ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ «НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ «ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»

*Московский институт электроники и математики им. А.Н. Тихонова*

# Руководство разработчика по работе с приложением: «Определение токсичности грибов на основе морфологических признаков с использованием машинного обучения».

Разработчики:

Шмелёв Антон

Ро Александр

Чапайкин Арсений

Андреев Александр

Руководитель:

Полякова М. В.

Москва 2020 г.

## Требования к характеристикам компьютера и операционной системе

Наличие интерпретатора Python3.11+.

Операционная система – Windows 7, 8, 10, или MacOS, или Linux.

## Версии интерпретатора и используемых библиотек

Интерпретатор – Python 3.11+.

Используемые библиотеки:

|  |  |
| --- | --- |
| aiofiles | 24.1.0 |
| annotated-types | 0.7.0 |
| anyio | 4.9.0 |
| audioop-lts | 0.2.1 |
| catboost | 1.2.8 |
| certifi | 2025.4.26 |
| charset-normalizer | 3.4.2 |
| click | 8.1.8 |
| contourpy | 1.3.2 |
| cycler | 0.12.1 |
| fastapi | 0.115.12 |
| ffmpy | 0.5.0 |
| filelock | 3.18.0 |
| fonttools | 4.58.0 |
| fsspec | 2025.5.0 |
| gradio | 5.30.0 |
| gradio\_client | 1.10.1 |
| graphviz | 0.20.3 |
| groovy | 0.1.2 |
| h11 | 0.16.0 |
| httpcore | 1.0.9 |
| httpx | 0.28.1 |
| huggingface-hub | 0.31.4 |
| idna | 3.10 |
| Jinja2 | 3.1.6 |
| joblib | 1.5.0 |
| kiwisolver | 1.4.8 |
| markdown-it-py | 3.0.0 |
| MarkupSafe | 3.0.2 |
| matplotlib | 3.10.3 |
| mdurl | 0.1.2 |
| narwhals | 1.39.1 |
| numpy | 2.2.6 |
| orjson | 3.10.18 |
| packaging | 25.0 |
| pandas | 2.2.3 |
| pillow | 11.2.1 |
| plotly | 6.1.0 |
| pydantic | 2.11.4 |
| pydantic\_core | 2.33.2 |
| pydub | 0.25.1 |
| Pygments | 2.19.1 |
| pyparsing | 3.2.3 |
| python-dateutil | 2.9.0.post0 |
| python-multipart | 0.0.20 |
| pytz | 2025.2 |
| PyYAML | 6.0.2 |
| requests | 2.32.3 |
| rich | 14.0.0 |
| ruff | 0.11.10 |
| safehttpx | 0.1.6 |
| scikit-learn | 1.6.1 |
| scipy | 1.15.3 |
| seaborn | 0.13.2 |
| semantic-version | 2.10.0 |
| shellingham | 1.5.4 |
| six | 1.17.0 |
| sniffio | 1.3.1 |
| starlette | 0.46.2 |
| threadpoolctl | 3.6.0 |
| tomlkit | 0.13.2 |
| tqdm | 4.67.1 |
| typer | 0.15.4 |
| typing-inspection | 0.4.1 |
| typing\_extensions | 4.13.2 |
| tzdata | 2025.2 |
| urllib3 | 2.4.0 |
| uvicorn | 0.34.2 |
| websockets | 15.0.1 |

# Инструкция по установке приложения

1. *Перейдите в папку work и создайте виртуальное окружение*

cd work

python -m venv .venv

1. *Активируйте виртуальное окружение:*

Windows:

.venv\Scripts\activate

Linux/macOS:

