

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»

ОТЧЁТ

По практике системы искусственного интеллекта

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

Выполнил студент группы УБСТ2304:

Паокин А. И.

Проверил: Кандзюба Е.В.

Москва 2025

Содержание

Введение	3
1. Техника безопасности	4
2. Расширение базы данных	6
2.1 Добавление именованных сущностей	6
2.2 Изменение диаграммы «Сущность-связь»	7
2.3 Изменение схемы базы данных	7
3. Создание и внедрение модели машинного обучения	9
3.1 Выбор данных для обучения модели	9
3.2 Выбор средств для создания модели	11
3.3 Обучение модели Microsoft Machine Learning	11
3.4 Внедрение модели в приложение	12
Заключение	16

Введение

В современном мире технологии искусственного интеллекта и машинного обучения находят применение в самых различных областях, включая системы контроля и учета. Одной из таких задач является автоматизация процессов взвешивания транспортных средств, где важно не только фиксировать вес, но и анализировать данные для классификации транспорта, выявления перегруза и оптимизации работы весового оборудования. Данный отчет посвящен разработке и внедрению настраиваемой трехдиапазонной звуковой сигнализации, которая активируется в зависимости от веса объекта. В ходе работы были расширены возможности базы данных, добавлена классификация транспортных средств, создана и обучена модель машинного обучения для автоматического определения типа транспорта, а также реализовано её внедрение в WPF-приложение. Эти решения позволяют повысить точность учета, минимизировать человеческий фактор и улучшить аналитические возможности системы.

1. Техника безопасности

При эксплуатации персонального компьютера на работника могут оказывать действие следующие опасные и вредные производственные факторы:

- повышенный уровень электромагнитных излучений;
- повышенный уровень статического электричества;
- пониженная ионизация воздуха;
- статические физические перегрузки;
- перенапряжение зрительных анализаторов.

Требования безопасности перед началом работы

Подготовить рабочее место. Отрегулировать освещение на рабочем месте, убедиться в отсутствии бликов на экране. Проверить правильность подключения оборудования к электросети. Проверить исправность проводов питания и отсутствие оголенных участков проводов. Убедиться в наличии заземления системного блока, монитора и защитного экрана. Протереть антистатической салфеткой поверхность экрана монитора и защитного экрана.

Проверить правильность установки стола, стула, подставки для ног, пюпитра, угла наклона экрана, положение клавиатуры, положение "мыши" на специальном коврик, при необходимости произвести регулировку рабочего стола и кресла, а также расположение элементов компьютера в соответствии с требованиями эргономики и в целях исключения неудобных поз и длительных напряжений тела.

Требования безопасности во время работы

Работнику при работе на ПК запрещается:

- прикасаться к задней панели системного блока (процессора) при включенном питании;

- переключать разъемы интерфейсных кабелей периферийных устройств при включенном питании;

- допускать попадание влаги на поверхность системного блока (процессора), монитора, рабочую поверхность клавиатуры, дисководов, принтеров и других устройств;

- производить самостоятельное вскрытие и ремонт оборудования;

- работать на компьютере при снятых кожухах;

- отключать оборудование от электросети и выдергивать электровилку, держась за шнур.

Продолжительность непрерывной работы с компьютером без регламентированного перерыва не должна превышать 2-х часов.

Во время регламентированных перерывов с целью снижения нервно - эмоционального напряжения, утомления зрительного анализатора, устранения влияния гиподинамии и гипокинезии, предотвращения развития познотонического утомления выполнять комплексы упражнений.

Требования безопасности по окончании работы

Отключить питание компьютера. Привести в порядок рабочее место. Выполнить упражнения для глаз и пальцев рук на расслабление.

2. Расширение базы данных

2.1 Добавление именованных сущностей

В обновленной версии системы появилась новая сущность - тип транспортного средства, которая расширила возможности работы с взвешиваниями. Теперь каждое взвешивание может быть связано с конкретным классом транспорта, что позволяет учитывать его характеристики и ограничения. Это изменение значительно повысило детализацию данных и открыло новые возможности для анализа.

