Московский государственный технический университет им. Н.Э. Баумана

Факультет «Информатика и системы управления»
Кафедра ИУ5 «Системы обработки информации и управления»

Курс «Теория машинного обучения»

Отчет по рубежному контролю №2

Выполнил:
студент группы ИУ5-63Б
Ветошкин Артём
Подпись и дата:
14.06.22

Проверил:
Юрий Евгеньевич Гапанюк
Подпись и дата:

Москва, 2022 г.

Рубежный контроль №2

Загружаем библиотеки

```
import pandas as pd
from sklearn.tree import DecisionTreeRegressor
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score
from sklearn.preprocessing import LabelEncoder
from io import StringIO
from sklearn.tree import export_graphviz
from sklearn.ensemble import RandomForestRegressor
import pydotplus
from IPython.display import Image
import matplotlib.pyplot as plt
```

Загружаем данные

```
df = pd.read_csv('Admission_Predict.csv')
```

```
df
```

```
.dataframe tbody tr th {
    vertical-align: top;
}

.dataframe thead th {
    text-align: right;
}
```

	Serial No.	GRE Score	TOEFL Score	University Rating	SOP	LOR	CGPA	Research	Chance of Admit
0	1	337	118	4	4.5	4.5	9.65	1	0.92
1	2	324	107	4	4.0	4.5	8.87	1	0.76
2	3	316	104	3	3.0	3.5	8.00	1	0.72
3	4	322	110	3	3.5	2.5	8.67	1	0.80

	Serial No.	GRE Score	TOEFL Score	University Rating	SOP	LOR	CGPA	Research	Chance of Admit
4	5	314	103	2	2.0	3.0	8.21	0	0.65
...	...	...	...	...	...	...	...	...	...
395	396	324	110	3	3.5	3.5	9.04	1	0.82
396	397	325	107	3	3.0	3.5	9.11	1	0.84
397	398	330	116	4	5.0	4.5	9.45	1	0.91
398	399	312	103	3	3.5	4.0	8.78	0	0.67
399	400	333	117	4	5.0	4.0	9.66	1	0.95

400 rows × 9 columns

Разделим на обучающую и тестовую выборку

```
df_X_train, df_X_test, df_y_train, df_y_test = train_test_split(
    df.drop(columns='Chance of Admit '), df['Chance of Admit '], test_size=0.2,
random_state=171)
```

Дерево решений

```
tree = DecisionTreeRegressor()
tree.fit(df_X_train, df_y_train)
```

```
DecisionTreeRegressor()
```

```
tree_predict = tree.predict(df_X_test)
```

Случаный лес

```
forest = RandomForestRegressor()
forest.fit(df_X_train, df_y_train)
```

```
RandomForestRegressor()
```

