

Московский государственный технический университет имени Н. Э. Баумана

Факультет «Информатика и системы управления»

Кафедра ИУ5

Отчёт по

лабораторной работе № 6

«Технологии машинного обучения»

Подготовил:

Кан Андрей Дмитриевич

Группа ИУ5-64Б

Подпись\_\_\_\_\_

Дата\_\_\_\_\_

Москва  
2021г.

**Цель лабораторной работы: изучение возможностей демонстрации моделей машинного обучения с помощью веб-приложений.**

**Задание:**

Разработайте макет веб-приложения, предназначенного для анализа данных.

Вариант 1. Макет должен быть реализован для одной модели машинного обучения. Макет должен позволять:

- задавать гиперпараметры алгоритма,
- производить обучение,
- осуществлять просмотр результатов обучения, в том числе в виде графиков.

Вариант 2. Макет должен быть реализован для нескольких моделей машинного обучения. Макет должен позволять:

- выбирать модели для обучения,
- производить обучение,
- осуществлять просмотр результатов обучения, в том числе в виде графиков.

Для разработки рекомендуется использовать следующие (или аналогичные) фреймворки:

- [streamlit](#)
- [gradio](#)
- [dash](#)

**Текст программы:**

```
from sklearn.datasets import *

import streamlit as st

import seaborn as sns

import pandas as pd

import numpy as np

from sklearn.preprocessing import MinMaxScaler

from sklearn.model_selection import cross_val_score
```

```
from sklearn.neighbors import KNeighborsRegressor, KNeighborsClassifier

import matplotlib.pyplot as plt


@st.cache

def load_data():

    """

    Загрузка данных

    """

    wine = load_wine()

    # Сформируем DataFrame

    wine_df = pd.DataFrame(data=np.c_[wine['data']],

                           columns=wine['feature_names'])

    sc = MinMaxScaler()

    wine_sc = sc.fit_transform(wine.data)

    return wine_sc, wine.target, wine_df.shape[0], wine_df


st.header('Обучение модели ближайших соседей')


data_load_state = st.text('Загрузка данных...')

data_X, data_Y, data_len, data = load_data()

data_load_state.text('Данные загружены!')
```

```
st.write(data.head())
```

```
cv_slider = st.slider('Количество фолдов:', min_value=3, max_value=10, value=5, step=1)
```

```
#Вычислим количество возможных ближайших соседей
```

```
rows_in_one_fold = int(data_len / cv_slider)
```

```
allowed_knn = int(rows_in_one_fold * (cv_slider-1))
```

```
st.write('Количество строк в наборе данных - {}'.format(data_len))
```

```
st.write('Максимальное допустимое количество ближайших соседей с учетом выбранного  
количества фолдов - {}'.format(allowed_knn))
```

```
cv_knn = st.slider('Количество ближайших соседей:', min_value=1, max_value=allowed_knn,  
value=5, step=1)
```

```
scores = cross_val_score(KNeighborsClassifier(n_neighbors=cv_knn),  
data_X, data_Y, scoring='accuracy', cv=cv_slider)
```

```
st.subheader('Оценка качества модели')
```

```
st.write('Значения ассурасу для отдельных фолдов')
```

```
st.bar_chart(scores)
```

```
st.write('Усредненное значение ассурасу по всем фолдам - {}'.format(np.mean(scores)))
```

## Экранные формы:

### Обучение модели ближайших соседей

Данные загружены!

	alcohol	malic_acid	ash	alcalinity_of_ash	magnesium	total_phenols
0	14.2300	1.7100	2.4300	15.6000	127	2.8000
1	13.2000	1.7800	2.1400	11.2000	100	2.6500
2	13.1600	2.3600	2.6700	18.6000	101	2.8000
3	14.3700	1.9500	2.5000	16.8000	113	3.8500
4	13.2400	2.5900	2.8700	21	118	2.8000

Количество фолдов:



Количество строк в наборе данных - 178

Максимальное допустимое количество ближайших соседей с учетом выбранного количества фолдов - 140

Количество ближайших соседей:



### Оценка качества модели

Значения ассигасу для отдельных фолдов



Усредненное значение ассигасу по всем фолдам - 0.9552380952380952