Эти изменения позволили системе учитывать не только сам факт взвешивания, но и особенности транспортного средства, что важно для контроля допустимых нагрузок и анализа статистики. Все существующие процессы работы с клиентами, оборудованием и пользователями сохранили свою функциональность, получив при этом дополнительные возможности для интеграции с новыми функциями.

Добавленные и изменённые сущности представлены в таблице 2.1.1.

Таблица 2.1.1 – Новые и изменённые сущности из предметной области

Сущность	Наименование	Тип данных	Описание
TypeVehicle	TypeID	int (PK)	Уникальный идентификатор весов.
	Name	string	Наименование типа машины
	MaxWeight	float	Максимальный вес брутто
	Description	DateTime?	Подробное описание типа машины
Weighings	WeighingID	int (PK)	Уникальный ID взвешивания.
	ScaleID	int (FK Scales)	Ссылка на весы, на которых проводилось взвешивание.
	ClientID	int? (FK Clients)	Ссылка на клиента (если известен).
	VehicleNumber	string	Номер транспортного средства.
	TypeVehicleID	int (FK TypeVehicle)	Ссылка на тип машины
	GrossWeight	decimal	Полный вес ТС с грузом (брутто).
	TareWeight	decimal	Вес пустого ТС (тара).
	WeighingDateTime	DateTime	Дата и время взвешивания.
	OperatorID	int (FK Users)	ID оператора, проводившего взвешивание.

2.2 Изменение диаграммы «Сущность-связь»

Диаграмма сущность-связь, разработанная в Visio 2016, представляет собой модель системы учета весового контроля перед мостами. Она включает ключевые компоненты, такие как данные пользователей, сведения о весовом оборудовании, информацию о клиентах-перевозчиках, результаты взвешиваний и записи технического обслуживания.

Из-за изменения структуры предметной области совместно изменилась Диаграмма «Сущность связь», которая изображена на рисунке 2.2.1.

В нём добавилась таблица TypeVehicle и внешний ключ таблицы Weighings.

