

МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ, СВЯЗИ И МАССОВЫХ  
КОММУНИКАЦИЙ РОССИЙСКОЙ ФЕДЕРАЦИИ  
Ордена Трудового Красного Знамени федеральное государственное бюджетное  
образовательное учреждение высшего образования  
**«Московский технический университет связи и информатики» (МТУСИ)**  
Кафедра «Информационные системы и сетевые технологии»

# **Презентация по защите практике системы искусственного интеллекта**

По теме:

**«Настраиваемая, трехдиапазонная, звуковая сигнализация в  
зависимости от веса объекта»**

Подготовил студент группы УБСТ2304  
Паокин А.И.

руководитель практики  
Кандзюба Е.В.

Москва 2025

# Цели и задачи практики

**Цель проекта:** Автоматизация процессов взвешивания транспортных средств с использованием ИИ и машинного обучения.

## Задачи:

- Расширение базы данных для классификации транспорта.
- Создание модели машинного обучения.
- Интеграция модели в WPF-приложение.

**Результат:** Повышение точности учета, снижение ошибок, улучшение аналитики.



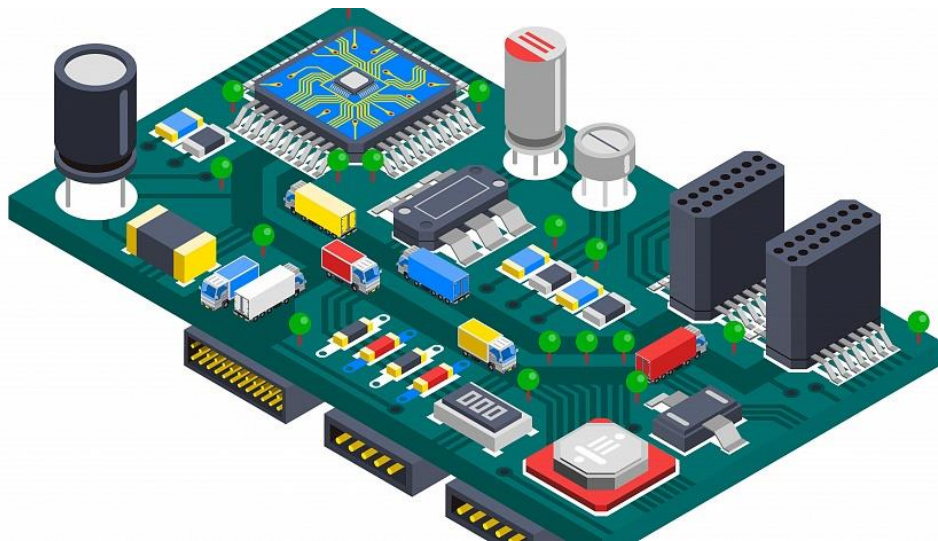
# Техника безопасности

## Основные требования:

- Проверка оборудования перед началом работы.
- Запрет на самостоятельный ремонт.
- Ограничение непрерывной работы (не более 2 часов).

## Рекомендации:

Регулярные перерывы и упражнения для глаз.