source .venv/bin/activate

1. *Установите зависимости:*

pip install -r requirements.txt

## Инструкция по запуску и настройке приложения

Команда для запуска: python scripts/main\_interface.py

## Описание структуры БД

|  |  |  |  |
| --- | --- | --- | --- |
| cap-color | Feature | Categorical | Цвет шляпки |
| does-bruise-or-bleed | Feature | Categorical | Выделяет сок или изменяет цвет при повреждении |
| gill-attachment | Feature | Categorical | Тип крепления пластинок к ножке |
| gill-spacing | Feature | Categorical | Расстояние между пластинками |
| gill-color | Feature | Categorical | Цвет пластинок |
| stem-height | Feature | Continuous | Высота ножки |
| stem-width | Feature | Continuous | Ширина ножки |
| stem-root | Feature | Categorical | Форма ножки |
| stem-surface | Feature | Categorical | Поверхность ножки |
| stem-color | Feature | Categorical | Цвет ножки |
| veil-type | Feature | Categorical | Тип покрывала |
| veil-color | Feature | Categorical | Цвет покрывала |
| has-ring | Feature | Categorical | Наличие кольца |
| ring-type | Feature | Categorical | Тип кольца |
| spore-print-color | Feature | Categorical | Цвет спорового отпечатка |
| habitat | Feature | Categorical | Место обитания |
| season | Feature | Categorical | Сезон произрастания |

## Структура каталогов

|  |  |  |
| --- | --- | --- |
| **Первый уровень** | **Второй уровень** | **Объяснение** |
| Work |  | Основной каталог с подкаталогами |
|  | Data | Каталог с датасетом и пояснительнфм файлом к датасету |
|  | Graphics | Католог для сохранения графиков |
|  | Images | Каталог с картинками |
|  | Library | Каталог для хранения необходимых библиотек |
|  | Notes | Каталог с логами |
|  | Output | Директория для сохранения в нее данных |
|  | Resources | Каталог с пояснительыми картинками к признакам грибов |
|  | Scripts | Каталог со всеми скриптами |

## Архитектура приложения

|  |  |  |
| --- | --- | --- |
| **Модуль** | **Местонахождение** | **Функция** |
| data\_prcessing.py | Work/Scripts | Код препроцессора. |
| main.py | Work/Scripts | Предобработка данных и построение отчетов. |
| main\_interface.py | Work/Scripts | Код интерфейса программы и основной логики. Entry point. |