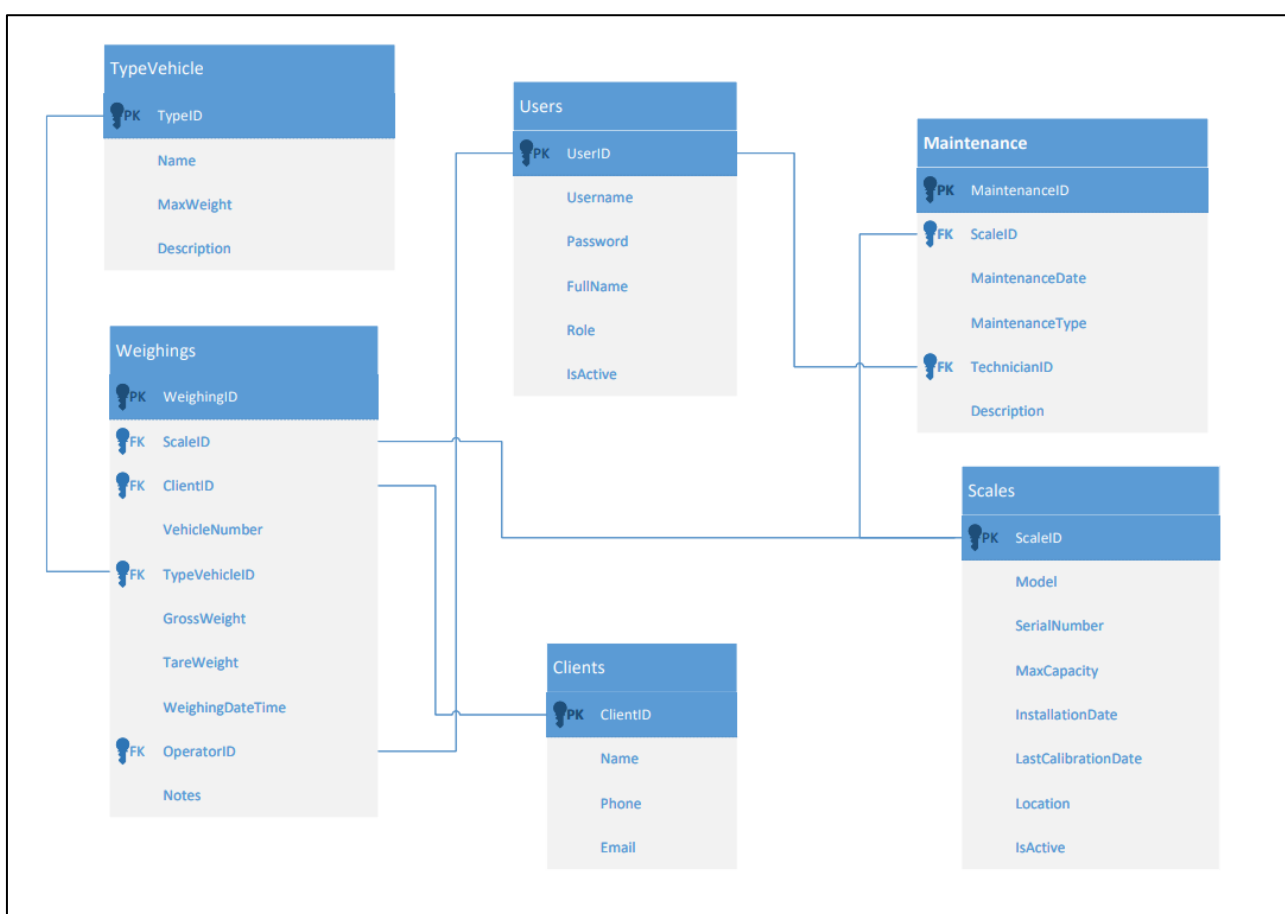


Рисунок 2.2.1 – Диаграмма «Сущность связь»

2.3 Изменение схемы базы данных

В обновленной версии базы данных BridgeScales произошли существенные изменения структуры и наполнения. Главным нововведением стало добавление системы классификации транспортных средств через отдельную таблицу, что позволило более детально категоризировать процессы взвешивания. Теперь

каждое взвешивание может быть привязано к определенному типу транспорта с указанием его характеристик. Это изменение потребовало модификации существующей таблицы взвешиваний, куда добавили соответствующее поле для связи.

Реализация изменённой схемы данных представлена на рисунке 2.3.1.

Эти изменения направлены на повышение гибкости системы учета и улучшение аналитических возможностей. Введение классификации транспорта открывает новые возможности для отчетности и анализа нагрузок на весовое оборудование. Расширенный набор тестовых данных позволяет более полно тестировать функциональность приложений, работающих с этой базой. Все модификации сохраняют обратную совместимость с существующими структурами, добавляя новые возможности без нарушения работы старого функционала.

