**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

С помощью разработанного веб-приложения на базе обученной модели появится возможность осуществлять прогноз стоимости автомобиля по его исходным параметрам с 98% предсказательной способностью. Вся работа состоит из 10 основных шагов. В данной работе решалась задача регрессии с использованием классически методов машинного обучения. Датасет был взят с сайта Kaggle.com.

Ссылка на исходный датасет представлена ниже.

[Vehicle dataset (kaggle.com)](https://www.kaggle.com/datasets/nehalbirla/vehicle-dataset-from-cardekho/data)

Основные этапы работы с датасетом для решения задачи регрессии при прогнозе стоимости автомобиля включают следующие шаги:  
1. Загрузка данных: Сначала необходимо загрузить датасет, содержащий информацию о различных автомобилях, и сохранить его в удобном формате (например, в формате CSV или Excel).

2. Изучение данных: Важно провести предварительный анализ данных, чтобы понять их структуру и содержание. Этот шаг включает ознакомление с названиями и типами переменных, проверку наличия пропущенных значений, анализ распределений переменных и поиск возможных выбросов.  
3. Предобработка данных: На этом этапе необходимо выполнить ряд операций для подготовки данных к обучению модели. Возможные операции включают удаление ненужных переменных, заполнение пропущенных значений, преобразование категориальных переменных в числовой формат и нормализацию числовых переменных.

4. Разделение на обучающую и тестовую выборки: Для оценки качества модели необходимо разделить данные на две части: обучающую выборку, на которой модель будет обучаться, и тестовую выборку, на которой будет оцениваться ее производительность. Обычно используется случайное разделение данных, например, в соотношении 70:30 или 80:20 между обучающей и тестовой выборками соответственно.

5. Выбор модели: Существует множество алгоритмов регрессии, которые можно использовать для прогнозирования стоимости автомобиля. Выбор модели зависит от характеристик данных и требуемой производительности. Некоторые популярные модели включают линейную регрессию, регрессию на основе деревьев решений и методы глубокого обучения.

6. Обучение модели: На этом этапе выбранная модель обучается на обучающей выборке. Это включает подгонку параметров модели к данным с использованием различных алгоритмов оптимизации.

7. Оценка производительности модели: После обучения модели необходимо оценить ее производительность на тестовой выборке. Это может быть выполнено с помощью различных метрик, таких как средняя абсолютная ошибка (MAE), средняя квадратичная ошибка (MSE) или коэффициент детерминации (R^2).

8. Тюнинг модели: В случае неудовлетворительной производительности модели можно провести ее тюнинг, изменяя параметры модели или используя другую модель. Этот шаг может включать перебор различных комбинаций параметров модели и выбор наилучшей комбинации.

9. Прогнозирование: После того как модель была обучена и оценена, ее можно использовать для прогнозирования стоимости автомобиля на новых данных. Это может быть выполнено путем подачи новых наблюдений на вход обученной модели.

10. Оценка результатов: Наконец, необходимо оценить результаты прогнозирования, сравнив их с фактическими значениями. Это поможет понять, насколько точно модель способна предсказывать стоимость автомобиля.  
Важно отметить, что каждый из этих этапов требует тщательного анализа и может потребовать дополнительных действий в зависимости от особенностей данных и поставленной задачи.

**Шаг 1. Подготовка исходных данных**

Для начала необходимо загрузить все необходимые библиотеки. Для работы потребуются: Numpy, Pandas, Matplotlib, Seaborn, Sklearn.

Далее с самого сайта небходимо скачать датасет под названием ‘car data.csv’. Ниже представлена его визуализация.

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

**Рис. 1** – Исходный датасет

Данный датасет состоит из 301 строчки и 9 столбцов. Описание столбцов представлено ниже:

1. Car\_Name – Название автомобиля
2. Year – Год покупки автомобиля
3. Selling\_Price – Цена по которой владелец хочет продать автомобиль (Целевая переменная)
4. Present\_Price – Это текущая цена автомобиля без учета выставочного зала
5. Kms\_Driven – Пробег автомобиля в километрах
6. Fuel\_Type – Тип потребляемого топлива
7. Seller\_Type – Тип продавца, Диллер или Физ. лицо
8. Transmission – Тип Трансмиссия
9. Owner – Количество владельцев автомобиля

Основная информация по датафрейму указана ниже.