## Листинг основного скрипта и всех модулей.

|  |  |
| --- | --- |
| **Модуль** | **Функции с докстрингами** |
| data\_processing.py  """  Модуль с кодом препроцессора  """ | # Автор: Шмелев Антон  class DataPreprocessor(TransformerMixin, BaseEstimator):  """  Класс препроцессора  """ |
| main.py  """  Модуль для предобработки данных и построения отчетов  """ | # Автор: Шмелев Антон  def configure\_guides(path):  """  Конфигурирует справочники из MS Excel файла, также записывает их в pickle-файлы.  Args:  path (str): Путь к Excel файлу и к тому, куда будут сохранены pickle-файлы  Returns:  Dict[str, pd.DataFrame]: Словарь, включающий в себя:  - key (str): Название признака  - value (pd.DataFrame): Соответствующий справочник  """  # Автор: Чапайкин Арсений  def count\_na\_percentage(dataframe, feature):  """  Подсчитывает процент NaN значений от общего числа в колонке feature датафрейма df.  Args:  dataframe (pd.DataFrame): Рассматриваемый датафрейм  feature (str): Название признака  Returns:  int: Процент NaN значений от общего числа  """  # Автор: Ро Александр  def feature\_class\_correlation(dataframe, feature):  """  Считает количество встречаемых значений указанного  признака в разбивке по классам (poisonous/edible).  Args:  dataframe (pd.DataFrame): Рассматриваемый датафрейм  feature (str): Название признака  Returns:  pd.Series: Series с мультииндексом в котором:  - 1-й уровень: уникальные значения признака  - 2-й уровень: метки классов (e - съедобный, p - ядовитый)  - значения: количество соответствующих наблюдений  """  # Автор: Ро Александр  def feature\_mean\_cap\_diameter(dataframe, feature\_1, feature\_2, feature\_3):  """  Строит сводную таблицу со средним диаметром шляпки  гриба по комбинациям трёх выбранных категориальных признаков.  Args:  dataframe (pd.DataFrame): Рассматриваемый датафрейм  feature\_1 (str): Название первого признака  feature\_2 (str): Название второго признака  feature\_3 (str): Название третьего признака  Returns:  pd.DataFrame: Сводная таблица по исходным данным  """  # Автор: Ро Александр  def class\_ranged\_by\_stem\_height(dataframe, begin, end):  """  Отбирает грибы у которых значение высоты  ножки лежит в заданном диапазоне и возвращает серию с их классами.  Args:  dataframe (pd.DataFrame): Рассматриваемый датафрейм  begin (int): Нижняя граница рассматриваемого диапазона  end (int): Верхняя граница рассматриваемого диапазона  Returns:  pd.Series: Серия с классами грибов, удовлетворяющих условию  """  # Автор: Андреев Александр  def cap\_diams\_stem\_heights(dataframe, width\_begin, width\_end, season):  """  Отбирает грибы по конкретному сезону произрастания у которых  значение ширины ножки лежит в заданном диапазоне  и возвращает серию с их диаметрами шляпки и высотами ножки.  Args:  dataframe (pd.DataFrame): Рассматриваемый датафрейм  width\_begin (int): Нижняя граница рассматриваемого диапазона  width\_end (int): Верхняя граница рассматриваемого диапазона  season (str): Сезон в формате:  - a - осень  - w - зима  - u - лето  - s - весна  Returns:  pd.Series: Серия с диаметрами шляпки и  высотами ножки грибов, удовлетворяющих условию  """  # Автор: Андреев Александр  def class\_boxplot(dataframe, numeric\_feature):  """  Строит boxplot (ящик с усами) для числового признака,  сгруппированного по классам грибов (poisonous/edible).  Args:  dataframe (pd.DataFrame): Рассматриваемый датафрейм  numeric\_feature (str): Название признака  Returns:  matplotlib.axes.Axes: Объект Axes с построенным boxplot  """  # Автор: Андреев Александр  def cap\_diameter\_histplot(dataframe, hue):  """  Строит гистограмму распределения диаметров шляпки  грибов с разделением по категориальному признаку.  Args:  dataframe (pd.DataFrame): Рассматриваемый датафрейм  hue (str): Название категориального признака  Returns:  matplotlib.axes.Axes: Объект Axes с построенной гистограммой  """  # Автор: Чапайкин Арсений  def stem\_height\_scatterplot(dataframe, numeric\_feature, hue):  """  Строит точечный график зависимости между высотой ножки  гриба и заданным числовым признаком с разделением по некоторому категориальному признаку.  Args:  dataframe (pd.DataFrame): Рассматриваемый датафрейм  numeric\_feature (str): Название числового признака  hue (str): Название категориального признака  Returns:  matplotlib.axes.Axes: Объект Axes с построенным точечным графиком  """  # Автор: Чапайкин Арсений  def stem\_width\_boxplot(dataframe, object\_feature):  """  Строит boxplot (ящик с усами) для диаметра шляпки,  сгруппированного по заданному категориальному признаку.  Args:  dataframe (pd.DataFrame): Рассматриваемый датафрейм  object\_feature (str): Название категориального признака  Returns:  matplotlib.axes.Axes: Объект Axes с построенным boxplot  """ |
| main\_interface.py  """  Модуль с интерфейсом и основной логикой программы  """ | # Автор: Ро Александр  def fetch\_parameters():  """  Загружает параметры из JSON-файла и форматирует их в список словарей.  Returns:  list: Список параметров, каждый из которых представлен словарём  с ключами: 'name', 'type', 'values', 'image', 'prerequisites'.  """  # Автор: Шмелев Антон  def create\_component(param):  """  Создаёт UI-компонент Gradio на основе типа параметра.  Args:  param (dict): Словарь параметра с ключами 'name', 'type', 'values', 'image'.  Returns:  tuple: Кортеж из изображения (gr.Image или None) и соответствующего компонента Gradio.  Raises:  KeyError: Если указан неизвестный тип компонента.  """  # Автор: Ро Александр  def setup\_visibility(components, connections):  """  Настраивает динамическую видимость компонентов интерфейса на основе связей.  Args:  components (dict): Словарь компонентов, где ключ — имя параметра,  значение — кортеж (изображение, компонент).  connections (list): Список связей вида (parent\_name, [child\_names]),  определяющих, какие компоненты должны появляться при выборе.  """  # pylint: disable=too-many-locals  # pylint: disable=too-many-statements  # Авторы: Чапайкин Арсений, Андреев Александр  def head():  """  Главная функция отображения интерфейса  """ |