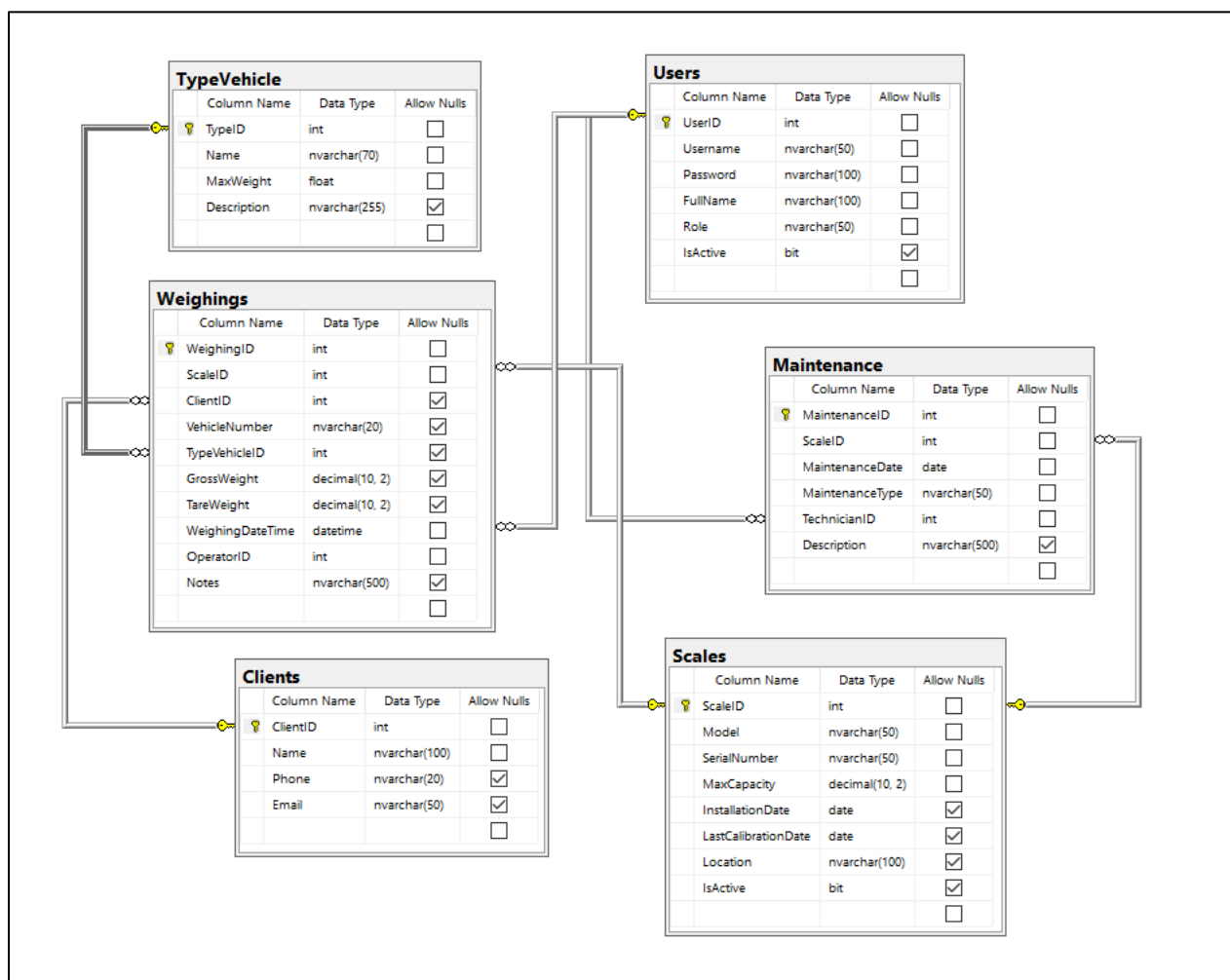


Рисунок 2.3.1 – Схема данных БД BridgeScales

3. Создание и внедрение модели машинного обучения

3.1 Выбор данных для обучения модели

Для машинного обучения необходима определённая выборка данных, её решил выгрузить из базы данных. Запрос для вывода данных для обучения модели изображён на рисунке 3.1.1.

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

Рисунок 3.1.1 – Запрос для вывода данных для обучения модели

Данные содержат информацию о взвешиваниях транспортных средств различных категорий, включая идентификатор типа транспортного средства, его название, максимально допустимый вес, описание категории, полный вес транспортного средства с грузом и вес пустого транспортного средства. В данных представлены четыре категории транспортных средств: легковые автомобили, грузовые автомобили 1 класса, грузовые автомобили 2 класса и грузовые автомобили 3 класса. Для легковых автомобилей максимальный вес составляет 3000 кг, а диапазоны полного веса и веса пустого транспортного средства варьируются от 1000 до 2900 кг и от 150 до 1800 кг соответственно. Грузовые автомобили 1 класса имеют максимальный вес 15000 кг с диапазонами полного веса от 8500 до 14800 кг и веса пустого транспортного средства от 4500 до 7500 кг. Грузовые автомобили 2 класса характеризуются максимальным весом 40000 кг, полным весом от 23500 до 39500 кг и весом пустого транспортного средства от 11500 до 13500 кг. Грузовые автомобили 3 класса обладают максимальным

весом 95000 кг, полным весом от 48200 до 92000 кг и весом пустого транспортного средства от 12000 до 27000 кг. Данные файла SQLData.xlsx представлены на рисунке 3.1.2.