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

**Рис. 2** – Статистика датасета

Исходя из данных статистики ни в одной из переменных нет пропущенных значений. Рассмотрим отдельно статистические показатели по численных и категориальным переменным.

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

**Рис. 3** – Статистика численных переменных

Исходя из Рис. 3 видно, что показатели расположены в пределах допустимых и возможных значений.

Рассмотрим категориальные переменные.

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

**Рис. 4** – Статистика категориальных переменных

**Шаг 2. Подготовка данных**

В результате исследования было выяснено, что в колонке Car\_Name находится 98 уникальных значений, что в свою очередь не может быть использовано в качестве зависимой переменной. Было принято решение убрать данный показатель из датафрейма.

Так как столбец Year определяет год, для лучшей работы, было принято решение провести разбалловку, где минимальное значение будет принимать максимальный год, а максимальное значение минимальный.

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

**Рис. 5** – Обновленный Датасет

Для определения и визуализации выбросов была написана отдельная функция. Ее результаты представлены на Рис. 6.

Изображение выглядит как снимок экрана, текст, диаграмма, Прямоугольник

Автоматически созданное описание

**Рис. 6** – Анализ выбросов датасета

Под первоначальный фильтр выбросов попало множество индексов. Однако недопустимо отбрасывать наблюдение только потому, что оно является выбросом. Они могут быть законными наблюдениями, и важно изучить природу выброса, прежде чем принимать решение о том, удалять его или нет. Было принято решение об удалении выбросов в двух случаях:

* Выброс обусловлен неверно введенными или измеренными данными
* Выброс создает значимую ассоциацию

Для данного анализа были построен следующий ряд графиков, представленный ниже.

Изображение выглядит как текст, снимок экрана, диаграмма

Автоматически созданное описание

**Рис. 7** – Поиск кандидатов на удаление выбросов

В результате анализа было удалено два выброса, которые сильно выбивались из общей картины. В результате анализа, пропущенных значений выявлено не было. Было обнаружено два дубликата, представлено ниже.

Изображение выглядит как текст, снимок экрана, Шрифт, линия

Автоматически созданное описание

**Рис. 8** – Поиск дубликатов

В результате чего данные строки не был удалены, поскольку есть допущения наличия двух автомобилей с одинаковыми характеристиками.

**Шаг 3. Разведочный анализ данных**

Проведен анализ категориальных переменных.

Существует 3 категории Fuel\_Type. Наибольшую частоту встречаемости имеет бензин, наименьшую – КПГ. Существует 2 категории типа продавца. Наибольшая частота встречается у дилера, наименьшая - у частного лица. Существует 2 категории трансмиссии. Наибольшая частота встречается у механической коробки передач, наименьшая – у автоматической.Изображение выглядит как снимок экрана, диаграмма, текст, График

Автоматически созданное описание

**Рис. 9** – Анализ категориальных переменных

Анализ численных переменных представлен также ниже. ### Вывод:

По мере роста текущей цены увеличивается и продажная цена. Таким образом, Цена продажи прямо пропорциональна Текущей цене.

По мере увеличения пробега автомобиля в километрах его цена продажи уменьшается. Таким образом, цена продажи обратно пропорциональна Пробегу автомобиля.

По мере старения автомобиля его Цена продажи уменьшается. Таким образом, Цена продажи обратно пропорциональна возрасту автомобиля. По мере того, как количество предыдущих автомобилей уменьшается, Цена продажи уменьшается.

С увеличением числа предыдущих владельцев автомобиля его Цена продажи уменьшается. Таким образом, Цена продажи обратно пропорциональна количеству владельцев.

Изображение выглядит как текст, диаграмма, снимок экрана, План

Автоматически созданное описание

**Рис. 10** – Анализ численных переменных

Дизельные автомобили дороже автомобилей на КПГ, а те уже дороже Бензиновых автомобилей.

Цена продажи автомобилей, продаваемых частными лицами, ниже, чем цена автомобилей, продаваемых дилерами.

Автомобили с автоматической коробкой передач стоят дороже, чем автомобили с механической коробкой передач.