# Расширение базы данных

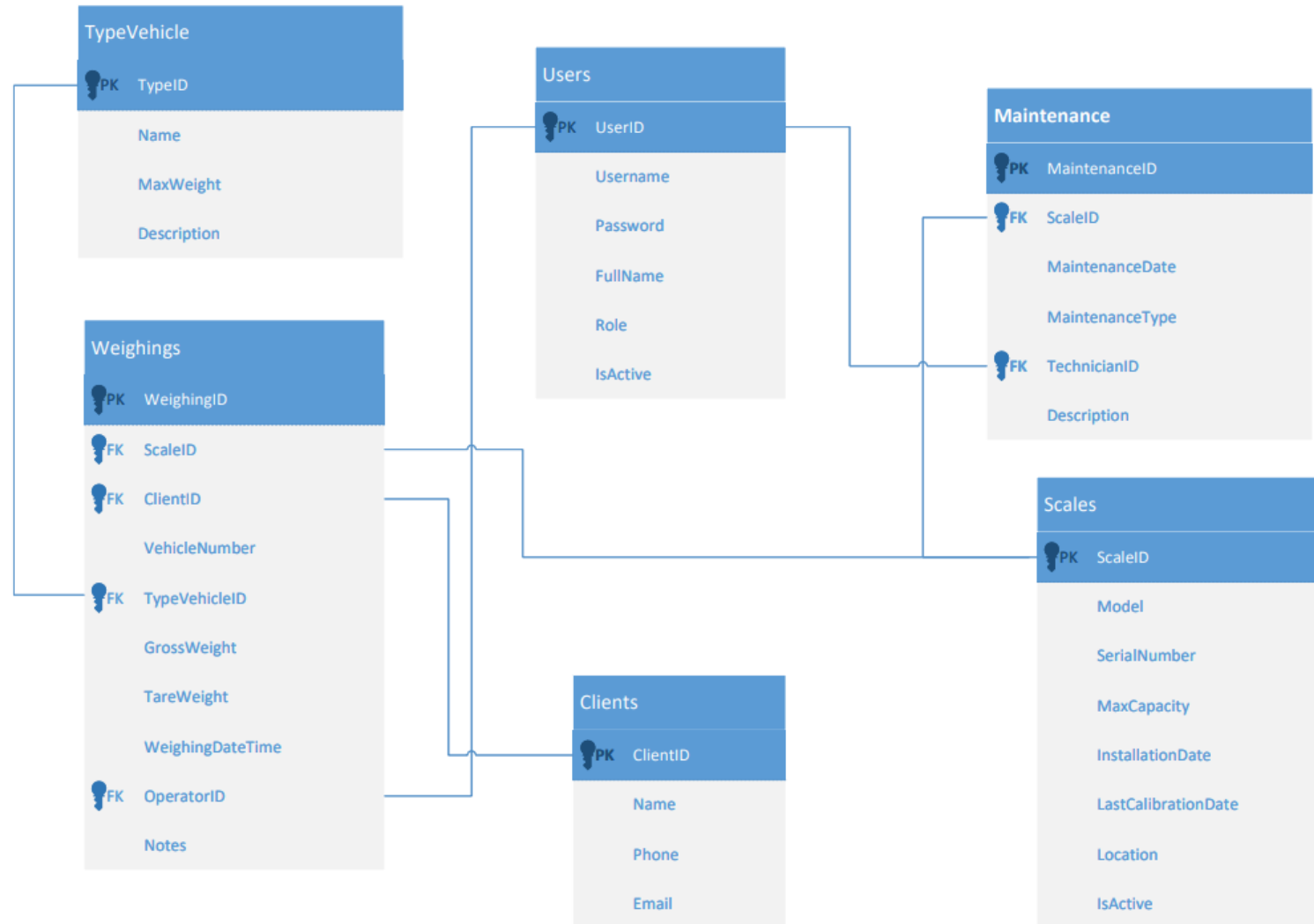
## Новые сущности:

- Таблица TypeVehicle (тип транспорта, максимальный вес).
- Связь с таблицей Weighings (внешний ключ TypeVehicleID).

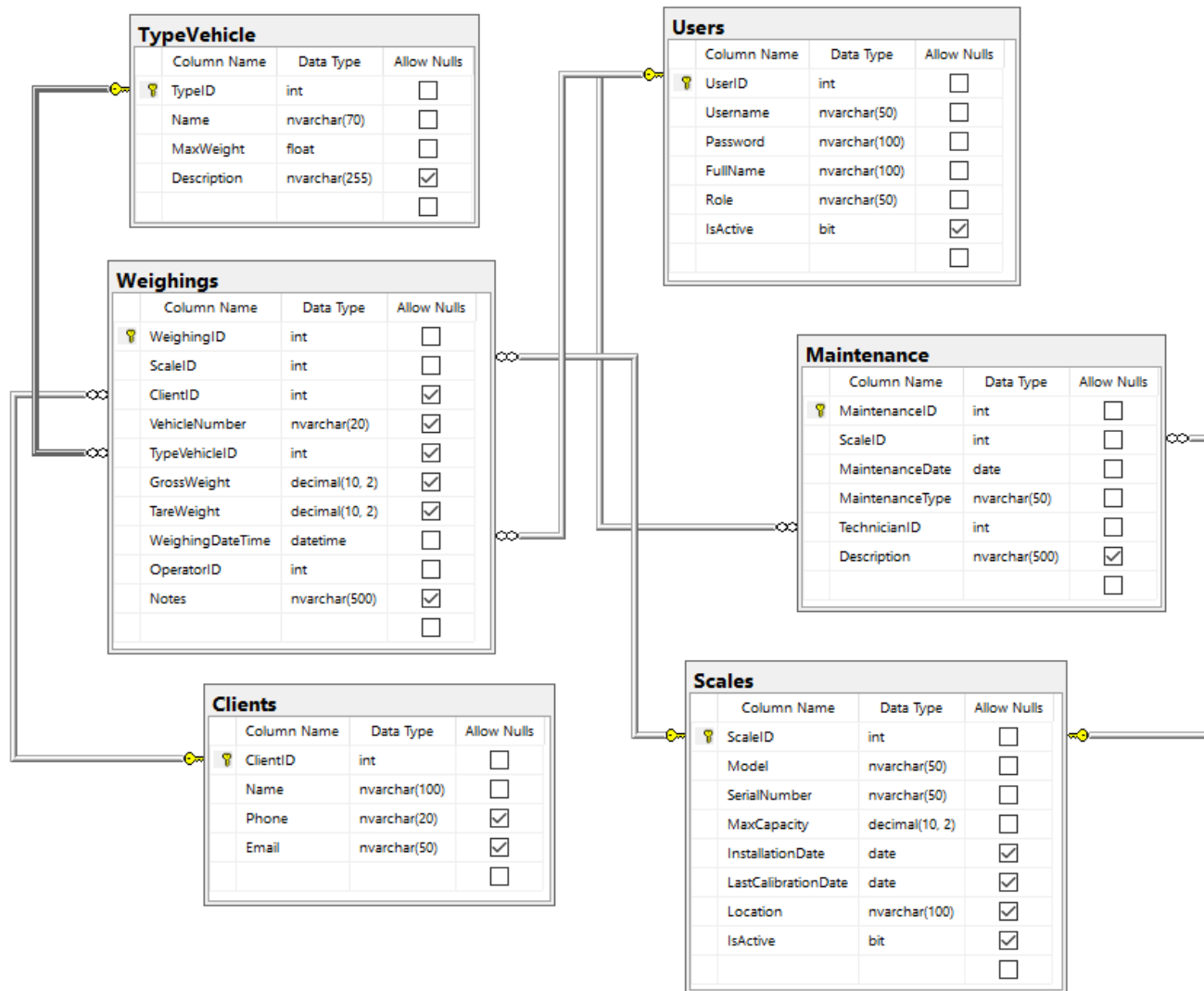
## Преимущества:

- Детализация данных.
- Учет характеристик транспорта.
- Обратная совместимость.

# Диаграмма «Сущность-связь»



# Схема данных



# Запрос для вывода данных

```
Select
    [TypeVehicleID]
    ,TV.[Name]
    ,TV.[MaxWeight]
    ,TV.[Description]
    ,[GrossWeight]
    ,[TareWeight]
from dbo.Weighings Wei
JOIN
    dbo.TypeVehicle TV ON Wei.TypeVehicleID = TV.TypeID
```

# Выбор данных для обучения модели

TypeVehicleID	Name	MaxWeight	Description	GrossWeight	TareWeight
1	Легковой автомобиль	3000	Масса до 3 тонн	1200.00	800.00
1	Легковой автомобиль	3000	Масса до 3 тонн	1800.00	1100.00
1	Легковой автомобиль	3000	Масса до 3 тонн	2200.00	1400.00
1	Легковой автомобиль	3000	Масса до 3 тонн	1900.00	1200.00
1	Легковой автомобиль	3000	Масса до 3 тонн	2500.00	1600.00
1	Легковой автомобиль	3000	Масса до 3 тонн	2700.00	1500.00
1	Легковой автомобиль	3000	Масса до 3 тонн	2300.00	1400.00
1	Легковой автомобиль	3000	Масса до 3 тонн	2100.00	1300.00
1	Легковой автомобиль	3000	Масса до 3 тонн	2400.00	1500.00
1	Легковой автомобиль	3000	Масса до 3 тонн	2600.00	1600.00
1	Легковой автомобиль	3000	Масса до 3 тонн	2800.00	1700.00
1	Легковой автомобиль	3000	Масса до 3 тонн	2900.00	1800.00
1	Легковой автомобиль	3000	Масса до 3 тонн	2000.00	1200.00
1	Легковой автомобиль	3000	Масса до 3 тонн	2200.00	1400.00
1	Легковой автомобиль	3000	Масса до 3 тонн	2400.00	1500.00
1	Легковой автомобиль	3000	Масса до 3 тонн	2600.00	1600.00
1	Легковой автомобиль	3000	Масса до 3 тонн	2800.00	1700.00



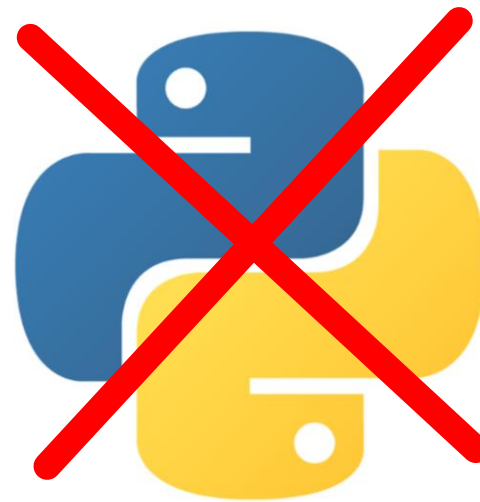
# Выбор инструментов

## ML.NET:

- Прямая интеграция с WPF.
- Высокая производительность.
- Простота развертывания.