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

Рисунок 3.1.2 – Данные запроса выгруженные в файл Excel

SQL-запрос извлекает данные для обучения модели из реляционной базы данных, соединяя таблицу взвешиваний (dbo.Weighings) со справочником типов транспортных средств (dbo.TypeVehicle) по полю TypeVehicleID. В результате запрос возвращает полную информацию о каждом взвешивании, включая характеристики типа транспортного средства, такие как название, максимальный вес и описание.

Эти данные могут быть использованы для решения задач машинного обучения, включая классификацию транспортных средств по типам на основе их весовых характеристик, прогнозирование допустимой загрузки, обнаружение аномалий (например, перегруза) и построение регрессионных моделей зависимости полного веса от массы пустого транспортного средства. Для улучшения качества модели можно рассчитать дополнительные признаки, такие как разница между полным весом и весом пустого транспортного средства (фактическая масса груза), отношение полного веса к максимально допустимому весу (коэффициент загрузки) и процент использования грузоподъемности. Эти признаки помогут модели лучше выявлять закономерности и повысить точность прогнозов.

3.2 Выбор средств для создания модели

Средство, которое было выбрано для обучения и сохранения моделей это Microsoft.ML. Главное преимущество Microsoft.ML — это прямая и бесшовная интеграция машинного обучения в WPF-приложения без костылей и сторонних зависимостей. В отличие от Python-решений, которые требуют сложных обходных путей вроде отдельного API-сервера или IPC-взаимодействия, ML.NET работает нативно внутри .NET-окружения. Это значит, что ваша модель машинного обучения становится полноценной частью приложения — вы просто вызываете методы предсказания прямо из C#-кода, как любую другую логику, без JSON-конвертаций, сетевых задержек или проблем с сериализацией.

Для WPF-разработчика это ключевое преимущество: представьте, что вам нужно добавить классификатор изображений или предсказание цен в интерфейс. С Python вы столкнётесь с необходимостью разворачивать Flask-сервер, настраивать виртуальные окружения и решать проблемы с блокировкой UI при долгих вычислениях. В ML.NET всё решается естественно для .NET — модель загружается в память, работает в фоне через `async/await` и интегрируется с MVVM-логикой без лишних слоёв.

Производительность тоже играет роль: вы избегаете накладных расходов на межпроцессное взаимодействие, а благодаря компилируемой природе C# и оптимизациям ML.NET некоторые операции выполняются быстрее, чем в Python-аналогах. Развёртывание становится тривиальным — ваше приложение по-прежнему собирается в единый EXE-файл или MSI-установщик без требований к установке Python или дополнительных библиотек.

3.3 Обучение модели Microsoft Machine Learning

База данных BridgeScales представляет собой систему управления и учета операций взвешивания на мостовых и железнодорожных весах. Она предназначена для автоматизации работы предприятий, занимающихся взвешиванием грузов, и включает в себя несколько взаимосвязанных таблиц,

обеспечивающих полный контроль над процессами. Реализация схемы данных представлена на рисунке 3.3.1.

```
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);
}
```

Рисунок 3.3.1 – Метод обучения и сохранения модели

3.4 Внедрение модели в приложение

Современные приложения всё чаще используют технологии машинного обучения для автоматизации сложных задач и принятия интеллектуальных решений. В данном случае рассматривается внедрение предварительно обученной модели классификации транспортных средств в WPF-приложение для учета взвешивания на мостовых весах. Модель определяет тип транспортного средства на основе его весовых характеристик: полной массы (GrossWeight) и массы без груза (TareWeight). Эта функциональность интегрирована в интерфейс добавления и редактирования данных взвешивания, что позволяет автоматизировать процесс и снизить вероятность ошибок.

