11
LEVEL 4 [=====] 100% [00m:00s] (3 combinations) error=1.0821e+11
LEVEL 5 [=====] 100% [00m:00s] (3 combinations) error=1.07984e+11
LEVEL 6 [=====] 100% [00m:00s] (3 combinations) error=1.07e+11
LEVEL 7 [=====] 100% [00m:00s] (3 combinations) error=1.07661e+11
```

```
f1_1 = 195.7217*x1 - 1381.1177*x7 + 0.8488*x1*x7 + 0.0005*x1^2 + 4.1399*x7^2 + 185260.1645
f1_2 = 199.5486*x1 + 11224.7911*x4 - 0.8857*x1*x4 + 0.0004*x1^2 + 1080.3525*x4^2 + 17734.0865
f1_3 = 198.2876*x1 + 14186.1839*x2 - 0.1645*x1*x2 + 0.0006*x1^2 - 495.9611*x2^2 + 20252.9199
```

```
f2_1 = 0.5483*f1_1 + 0.4554*f1_2 - 4.92316e-07*f1_1*f1_2 + 2.13824e-07*f1_1^2 + 2.78673e-07*f1_2^2 - 2211.9558
f2_2 = 0.6404*f1_1 + 0.3627*f1_3 - 3.09179e-06*f1_1*f1_3 + 1.52362e-06*f1_1^2 + 1.56768e-06*f1_3^2 - 2639.0489
f2_3 = 0.6093*f1_2 + 0.3936*f1_3 + 1.73745e-06*f1_2*f1_3 - 8.6425e-07*f1_2^2 - 8.73837e-07*f1_3^2 - 992.5241
```

```
f3_1 = 0.8304*f2_1 + 0.171*f2_2 + 1.61947e-05*f2_1*f2_2 - 8.06608e-06*f2_1^2 - 8.12945e-06*f2_2^2 + 534.597
f3_2 = 0.6157*f2_2 + 0.3862*f2_3 - 2.42765e-06*f2_2*f2_3 + 1.13739e-06*f2_2^2 + 1.28966e-06*f2_3^2 - 1278.654
f3_3 = 0.8897*f2_1 + 0.1186*f2_3 - 2.03349e-05*f2_1*f2_3 + 1.01501e-05*f2_1^2 + 1.01853e-05*f2_3^2 - 1818.6693
```

```
f4_1 = 0.8033*f3_1 + 0.1964*f3_2 - 3.85683e-05*f3_1*f3_2 + 1.93255e-05*f3_1^2 + 1.92432e-05*f3_2^2 - 910.3686
f4_2 = 0.2398*f3_2 + 0.7608*f3_3 + 4.93987e-06*f3_2*f3_3 - 2.52441e-06*f3_2^2 - 2.41583e-06*f3_3^2 - 44.9087
f4_3 = 0.1124*f3_1 + 0.8873*f3_3 + 5.67486e-05*f3_1*f3_3 - 2.81528e-05*f3_1^2 - 2.85957e-05*f3_3^2 + 331.0648
```

```
f5_1 = 0.1676*f4_1 + 0.8293*f4_3 + 0.0005*f4_1*f4_3 - 0.0002*f4_1^2 - 0.0002*f4_3^2 + 2292.284
f5_2 = - 0.4664*f4_1 + 1.4656*f4_2 + 0.0002*f4_1*f4_2 - 0.0001*f4_1^2 - 0.0001*f4_2^2 + 906.7248
```

```
y = - 0.2929*f5_1 + 1.2931*f5_2 + 5.672e-05*f5_1*f5_2 - 2.73929e-05*f5_1^2 - 2.93275e-05*f5_2^2 + 146.4503
```

```
R^2: 0.9915653888244382
```

```
RMSE: 515869824.5031711
```

```
MAE: 18327.725357127594
```