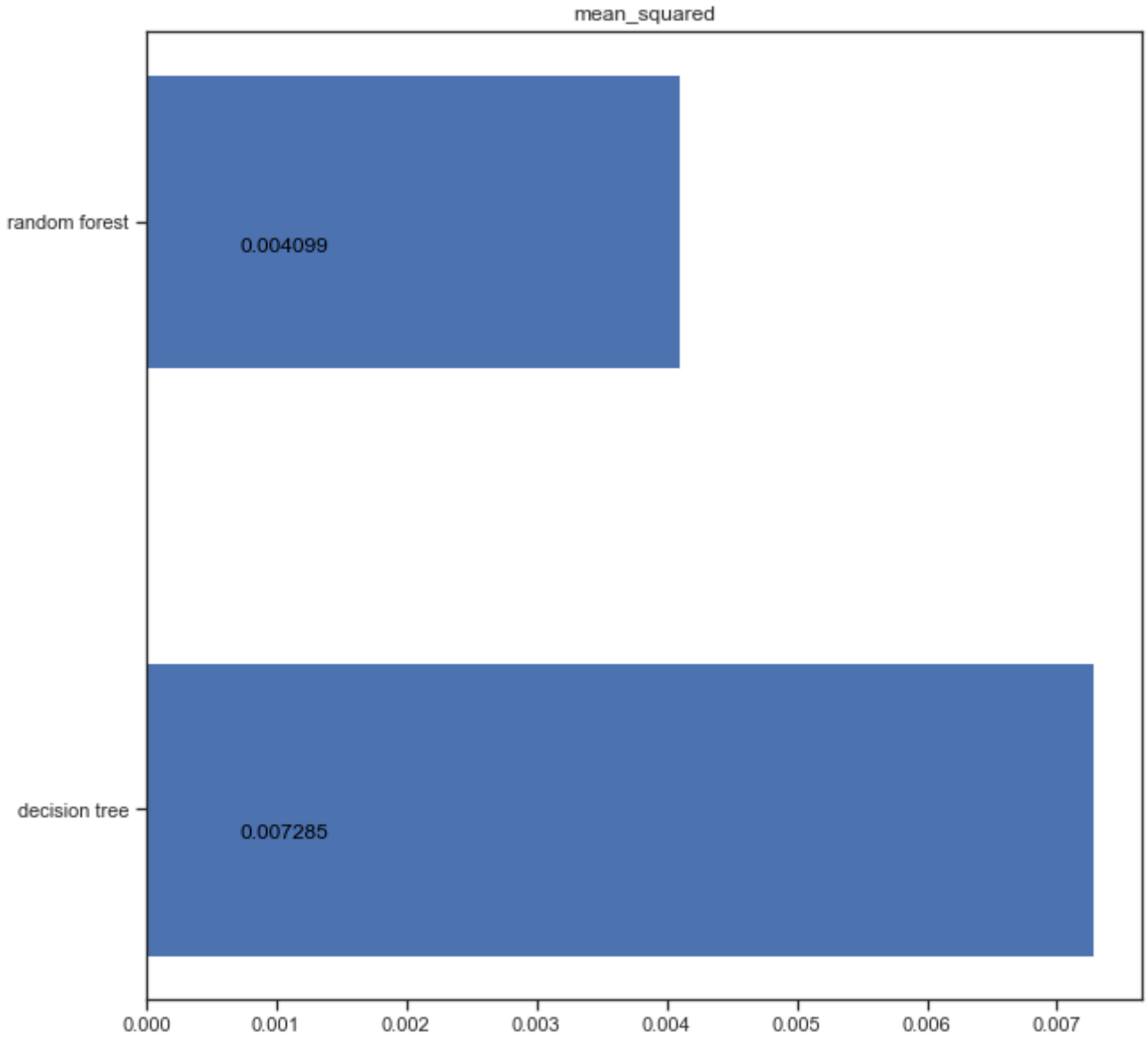
```
forest_predict = forest.predict(df_X_test)
```