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. На рисунке 3.4.1 изображён класс `VehicleData`.

```
using Microsoft.ML.Data;

namespace WpfAppUserInterface.Classes.LerningModel
{
    [Serializable]
    Ссылка: 4
    public class VehicleData
    {
        Ссылка: 1
        public VehicleData(float grossWeight, float tareWeight)
        {
            GrossWeight = grossWeight;
            TareWeight = tareWeight;
        }

        [LoadColumn(0)] public float TypeVehicleID;
        [LoadColumn(1)] public string Name;
        [LoadColumn(2)] public float MaxWeight;
        [LoadColumn(3)] public string Description;
        [LoadColumn(4)] public float GrossWeight;
        [LoadColumn(5)] public float TareWeight;
        Ссылка: 0
        public float NetWeight => GrossWeight - TareWeight;
    }
}
```

Рисунок 3.4.1 – Класс `VehicleData` для обработки данных моделью

Принцип работы системы заключается в следующем: при сохранении данных о взвешивании приложение получает текущие показания весов (`GrossWeight`), принимает от пользователя массу без груза (`TareWeight`), передаёт эти данные в модель и получает предсказанный тип транспортного средства. Затем все данные сохраняются в базу данных. Такой подход не только

автоматизирует классификацию, но и ускоряет процесс ввода, минимизируя влияние человеческого фактора.

На рисунке 3.4.2 в конструкторе класса WindowAddEditWeighing происходит загрузка и подготовка модели машинного обучения для классификации транспортных средств. Сначала создается контекст ML.NET, затем из файла vehicle_classifier_mlnet.zip загружается предварительно обученная модель. На основе загруженной модели инициализируется PredictionEngine, который будет использоваться для выполнения предсказаний. Этот механизм принимает данные типа VehicleData и возвращает результат в формате VehiclePrediction. После настройки модели создается новая запись о взвешивании с текущей датой, загружаются данные в элементы управления интерфейса и обновляются показания с весов.

```
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();
}
```

Рисунок 3.4.2 – Конструктор класса страницы с загрузкой модели

На рисунке 3.4.3 сохранении данных взвешивания система использует обученную модель для автоматического определения типа транспортного средства. Для этого создается объект VehicleData, содержащий текущий вес с датчика (_sensor.weight) и введенную пользователем массу без груза (tareweight). Модель обрабатывает эти данные и возвращает числовой идентификатор типа транспортного средства, который сохраняется в свойстве TypeVehicleId записи

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

Рисунок 3.4.3 – Сохранение данных тип машины

взвешивания. Это позволяет автоматизировать процесс классификации, исключая ручной ввод и снижая вероятность ошибок.

Главные преимущества этой реализации — автоматизация рутинных операций, бесшовная интеграция ML-функциональности в пользовательский интерфейс, высокая производительность благодаря использованию предварительно обученной модели, а также масштабируемость, позволяющая обновлять модель без изменения кода приложения. В дальнейшем это открывает возможности для расширения функционала, например, добавления анализа аномалий или прогнозирования загрузки весов.

Заключение

Проделанная работа демонстрирует эффективность применения машинного обучения для автоматизации процессов взвешивания транспортных средств. Внедрение модели классификации на основе данных о весе позволило значительно упростить работу операторов, снизить количество ошибок и повысить скорость обработки информации. Расширение базы данных и добавление новых сущностей открыло дополнительные возможности для анализа и отчетности. Интеграция модели в WPF-приложение подтвердила преимущества использования ML.NET для бесшовного внедрения искусственного интеллекта в существующие системы. Результаты проекта могут быть применены для дальнейшего развития системы, включая добавление новых функций, таких как прогнозирование загрузки или детектирование аномалий. Таким образом, проделанная работа вносит вклад в совершенствование технологий учета и контроля весовых параметров, делая их более интеллектуальными и удобными в использовании.