**Prediction Model For Car Price**

Synopsis Report submitted in partial fulfillment

of the requirement of

**Major Project 1(CSP701)**

for the degree of

B. E.(Computer Engineering)

By

**Yash Patil 19102A0037**

**Omkar Jalgaonkar 19102A0051**

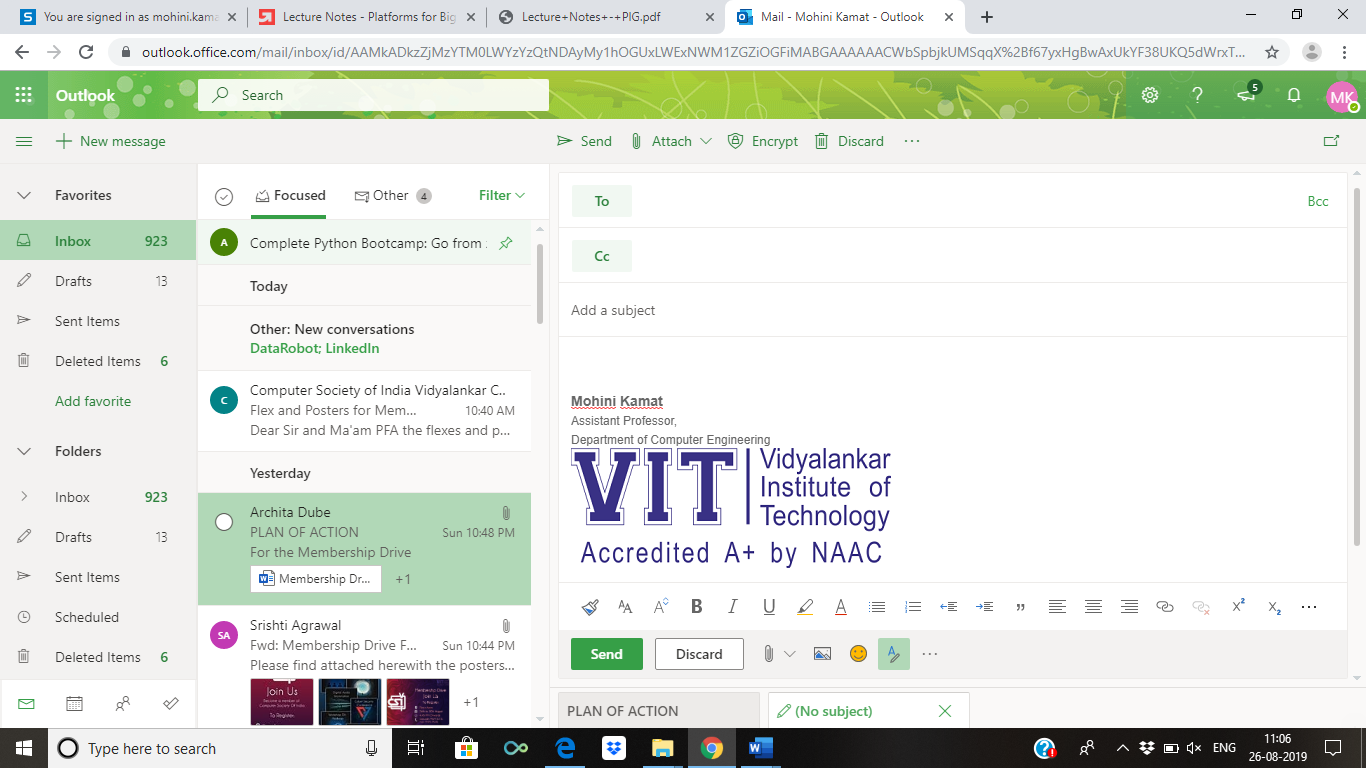
**Amey Thorat 19102B0049**

**Payas Patel 20102B2007**

Under the Guidance of

**Dr. Umesh Kulkarni**

Department of Computer Engineering



(An Autonomous Institute Affiliated to University of Mumbai)