**Шаг 3. Кодировка категориальных переменных**

Мы реализуем фиктивное кодирование для категориальных столбцов, поскольку все они являются номинальными переменными:

Изображение выглядит как текст, снимок экрана, линия, Шрифт

Автоматически созданное описание

**Рис. 11** – Кодировка категориальных переменных

В данном случае появляются переменные Fuel\_Type\_Diesel, при его наличии значение равно 1, при отсутствии 0. Точно такой же принцип у новой переменной Fuel\_Type\_Petrol. Переменная Seller\_Type\_Individual принимает значение 1 если продавец является физ. Лицом и 0 если диллером. Переменная Transmission\_Manual принимает значение 1 в случае, если коробка передач механическая и 0 если автоматическая.

**Шаг 4. Анализ Корреляций параметров**

Целевая переменная "Цена продажи" сильно коррелирует с Present\_Price, Seller\_Type и Fuel\_Type.

Некоторые независимые переменные, такие как Fuel\_Type\_Petrol и Fuel\_Type\_Disel, сильно коррелируют между собой, что называется мультиколлинеарностью.



**Рис. 12** – Анализ корреляций

**Шаг 5. Построение модели Линейной Регрессии**

Было произведено предварительное разбиение исходного датасета на тренировочную и обучающие выборки в соотношении 70:30.

Далее было проведено маштабирование переменных с помощью метода StandartScaler. Далее было проведено обучение для тренировочной выборки. Оценка модели осуществлялясь с помощью метрик MSE, MAE, RMSE, R2.

MSE (Mean Squared Error) - это средняя квадратичная ошибка, которая является одной из наиболее распространенных метрик для оценки производительности модели регрессии. MSE вычисляется путем нахождения среднего значения квадратов разностей между прогнозными значениями модели и фактическими значениями целевой переменной. Чем меньше значение MSE, тем лучше производительность модели.

MAE (Mean Absolute Error) - это средняя абсолютная ошибка, которая также используется для оценки производительности модели регрессии. MAE вычисляется путем нахождения среднего значения абсолютных разностей между прогнозными значениями модели и фактическими значениями целевой переменной. MAE также позволяет оценить точность модели, и чем меньше значение MAE, тем лучше производительность модели.

RMSE (Root Mean Squared Error) - это квадратный корень из средней квадратичной ошибки (MSE). RMSE также является распространенной метрикой для оценки производительности модели регрессии. RMSE имеет ту же интерпретацию, что и MSE, но в тех же единицах, что и целевая переменная. Чем меньше значение RMSE, тем лучше производительность модели.

R^2-score (Coefficient of Determination) - это метрика, которая позволяет оценить объяснительную способность модели. R^2-score выражает долю дисперсии целевой переменной, которая может быть объяснена моделью. Значение R^2-score находится в диапазоне от 0 до 1, где 0 означает, что модель не объясняет вариацию целевой переменной, а 1 означает, что модель объясняет всю вариацию. Чем ближе значение R^2-score к 1, тем лучше производительность модели.

Метрики MSE, MAE, RMSE и R^2-score позволяют оценить точность и объяснительную способность модели линейной регрессии. Они помогают сравнивать различные модели или изменения в модели и выбирать наилучшую модель для прогнозирования стоимости автомобиля.

**Шаг 6. Построение модели регрессии Лассо**  
Регрессия Лассо (Lasso regression) является одним из методов регуляризации линейной регрессии. Ее отличие от других методов, таких как регрессия гребня (Ridge regression), заключается в том, что она добавляет штраф к сумме квадратов коэффициентов регрессии, но этот штраф также включает L1-норму коэффициентов. Это приводит к тому, что некоторые коэффициенты регрессии становятся точно равными нулю, что позволяет выполнить отбор признаков и упростить модель.

Плюсы регрессии Лассо:

1. Отбор признаков: регрессия Лассо может автоматически выбирать наиболее важные признаки и установить остальные коэффициенты равными нулю. Это позволяет упростить модель и избежать переобучения.  
   2. Решение проблемы мультиколлинеарности: если в данных есть мультиколлинеарность (высокая корреляция между признаками), регрессия Лассо может выбрать один из коррелированных признаков и установить остальные коэффициенты равными нулю. Это помогает избежать проблемы неустойчивости оценок коэффициентов.  
   3. Интерпретируемость: из-за отбора признаков регрессия Лассо позволяет лучше понять, какие признаки влияют на целевую переменную и как.