## Преимущества перед Python:

- Отсутствие API-серверов.
- Нативная работа в .NET.



# Метод обучения и выгрузки модели

```
Ссылка 1
static void TrainAndSaveModel()
{
    Console.WriteLine("Режим обучения модели...");

    var data = LoadDataFromExcel("SQLData.xlsx");

    var mlContext = new MLContext();

    // Преобразование данных в IDataView
    var dataView = mlContext.Data.LoadFromEnumerable(data);

    // Конвейер обработки данных (упрощенный, без NetWeight в Features)
    var pipeline = mlContext.Transforms
        .Conversion.MapValueToKey(outputColumnName: "Label", inputColumnName: "TypeVehicleID")
        .Append(mlContext.Transforms.Concatenate(
            outputColumnName: "Features",
            nameof(VehicleData.GrossWeight),
            nameof(VehicleData.TareWeight)
        ))
        .Append(mlContext.MulticlassClassification.Trainers.LightGbm())
        .Append(mlContext.Transforms.Conversion.MapKeyToValue(outputColumnName: "PredictedLabel"));

    // Обучение модели
    Console.WriteLine("Обучение модели...");
    var model = pipeline.Fit(dataView);

    // Сохранение модели
    mlContext.Model.Save(model, dataView.Schema, "vehicle_classifier_mlnet.zip");
    Console.WriteLine("Модель сохранена в vehicle_classifier_mlnet.zip");

    // Тестирование сразу после обучения
    TestModel(mlContext, model);
}
```

# Внедрение модели в приложение

Основу реализации составляют три ключевых компонента. Класс `VehicleData` содержит структуру данных для передачи параметров в модель, включая вычисляемое свойство `NetWeight` (чистый вес), а атрибуты `LoadColumn` обеспечивают корректное сопоставление данных. Класс `VehiclePrediction` определяет структуру выходных данных модели, где свойство `PredictedLabel` хранит предсказанный идентификатор типа транспортного средства. В WPF-приложении (`WindowAddEditWeighing`) модель загружается при инициализации окна, а `PredictionEngine` используется для выполнения предсказаний при сохранении данных.

# Примеры программного кода

```
private readonly BridgeScalesContext _context;  
private PredictionEngine<VehicleData, VehiclePrediction> predictionEngine;  
Ссылка: 10  
private Weighing _weighing { get; set; }  
Ссылка: 6  
private Scale _scaling { get; set; }  
Ссылка: 4  
private SensorData _sensor { get; set; }  
  
Ссылка: 1  
public WindowAddEditWeighing(BridgeScalesContext context)  
{  
    _context = context;  
    // Загрузка модели  
    var mlContext = new MLContext();  
    var model = mlContext.Model.Load("vehicle_classifier_mlnet.zip", out var schema);  
    predictionEngine = mlContext.Model.CreatePredictionEngine<VehicleData, VehiclePrediction>(model);  
  
    _weighing = new Weighing { WeighingDateTime = DateTime.Now };  
    DataContext = this;  
    InitializeComponent();  
    LoadComboBoxData();  
    _scaling = UpdateScaleData();  
}
```

```
_weighing.TypeVehicleId =  
(int)predictionEngine.Predict(new VehicleData(_sensor.weight, tareweight)).PredictedLabel;
```

# Заключение

В ходе работы была успешно разработана и внедрена система настраиваемой звуковой сигнализации, активируемой в зависимости от веса транспортного средства. Проект включал расширение базы данных для классификации транспорта, создание модели машинного обучения на основе ML.NET и её интеграцию в WPF-приложение. Реализованное решение значительно повысило точность учета, автоматизировало процесс классификации и сократило влияние человеческого фактора. Полученные результаты демонстрируют эффективность применения технологий искусственного интеллекта в системах контроля и учета, а также открывают перспективы для дальнейшего развития функционала, включая анализ перегруза и прогнозирование нагрузок. Работа подтвердила практическую ценность использования машинного обучения для решения производственных задач.

# Ссылки на материалы и проект

**GitHub проекта:**

[github.com/Fiery-Paks/Praktika\\_Paokin\\_UBST\\_2304](https://github.com/Fiery-Paks/Praktika_Paokin_UBST_2304)

**Видео-отчёт:**

[https://drive.google.com/file/d/1\\_DK31r5MK67SCvXo6bQRvZ6tegRKvris/view?usp=sharing](https://drive.google.com/file/d/1_DK31r5MK67SCvXo6bQRvZ6tegRKvris/view?usp=sharing)