Vidyalankar Institute of Technology

Wadala(E), Mumbai 400 037

University of Mumbai

2022-23

**CERTIFICATE OF APPROVAL**

This is to certify that the project entitled

**“Prediction Model For Car Price”**

is a bonafide work of

**Yash Patil 19102A0037**

**Omkar Jalgaonkar 19102A0051**

**Amey Arun Thorat 19102B0049**

**Payas Patel 20102B2007**

in partial fulfillment of degree course in

Computer Engineering

As laid down by University of Mumbai during the academic year

2022-23

Guide Head of Department Principal Dr. Umesh Kulkarni Dr. Sachin Bojewar Dr. Sunil Patekar

|  |  |
| --- | --- |
| Examiner 1 | Examiner 2 |

Major Project Report Approval

This project report entitled ***Prediction Model For Car Price*** by

1. ***Yash Patil 19102A0037***
2. ***Omkar Jalgaonkar 19102A0051***
3. ***Amey Thorat 19102B0049***
4. ***Payas Patel 20102B2007***

is approved for Major Project 1(CSP701) for the B. E.(Computer Engineering)

|  |  |
| --- | --- |
| Internal Examiner | External Examiner |

Date:

Place:

Declaration

I declare that this written submission represents my ideas in my own words and where others' ideas or words have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Name of student Roll No. Signature

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | Amey Arun Thorat | 19102B0049 | A picture containing text, linedrawing  Description automatically generated |
| 2 | Omkar Jalgaonkar | 19102A0051 | Text, letter  Description automatically generated |
| 3 | Payas Patel | 20102B2007 | A close-up of a pen  Description automatically generated with medium confidence |
| 4 | Yash Patil | 19102A0037 | A picture containing text, businesscard  Description automatically generated |

Date:

Place:

**Acknowledgements**

This Project wouldn’t have been possible without the support, assistance, and guidance of a number of people whom we would like to express our gratitude to. First, we would like to convey our gratitude and regards to our mentor **Dr. Umesh Kulkarni** for guiding us with his constructive and valuable feedback and for his time and efforts. It was a great privilege to work and study under his guidance.

We would like to extend our heartfelt thanks to our Head of Department, **Dr. Sachin Bojewar** for overseeing this initiative which will in turn provide every Vidyalankar student a distinctive competitive edge over others.

We appreciate everyone who spared time from their busy schedules and participated in the survey. Lastly, we are extremely grateful to all those who have contributed and shared their useful insights throughout the entire process and helped us acquire the right direction during this research project.

**Abstract**

The price of a new car in the industry is fixed by the manufacturer with some additional costs incurred by the Government in the form of taxes. So, customers buying a new car can be assured of the money they invest to be worthy. But, due to the increased prices of new cars and the financial incapability of the customers to buy them, Used Car sales are on a global increase. Therefore, there is an urgent need for a Used Car Price Prediction system which effectively determines the worthiness of the car using a variety of features. Existing System includes a process where a seller decides a price randomly and buyer has no idea about the car and it’s value in the present day scenario. In fact, seller also has no idea about the car’s existing value or the price he should be selling the car at. To overcome this problem we have developed a model which will be highly effective. Regression Algorithms are used because they provide us with continuous value as an output and not a categorized value. Because of which it will be possible to predict the actual price a car rather than the price range of a car. User Interface has also been developed which acquires input from any user and displays the Price of a car according to user’s inputs.

**Table of Contents**

|  |  |  |
| --- | --- | --- |
| **Sr No** | **Description** | **Page No** |
| 1 | Introduction | 8 |
| 2 | Problem Definition | 9 |
| 3 | Literature Survey | 10 |
| 4 | Flow Chart | 12 |
| 5 | Algorithm Used | 13 |
| 6 | Code | 15 |
| 7 | Output | 20 |
| 8 | Conclusion | 28 |
| 9 | Future Scope | 29 |
| 10 | References | 30 |

**Introduction**

Determining whether the listed price of a used car is a challenging task, due to the many factors that drive a used vehicle’s price on the market. The focus of this project is developing machine learning models that can accurately predict the price of a used car based on its features, in order to make informed purchases. We implement and evaluate various learning methods on a dataset consisting of the sale prices of different makes and models . We will compare the performance of various machine learning algorithms like Linear Regression, Ridge Regression, Lasso Regression, Elastic Net, Decision Tree Regressor and choose the best out of it. Depending on various parameters we will determine the price of the car. Regression Algorithms are used because they provide us with continuous value as an output and not a categorized value because of which it will be possible to predict the actual price a car rather than the price range of a car. User Interface has also been developed which acquires input from any user and displays the Price of a car according to user’s inputs.

**Problem Definition**

The problem definition of prediction model for car price is as follows :

We will first collect the data in csv format and then we will pre-process the data and clean the data and remake the cleaned data file in csv file.

Then will we start showing the relations of one attribute with others.

Then we will use linear regression model and start testing and training the dataset which we have already made in the clean data set file.

Once we have done we will make a pipeline and see the accuracy of the project.

And after that we will give the inputs required ie. Company name, model name, kilometres travelled, transmission, year etc and predict the price of given car.