Минусы регрессии Лассо:

1. Выбор параметра регуляризации: регрессия Лассо требует выбора параметра регуляризации, который контролирует силу штрафа. Оптимальное значение параметра может быть сложно определить, и неправильный выбор может привести к недообучению или переобучению модели.
2. Отсутствие стабильности относительно малых изменений в данных: регрессия Лассо может быть чувствительна к выбросам или небольшим изменениям в данных, что может привести к большим изменениям в оценках коэффициентов.
3. Неустойчивость в случае коллинеарности: если в данных присутствует сильная коллинеарность, регрессия Лассо может выбрать только один из коррелированных признаков и установить остальные коэффициенты равными нулю. Это может привести к потере информации и снижению производительности модели.  
   Суть работы регрессии Лассо заключается в минимизации функции ошибки, которая включает сумму квадратов разностей между прогнозными значениями модели и фактическими значениями целевой переменной, а также штраф в виде L1-нормы коэффициентов регрессии. Это приводит к тому, что некоторые коэффициенты становятся равными нулю, что позволяет выполнить отбор признаков и упростить модель. Оптимизация параметров модели выполняется с использованием методов, таких как координатный спуск или градиентный спуск.

**Шаг 7. Построение модели регрессии Ridge**

Регрессия гребня (Ridge regression) является методом регуляризации линейной регрессии, который также добавляет штраф к сумме квадратов коэффициентов регрессии, но в отличие от регрессии Лассо, этот штраф включает L2-норму коэффициентов.  
Плюсы регрессии гребня:

1. Решение проблемы мультиколлинеарности: регрессия гребня может справиться с мультиколлинеарностью путем уменьшения влияния коррелированных признаков. Она не обнуляет коэффициенты, а просто уменьшает их значения, что позволяет сохранить информацию о взаимосвязи между признаками.  
   2. Улучшение устойчивости оценок коэффициентов: регрессия гребня уменьшает дисперсию оценок коэффициентов, что делает их более стабильными относительно малых изменений в данных.  
   3. Лучшая предсказательная способность: благодаря уменьшению влияния шума и мультиколлинеарности, регрессия гребня может улучшить предсказательную способность модели.

Минусы регрессии гребня:

1. Отсутствие отбора признаков: в отличие от регрессии Лассо, регрессия гребня не обнуляет коэффициенты, а только уменьшает их значения. Это означает, что все признаки остаются в модели, что может привести к избыточности и сложности интерпретации.  
2. Зависимость от выбора параметра регуляризации: также как и в регрессии Лассо, регрессия гребня требует выбора параметра регуляризации. Неправильный выбор может привести к недообучению или переобучению модели.  
Суть работы регрессии гребня заключается в минимизации функции ошибки, которая включает сумму квадратов разностей между прогнозными значениями модели и фактическими значениями целевой переменной, а также штраф в виде L2-нормы коэффициентов регрессии. Это позволяет уменьшить значения коэффициентов и снизить их влияние на модель, сохраняя при этом информацию о взаимосвязи между признаками. Оптимизация параметров модели выполняется с использованием методов, таких как координатный спуск или градиентный спуск.

**Шаг 8. Построение модели Random Forest**

Random Forest (случайный лес) - это ансамблевый метод машинного обучения, который объединяет несколько деревьев решений для получения более точных и стабильных предсказаний. Каждое дерево в случайном лесу строится независимо, используя случайные подвыборки данных и случайные подмножества признаков.

Плюсы Random Forest:

1. Хорошая предсказательная способность: Random Forest способен обрабатывать сложные задачи и достигать высокой точности предсказаний.  
   2. Устойчивость к переобучению: благодаря случайному выбору подмножеств данных и признаков, Random Forest имеет низкую вероятность переобучения и хорошую обобщающую способность.