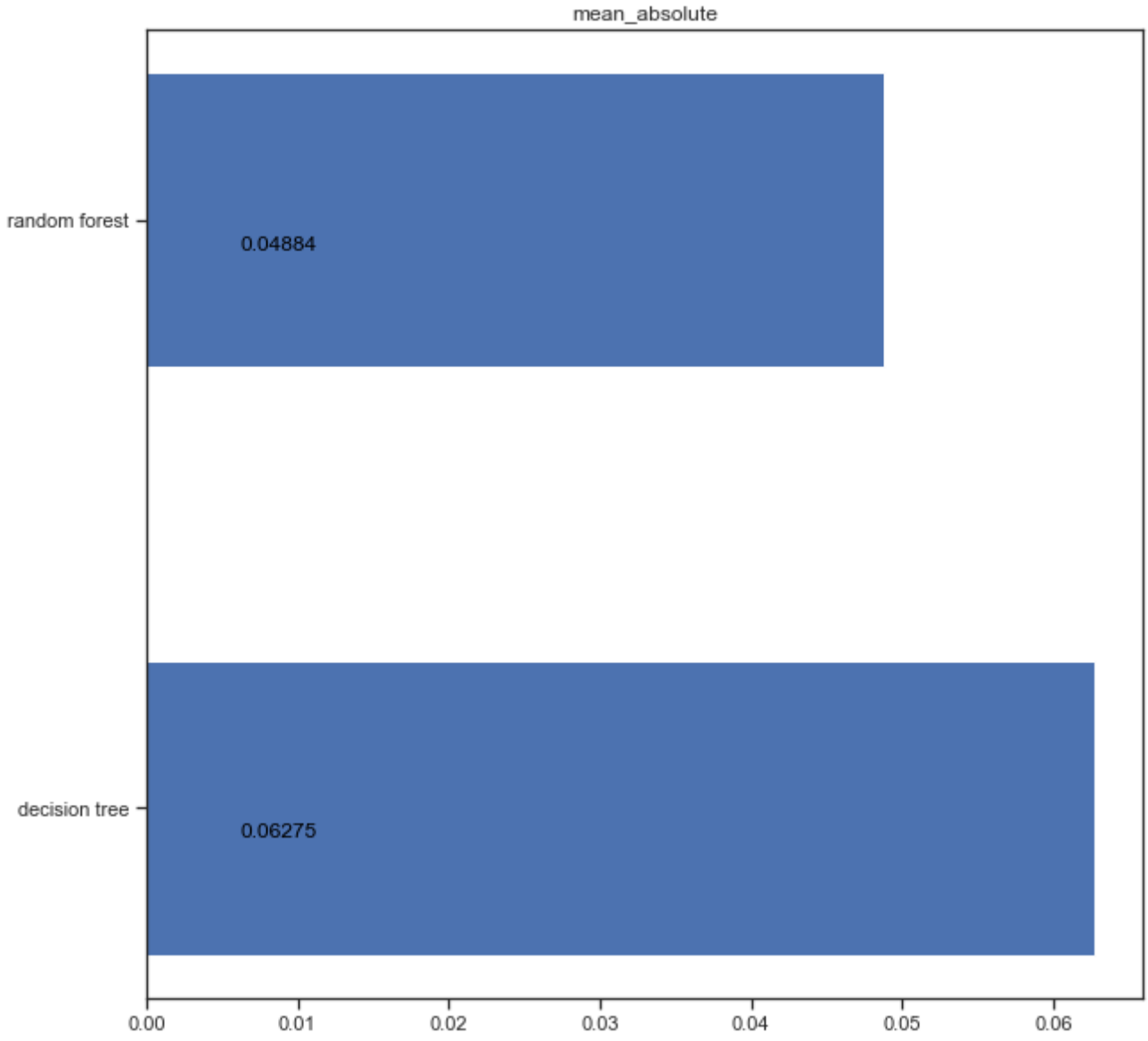
Оценка моделей

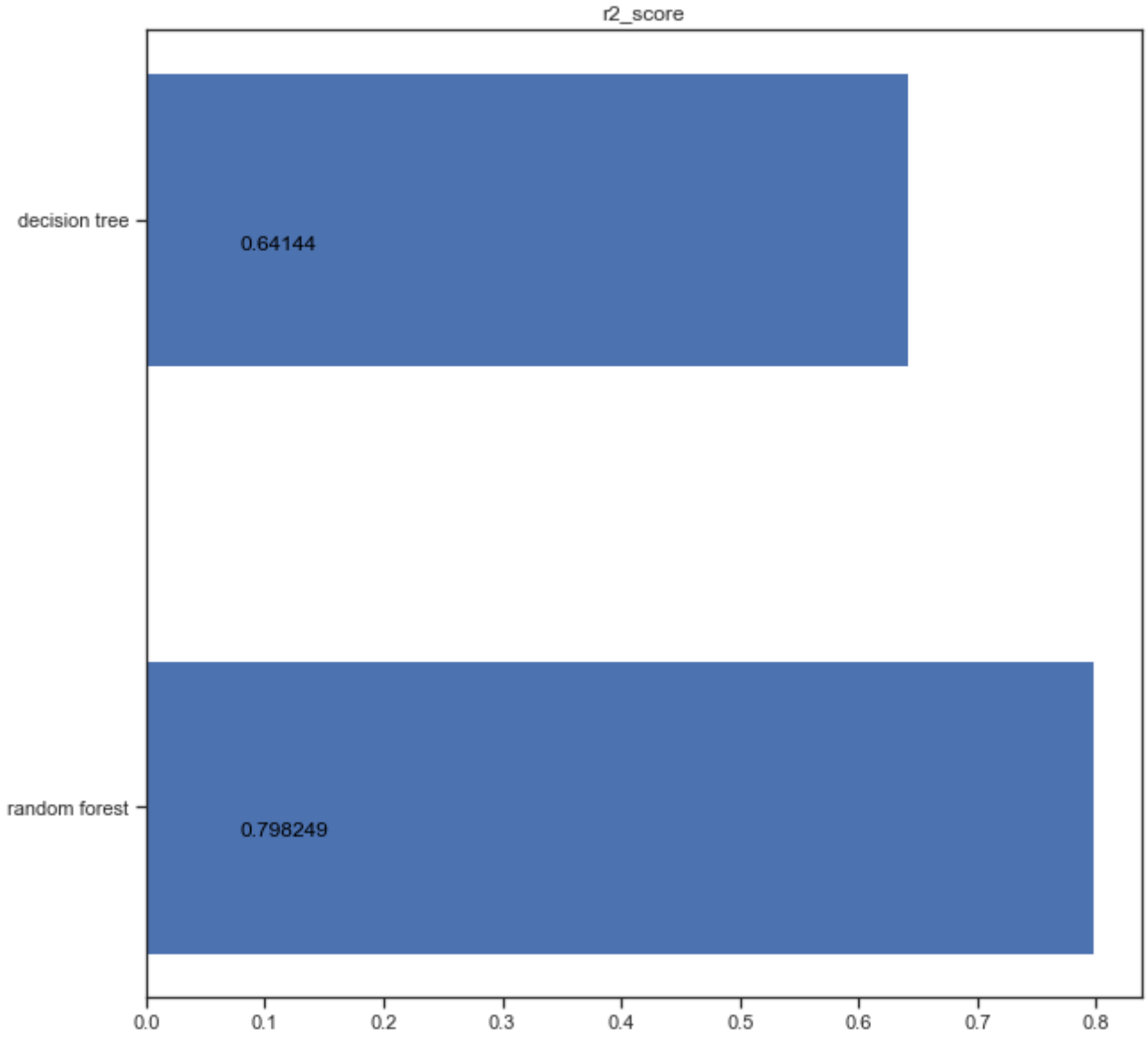
Для оценки будем использовать три метрики: **Средняя квадратичная ошибка**, **Средняя абсолютная ошибка**, **R2 score**.

```
def plot_metrics(metrics, models, test_y):
    for name, fun in metrics.items():
        fig, ax = plt.subplots(figsize=(10,10))
        results_metrics = []

        for nm, results in models.items():
            results_metrics.append(fun(test_y, results))

        sorted_el = list(sorted(list(zip(models.keys(), results_metrics)),
key=lambda x: -x[1]))
        results_metrics = list(map(lambda x: x[1], sorted_el))
        model_list = list(map(lambda x: x[0], sorted_el))

        pos = np.arange(len(model_list))
        rects = ax.barh(pos, results_metrics,
                    align='center',
                    height=0.5,
                    tick_label=model_list)
        ax.set_title(name)
        for a, b in zip(pos, results_metrics):
            plt.text(max(results_metrics) * 0.1, a-0.05, str(round(b,6)),
color='black')
        plt.show()
```

```
metrics = {
    'mean_squared':mean_squared_error,
    'mean_absolute':mean_absolute_error,
    'r2_score':r2_score
}

predictions = {
    'decision tree': tree_predict,
    'random forest': forest_predict
}

plot_metrics(metrics, predictions, df_y_test)
```


mean_squared
random forest
0.004099
decision tree
0.007285
0.000
0.001
0.002
0.003
0.004
0.005
0.006
0.007


mean_absolute
random forest
0.04884
decision tree
0.06275
0.00
0.01
0.02
0.03
0.04
0.05
0.06



Случаный лес лучше работает на данном наборе данных, что логично т.к. она явялется ансамблей моделей.