We will make frontend using python flask framework having 2 pages i.e. Prediction Page and Details Page including clock on it.

**Literature Survey**

According to author Sameerchand, they have done the predictions of vehicle price from the historical data that has been collected from daily newspapers. They have used the supervised machine learning techniques for predicting the price of vehicles. Many other algorithms such as multiple linear regression, k- nearest neighbor algorithms , naïve based, and some decision tree algorithms also been used. All the four algorithms are compared and found the best algorithm for prediction. They have faced some difficulties in comparing the algorithms, somehow they have managed.

According to authors Pattabiraman, this paper is more concentrated on the relation between seller and buyer. In order to predict the price of four wheelers, more features are required such as already given price, mileage, make, model, trim, type, cylinder, liter, doors, cruise, sound, leather. Using these features the price of vehicle has been predicted with the help of statistical analysis system for exploratory data analysis.

According to authors Enis Gegic et al, in this paper the mainly concentrate on collecting various data from web portal by using web scrap techniques. And those have been compared with the help of different machine learning algorithms to predict the vehicle price in easy manner. They classified the price according to different ranges of price that is already given. Artificial neural network, support vector machine, random forest algorithms were used on different datasets to build classifiers model.

Another approach was given by Richardson in his thesis work. In his theory it states more durable vehicles will be produced by vehicle producer. He compared the hybrid vehicles and traditional vehicles in hoe it actually retains their value for longer time using multiple regression techniques. This improves the environmental conditions, and also it helps to provide huge efficiency of using fuels.

Wu et al, in this paper they have used neuro fuzzy knowledge based system to demonstrate vehicle price prediction. By considering the following attributes such as brand, year of production and type of engine they predicted a model which has similar results as the simple regression model. Moreover, they made an expert system named ODAV (Optimal Distribution of Auction Vehicles) as there is a high demand for selling the by vehicles at the end of the leasing year by vehicle dealers. This system gives insights into the best prices for vehicles, as well as the location where the best price can be gained. To predict a price of vehicles, the K – nearest neighbor machine learning algorithm has been used which is based on regression models. More number of vehicles has been exchanged through this system so this particular system is more successfully managed.

The first paper is Predicting the price of Used Car Using Machine Learning Techniques. In this paper, they investigate the application of supervised machine learning techniques to predict the price of used cars in Mauritius. The predictions are based on historical data collected from daily newspapers. Different techniques like multiple linear regression analysis, k-nearest neighbours, naïve bayes and decision trees have been used to make the predictions.

The Second paper is Car Price Prediction Using Machine Learning Techniques. Considerable number of distinct attributes are examined for the reliable and accurate prediction. To build a model for predicting the price of used cars in Bosnia and Herzegovina, they have applied three machine learning techniques (Artificial Neural Network, Support Vector Machine and Random Forest).

The Third paper is Price evaluation model in second hand car system based on BP neural networks. In this paper, the price evaluation model based on big data analysis is proposed, which takes advantage of widely circulated vehicle data and a large number of vehicle transaction data to analyze the price data for each type of vehicles by using the optimized BP neural network algorithm. It aims to establish a second-hand car price evaluation model to get the price that best matches the car.

**Flow Chart**

Start

Data Collection

Importing Required Packages and Opening Data Set File

Creating Backup File and Data Pre Processing

Creating Cleaned Data Set File and Checking Relationship of attributes with each other

Training and Testing Data

Using Linear Regression Model Find Accuracy of Model

Predict the Price Of Car on Flask WebApp

Display Car Details on WebApp

Stop

**Algorithm Used**

**Linear Regression** is a machine learning algorithm based on **supervised learning**. It performs a **regression task**. Regression models a target prediction value based on independent variables. It is mostly used for finding out the relationship between variables and forecasting. Different regression models differ based on – the kind of relationship between dependent and independent variables they are considering, and the number of independent variables getting used.  
![Chart, scatter chart