3. Способность обрабатывать большие наборы данных: Random Forest хорошо масштабируется и может эффективно работать с большими объемами данных.  
4. Возможность оценки важности признаков: Random Forest позволяет оценить важность каждого признака, что может быть полезно для отбора наиболее значимых признаков.

Минусы Random Forest:

1. Сложность интерпретации: из-за использования нескольких деревьев, Random Forest может быть сложным для интерпретации результатов и понимания взаимосвязей между признаками.  
2. Время обучения: построение Random Forest может занимать больше времени, особенно при больших объемах данных и большом количестве деревьев.  
3. Подверженность шуму: если в данных присутствует много шума или выбросов, Random Forest может быть менее эффективным.  
Суть работы Random Forest заключается в построении ансамбля деревьев решений и комбинировании их предсказаний. Каждое дерево строится независимо на случайной подвыборке данных и случайном подмножестве признаков. Затем, для каждого нового наблюдения, все деревья предсказывают его значение. В случае регрессии, окончательное предсказание получается усреднением предсказаний всех деревьев. В случае классификации, окончательное предсказание получается путем голосования или усреднения вероятностей предсказаний всех деревьев.

**Шаг 9. Построение модели Gradient Boosting**

Gradient Boosting (градиентный бустинг) - это ансамблевый метод машинного обучения, который строит модель путем комбинирования слабых моделей (обычно деревьев решений) в последовательности, где каждая следующая модель исправляет ошибки предыдущей модели. Градиентный бустинг использует градиентный спуск для минимизации ошибки модели.

Отличие градиентного бустинга от других методов:

1. Последовательность: градиентный бустинг строит модель пошагово, добавляя новые модели для исправления ошибок предыдущих моделей, в отличие от случайного леса, где все деревья строятся независимо.  
2. Взвешивание: градиентный бустинг в каждом шаге обучения взвешивает ошибки предыдущих моделей, чтобы сосредоточиться на тех наблюдениях, которые были неправильно классифицированы или предсказаны с большой ошибкой.

Плюсы градиентного бустинга:

1. Высокая предсказательная способность: градиентный бустинг способен достигать высокой точности предсказаний и обрабатывать сложные задачи.  
2. Устойчивость к переобучению: благодаря последовательному обучению и взвешиванию ошибок, градиентный бустинг имеет низкую вероятность переобучения и хорошую обобщающую способность.  
3. Способность обрабатывать различные типы данных: градиентный бустинг может работать с различными типами данных, включая числовые и категориальные признаки.

4. Возможность оценки важности признаков: градиентный бустинг позволяет оценить важность каждого признака, что может быть полезно для отбора наиболее значимых признаков.

Минусы градиентного бустинга:

1. Время обучения: градиентный бустинг может занимать больше времени на обучение, особенно при больших объемах данных и большом количестве моделей.  
2. Чувствительность к выбросам: градиентный бустинг может быть чувствителен к выбросам или шуму в данных, что может привести к переобучению или недообучению модели.  
Суть работы градиентного бустинга заключается в построении последовательности слабых моделей, где каждая следующая модель исправляет ошибки предыдущей модели. На каждом шаге обучения градиентный бустинг вычисляет градиент функции потерь по отношению к предсказаниям текущей модели и использует его для обновления параметров модели. В итоге, окончательное предсказание получается путем комбинирования предсказаний всех моделей в последовательности.

**Шаг 10. Анализ итоговых результатов и выбор лучшей модели.**

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

**Рис. 13** – Анализ метрик моделей

**Шаг 11. Реализация веб-приложения и его использование.**

Веб-приложение было разработано с помощью библиотеки Flask. Запуск осуществляется через консоль и проводится с помощью команды flask run.

Изображение выглядит как текст, снимок экрана, дисплей, программное обеспечение

Автоматически созданное описание

**Рис. 14** – Интерфейс веб-приложения

С помощью данного интерфейса достаточно ввести исходные данные по году выпуска автомобиля, текущей стоимости, пробега, количества владельцев, типа трансмиссии, типа топлива и типа продавца для прогноза стоимости автомобиля.

Все данные представлены на открытом GitGub.

Ссылка ниже

[vvk79/Innopolis\_final\_work: My final machine learning homvework (github.com)](https://github.com/vvk79/Innopolis_final_work)