Description automatically generated](data:image/jpeg;base64,/9j/4AAQSkZJRgABAQEAYABgAAD/4RDaRXhpZgAATU0AKgAAAAgABAE7AAIAAAAFAAAISodpAAQAAAABAAAIUJydAAEAAAAKAAAQyOocAAcAAAgMAAAAPgAAAAAc6gAAAAgAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAGRlbGwAAAAFkAMAAgAAABQAABCekAQAAgAAABQAABCykpEAAgAAAAM2MQAAkpIAAgAAAAM2MQAA6hwABwAACAwAAAiSAAAAABzqAAAACAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAMjAxODowOToxMSAyMjoyMzo1MQAyMDE4OjA5OjExIDIyOjIzOjUxAAAAZABlAGwAbAAAAP/hCxdodHRwOi8vbnMuYWRvYmUuY29tL3hhcC8xLjAvADw/eHBhY2tldCBiZWdpbj0n77u/JyBpZD0nVzVNME1wQ2VoaUh6cmVTek5UY3prYzlkJz8+DQo8eDp4bXBtZXRhIHhtbG5zOng9ImFkb2JlOm5zOm1ldGEvIj48cmRmOlJERiB4bWxuczpyZGY9Imh0dHA6Ly93d3cudzMub3JnLzE5OTkvMDIvMjItcmRmLXN5bnRheC1ucyMiPjxyZGY6RGVzY3JpcHRpb24gcmRmOmFib3V0PSJ1dWlkOmZhZjViZGQ1LWJhM2QtMTFkYS1hZDMxLWQzM2Q3NTE4MmYxYiIgeG1sbnM6ZGM9Imh0dHA6Ly9wdXJsLm9yZy9kYy9lbGVtZW50cy8xLjEvIi8+PHJkZjpEZXNjcmlwdGlvbiByZGY6YWJvdXQ9InV1aWQ6ZmFmNWJkZDUtYmEzZC0xMWRhLWFkMzEtZDMzZDc1MTgyZjFiIiB4bWxuczp4bXA9Imh0dHA6Ly9ucy5hZG9iZS5jb20veGFwLzEuMC8iPjx4bXA6Q3JlYXRlRGF0ZT4yMDE4LTA5LTExVDIyOjIzOjUxLjYxMDwveG1wOkNyZWF0ZURhdGU+PC9yZGY6RGVzY3JpcHRpb24+PHJkZjpEZXNjcmlwdGlvbiByZGY6YWJvdXQ9InV1aWQ6ZmFmNWJkZDUtYmEzZC0xMWRhLWFkMzEtZDMzZDc1MTgyZjFiIiB4bWxuczpkYz0iaHR0cDovL3B1cmwub3JnL2RjL2VsZW1lbnRzLzEuMS8iPjxkYzpjcmVhdG9yPjxyZGY6U2VxIHhtbG5zOnJkZj0iaHR0cDovL3d3dy53My5vcmcvMTk5OS8wMi8yMi1yZGYtc3ludGF4LW5zIyI+PHJkZjpsaT5kZWxsPC9yZGY6bGk+PC9yZGY6U2VxPg0KCQkJPC9kYzpjcmVhdG9yPjwvcmRmOkRlc2NyaXB0aW9uPjwvcmRmOlJERj48L3g6eG1wbWV0YT4NCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgPD94cGFja2V0IGVuZD0ndyc/Pv/bAEMABwUFBgUEBwYFBggHBwgKEQsKCQkKFQ8QDBEYFRoZGBUYFxseJyEbHSUdFxgiLiIlKCkrLCsaIC8zLyoyJyorKv/bAEMBBwgICgkKFAsLFCocGBwqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKv/AABEIAXwCPAMBIgACEQEDEQH/xAAfAAABBQEBAQEBAQAAAAAAAAAAAQIDBAUGBwgJCgv/xAC1EAACAQMDAgQDBQUEBAAAAX0BAgMABBEFEiExQQYTUWEHInEUMoGRoQgjQrHBFVLR8CQzYnKCCQoWFxgZGiUmJygpKjQ1Njc4OTpDREVGR0hJSlNUVVZXWFlaY2RlZmdoaWpzdHV2d3h5eoOEhYaHiImKkpOUlZaXmJmaoqOkpaanqKmqsrO0tba3uLm6wsPExcbHyMnK0tPU1dbX2Nna4eLj5OXm5+jp6vHy8/T19vf4+fr/xAAfAQADAQEBAQEBAQEBAAAAAAAAAQIDBAUGBwgJCgv/xAC1EQACAQIEBAMEBwUEBAABAncAAQIDEQQFITEGEkFRB2FxEyIygQgUQpGhscEJIzNS8BVictEKFiQ04SXxFxgZGiYnKCkqNTY3ODk6Q0RFRkdISUpTVFVWV1hZWmNkZWZnaGlqc3R1dnd4eXqCg4SFhoeIiYqSk5SVlpeYmZqio6Slpqeoqaqys7S1tre4ubrCw8TFxsfIycrS09TV1tfY2dri4+Tl5ufo6ery8/T19vf4+fr/2gAMAwEAAhEDEQA/APpGiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKwvF9v4ku9D+z+Drqzs7+WUK9zdqWEMeDuZQOrdMA8UAbEN3b3EkscE8UrwttlVHBKH0IHQ/Wq9xrWl2c7Q3epWcEq/ejlnVWH4E147+z1aTaf4i+Idld3cl7PbauIZLmT70zK0qlz7nGa881T+wtW+IHi/xB468IeIdXsBehIrmyjdI7WJAQWdgQM42YHoDzQB9WC9tWsxdrcwm2IyJhINhHru6UttdW95CJbSeKeM9HicMPzFeD+IF0PW9c+G/gXQJpf8AhFL6KS68tJWzLGodgrMTnqpBHuaRp4fhT8Wtb0bw80kGmXPh9r+O0LFkimUOAVB6fc/WgD3b+0LL7X9l+1wfaP8Anj5o3/8AfOc0+4urezhMt3PHBGOryuFH5mvmoeDrF/2e38fmS4/4SjH2/wDtDzm37vN+716Y4xW2LqL4q/FPw9oniFpJ9Mh0BdQmtlYqsszBQS2OvWgD3yGaK4iWWCRJY2+66MCD+IqKLULOe4a3hu4JJl+9GkoLD6jOa+dZden+HFz8T/DOiXMyWlhBFcaejsW+zmVUyFJ6D95+go8S+EbDwN8LPDXjbQXng16O5tZLi7MzE3HmA7gwzyM0AfSVFc9rPjzw14btrOXxDq0GnC8jEkImz8wxnsPemaB8QvCfinUDZeH9ctr+5VC5iiJyFHfkUAdJRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQB5V8I9B1XSPGHxFuNSsprSK/1p5rWSRcCVDJKQy+o5H51mapr/xNi07WvDOp+E21i4vjJFZalbOqQCJhgbgTxjNe0UUAeF3Xw01/wdpfgfWdDtBrGo+Gw63dojgNKkgO4IT6bjV/RvC2t+PfHGseK/EujSaHbz6SdMsrW4YPJhgcu2OnLGvZaKAPnf8Asrx6vw0b4WjwpKXL+QNY85fs/keZu3euccf0ro9a8J634D8baJ4q8M6Q+uW9rpQ0y8toGCSkKBhxnr90V7LRQB4hp/w51vxjZeOtb8Q2I0m/8SxrFZ2krBmhRAApYjudq1nXumeOPG3hfQvAWo+FZtLisLiE3+qSyqYmjiB+4BySa+gKKAMbV/COga+kC61pVtfC3XbF5yZ2D2ryfwzo2naF+1PqNjo9nFZ2q6SrCKJcKCQMmvca8bsP+TttT/7A6fyFAHslFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAV43Yf8nban/wBgdP5CvZK8bsP+TttT/wCwOn8hQB7JRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFeN2H/ACdtqf8A2B0/kK9krxuw/wCTttT/AOwOn8hQB7JRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFY3ijxZovg3Rn1PxFfJaWy8LnJaRv7qqOSfpXEWf7RPw6vJI4xqs8LySiMCe1dAoP8RbG0KO/OfagD1Ciq9hqNnqtjHeaZdw3ltKN0c0EgdGHsRxVigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAK8bsP+TttT/7A6fyFeyV43Yf8nban/wBgdP5CgD2SiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooriPG3xd8JeArxLPW712vHG77PbxmR1Hq2On40AdvRWL4X8XaJ4y0oaj4dvku7fdtYgEMh9Cp5H41tUAFFFec/HbX9V8OfCi+u9DeSK4kkSFp4jhokY8sD26Y/GgDv4760muGgiuoXmQkNGsgLLjrkdah1fWNO0HTZdQ1m8hs7SL78szYA9vc+1fnxpet6lo+rwalpt5NDdwSCRJFc5zn9a+sPDXwu1PxrLZeJ/izqz6u7Ks1to8R2WsQIyNyjGT7dD3LdKAPKfjr4qu/iL9l1bQdMvz4b00tF9ukiIjlkJ+8PQds14tX6Nx6fZxWAsY7WFLQJsECxgIF9NvTFc3rnwt8E+Io0TVvDdjJ5f3WiQwsPbdGVOPbNAHjv7Js2pNB4hhfedLVomTd90THdux77dufw9a+jqoaLoWl+HdMj0/Q7GGxtI/uxQrgfU+p9zV+gAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAK8bsP+TttT/wCwOn8hXsleN2H/ACdtqf8A2B0/kKAPZKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACvh/446Pqml/FrWZdUR9l3OZreVuQ8Z+7g+3THavuCqeo6RpusRLFq2n2t9GhyqXUCyBT6gMDQB8+/sp6NqtuNb1WeGSPTLhEihZ8gSSKSSQO+B3r0XUvj78PtL1yTS59XaWSJ/LeaCFpIlPf5xwceoyK7LxHaSN4L1a000pbStYTRwMG8tY2MbBTn+HB79q/PRg3mFTy2ccc5NAH6Aa9498NeG/Dkeuapq1uljMm63eNg5uOMgRgcsfp+OK8vvX8a/HKFrextn8MeDJcB5ryMG4vVyDlV7D3zj3NeP/Dzxh4H8Of2fd+L9O1LXdRgLIiTOGt7JQfl2o3U4/AenevsDw94g03xRodvq+i3AuLO4GUccEeoI7H2oA/Pa+ga01K4t2jaJoZWQox5Qg4wa+tP2Zde1bWPh7dQaq0k0NjdeTbTyNkldoJX14J6++O1dD4r+Bfgrxfrj6tqNrcW93KczNaTbBK2fvMCDz7jFdn4e8PaZ4W0SDSdDtVtrSAYVF6k9yT3J9aANOiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAqlq2safoWmy6hrF3FZ2sQy8srbQP/AK9Xaz9Z0HTPEFrFba1ZRXsEUyzJHKuVDgEA4/E0Acp8NfihbfEm4137DYNa2ul3CxRTPJlrhW3YYrtGz7vTJ61haj8bZo7rU7jQvCV9q+h6TK0V7qkUwVVKjLFVwdwH1FZfwYiK+KvitFaKqMNalWJVGAp3zAYqr8KpI4/2efFHnsoMc18Jsnodvf8ADFAHd+Ivirpuk+H9Gv8ASbOfWLrXSBp1lAQryk9cnsB360eEPiauu6lqOkeIdHn8Paxp0IuJrWeQOpixnergDI/CvGfBEd1B4o+EEl8wFs1tdCPdxgkSY/mK6T4gR3Vx8eNVGmMN0fhKUTY7H951/MUAb5+O7iE6yPCGoHwoLjyDrPmj+/t3eXj7ue+a6Pxd8TY9CvtN0vw/pE3iHV9Th8+3tbeQIPKxnezEHA/CvO45YP8AhjBzuXb9i2df4/Oxj65pPAMd1b/HPQRqTDL+EoxDn1wnH5UAei+G/ipp2saJrV3q9nPot5oJI1GyuCGaLHQg8bgce1YNh8cJWuNNutc8I32k6Dqsyw2eqyShlYtnaWXA2g49TXnXjmO6n8YfF+SwYG2Wytll2+oWLP8AJq6b4oyRP+zb4bELKTJNYiHB6naTx+ANAHvFeN2H/J22p/8AYHT+QrqvGMvxEt7PTF8DW2n3D+SBeG8cLhsDpn8a83+HsniWX9pO/bxrDbQ6r/ZY3patlNuBt/SgD6CooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiivDvj18ZNU8D6ha6B4a8uO+mgFxPcuoby0LEKoHTJ2n9PWgD3Givnn4H/ABy1rxR4rh8L+KtlzLdI5tbpECsGRC5VgOuVU8+o969/vb2102ylvNQuI7a2hUvJNK4VUA7kmgDzb4hfHnw54B1ptHlt7nUtRjCmaK32hYsjOGYnrg5wM9R0pIf2g/BE3hZdXS4uDO7GNNNEWbhpAAdoUcdxznHvXy18QP8AifeNte8Q6LHe3uj3eoSNFevbsEJY7tmcYGM4A64xXtv7Nfw1urD7b4n8R6Z5RmRYtPS5i+fHVpACOM/KAfY+1AHQL4b8b/F+YXPjCefwv4XJBj0i3bE90uc5kPbP/wCod69Bt/h34Y03QW03R9EsbULC0cUnkhnUkYDFzlie+SSa6iigD8+/EXg7XvD/AIhuNM1HTroXCSsqnyy3m8/eBHXPWvrb4BeFNV8J/DSODXFkhuLqdpxbPnMKnAAI7E4z+VelvDHI6PJGrMhyjMoJX6elPoAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooA5fwn4FsfCOseINQsbiaWTXb03k6yYwjFmbC+3zmue1f4HeHdV1e7u473VLC1v5PMvdPs7ox29w3qU6V6TRQByniX4c6B4m8P2OlXEMlomnFTYz2j+XJbEdCrdugqPwd8N9I8HveXEc13qd/fqFur3UJTLJIoGNuT29q6+igDzI/AfwwdR3/bNVGlmf7QdHF232XzM5zs9P85rf8Y/DfSPGLWdxNLdabf2IK2t7p8piliX+7kdvauuooA5Tw18OdB8M6DfaXBFLeLqRZr6e8k8yS5J6lm79TWDpPwO8OaXrFpdve6pe2thL51lp91dF7e3bsQnt/nNek0UAFeN2H/J22p/9gdP5CvZK8bsP+TttT/7A6fyFAHslFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFeffGL4kv8NvCcd5ZwJPf3khhtlkztBAyWPrjjivQa4X4s/DeP4leFF09LlbS+tpPNtZnXKhsYKt3APqP1oA8K8D/tIeJ18VWkHil4LzTriRYpNkQRo8nAYEema+rlYOoZTlWGQR3FfL/gf9mbXoPFVpd+LLi0i0+1kErR28pd5SDkKOOBnqa+oAAqgKMADAA7UAZmv+JtF8L2P2zxBqUFhBnAaVup9gOTXjHxL8D6R8cfJ17wBrlnd6hYxi2mQuQrpksoORlSCW5xzmuI/al/tMeP7H7SX/ALONmv2YZO3dk7/bPSuP+DegeIfEvjJ9M8Oavc6NHNAwvbyAE7Yu44xyeg5FAHX+D9As/hF4qjvtSlXX/F6o0en6NpbGTyndSpaRxwBtYjH19q9StPhr4k+IN9HqnxZvttkjb4NAtHKxJ/vkdT+v0rt/Bfw48OeBbUrotnuupB++vpzvnlPux6D2GBXVUAV7DT7PSrCKx022itLWFdscMKBVQewFWKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACvG7D/k7bU/+wOn8hXsleN2H/J22p/9gdP5CgD2SiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAK53xP4/8ACvg1o08S61bWMsoDJE2XkIzjOxQWxnvjFdFXwv8AGpNUT4x+If7aDCVrotCSDgwY/dY9tm38c0Afbekazpuv6bFqGi3sN7aSjKSwvuB/wPsau184/smpqog8Qs28aQzRbN2dpn53bf8AgOM/8Bqh8c/jR4j03xpdeG/DF82nW9kFWeWJR5kjkBupHAGR0oA+nSQqkscAckntXl3ib4xK+st4Z+HFg3iPXy2xmj/49rb1Z36HH4D37V5F8PPE3i74y6knhLxD4pnttPhgaWdrZAk12oYfIzD2P5DnNfSvhfwlovg3R103w9Yx2kA5YqMtI395m6sfrQBwGjfByTWtQ/tz4rX/APwkGosDsswSLa3B/hUd/rxXoOgeFdC8LW8kHh7S7bT45DucQJjcfc1r0UAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABXL+ONT8T6LZ2mo+FtMh1aKCQm/si22WSLHWI/3ge3f0rqK53xn420zwRpcV1qSzTz3MnlWlpbxl5LmT+6oFACeEfHeieNLNpNJnK3EXE9nONk0Ddwyn+fSujrxWD4W6v4/8Qv4p8XlvDLyR7ILLS38ufb1zLIOp9q7Xwh8M7Pwfqz39trOr3rtGY/LvbtpEGe+D3oA7WiiigAooooAKKKKACiiigAooooAK8bsP+TttT/7A6fyFeyV43Yf8nban/wBgdP5CgD2SiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAK57xX4W8Ka7a/afF2m2FxHbjPn3SgeWOv3uoFZPj34paN4HVLTD6nrdxhbbS7T55ZGJwMgZx/M9q4/TvAXi/4lTxan8Vro2OmBvMg0C1O1cdR5pHfnpyfWgBH+Id5q/wDxSvwM0CJo4T5cmrSRiKztQDglRj5j1/LgNXOeJv2YdU1e3TUIvFn23XZvmvZL9CIpXP8AdKgsoHTkHPt0r3/S9JsNE02Kw0m0itLWEYSKJcAVcoA8w+D/AMG4PhnDc3l5eLf6vdoI5JI1KxxIDnaueTngkn07d/T6KKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKhltLea4hnmgjkmgyYpGQFo8jB2ntkelTUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFeN2H/ACdtqf8A2B0/kK9krxuw/wCTttT/AOwOn8hQB7JRRRQAUUUUAFFFFABRRRQAUUVl+J9XOgeE9V1dYzKbG0luAgGS2xS2P0oA1KK+B9Q+J/jPUdbfVZvEN8lyz7x5UxRU9AqjgD2r7H+FHi268b/DTS9a1KMJeSK8c5VcK7IxXcPrjPHAJI7UAdjRXjXxk+Oj/D3VI9E0Gxt7zVCiyzPdbjFCp6DapBJI56jHvXL+HP2mNb1vTZNPh8LJe+JZnCWcdpuEDA/xMCSwx6Z59RQB77rWu6Z4d017/W72Gztk6vK2Mn0HqfavDvHPxc8U6/4V1TU/Atq+k+HrKL97rN0u2ScswRVhHbJIGfftit6x+Gr3UqeKvjdrsN/cIC6ae8gSzs88454Ygfh/vda7bUrDw18SPAF/oelahaT6ddReUJLJldYmBDKQBxwQp/CgD488C/EvVPA/iM6vDbWt/cTPm5mu0LzSKT8wEhOVJ9fzzX2B4I+KPhfx8mzQ77/S1Te9pKNsijvx3/Cvkj4hfCHxH8Ogs+qCG5sZJCkV1bElT6bgR8pPoas/AnTNUvvi5o02lxSmK1m826kTIVYgOcn39KAPt6iiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAK8bsP+TttT/7A6fyFeyV43Yf8nban/2B0/kKAPZKKKKACiiigAooooAKKKKACvjn46fEfX9V+IOp6LBf3FppumzNbpBBIUDkcMzY659DX1J8QLvULH4ea5daMGN9FZu0O0ZIOOo/DNfAM0kk0zyTszyOxLMxySe+aAPQfg78NbX4l6/d2V7eXFpFaRLKzQxhtwJwQSentX1jqWt+FPhL4MtYLqeOwsLWLy7W3BzJKRyQo6sSTkn1OTXyV4K+Ld/8P9BurLw1pdpFeXY/e38xMj57YHAAHYevXNdN438FS6x8H9P+IX9r3niLU72RTeXExOLZBuDRqvYK/Hp6AUAYfjC6m+M3j6+1fw5Ba21w6okdhcXSxzTqiYLgthOAvI3ZwOhrovgXH4Y8HeLDrPi/xJptrd7GgtbZJvNAJ6u0iZQDjA5714nRQB75+1Pql9c+INChil36NLZfaLd43zHM5bkgjg4XZg/7XvXK/s73+r23xg0630oyNb3KSLexjOzyghO5vTDBcH1wO9eu/BPwba+MPgdDYeONON7aLfSvp4nyrRxFVwUIwQNxf8/SvTfB/wAO/DHgSOQeG9LjtpZhiW4Yl5XHHG48gcDgYFAHQ3llbahavbX1vFcwSDDxSoGVvqDVbStD0rQ7cw6Pp9tZRnqIIgue/OOtX6KACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAr5N1nUPCvib4oeKr7xvr2u6daperb2X9mA7digqzOdjADCp055NfWLZ2nb1xxmvDNW8eapBoviHwtr3gG8XV70yw2o02x3W90pGFcv+pPP4UAHiURvdeAPht4d128/sbUkeaa+SYGa4hUMwG8AdcHt/Kk0/UpvhR8SNZ8MW2o3V7pDaI2p2kN7L5jQyKG4B9Ds/X2qhL4L13wHY/D3xJLp1zqkmgJJFqFvaL5ksSSBvuqOuN5/KtHT9Ju/in8RNZ8UjSbzTdK/sVtNs21CExPM7Bvm2+nzdeelAHODQ9Sf4Nt8Uv8AhItVHiQH7YG+0fuQnm48vZ6bf19uK6DUNVn+KvxC0Dwxc6jc2Wltoi6neR2UvltNIwXgn0BasEanrifB9vhZ/wAIrrH/AAkBP2QSC2P2fy/Nz5nmdMY79Peuh1HRrz4W/EHQvFH9lXmp6XHoq6Zetp8RleJ1A+bb6ZXr70AZ1t4w1L4cx/ETwzHqU98mhwpPpcl0++SISKp25748wfiPeqmr6JffD/wJ4e+IWn6/qc+rzXFu2oLcXBaK4SUHcu3tzj8KvW/grVviFD8QvEr6Xcac2vRJDpkF6nlyOsYXDFT0zsH51U1bU9X8f+CPD/w9tfDGr2ep29xbjUZ7q1KQQJGDlt5656j9M0Ae93Wv6TYQQS6hqFtaLcIHj8+UJuGM8ZryTQr+01L9q3UrjT7mK5hOkIBJE4ZSQB3Feh+J/hx4a8YxWcfiCx+0iyTZD85XaPw+leXeC/DOl+Ef2nNQ0rQrf7PZppSuse4nBIBPJoA96ooooAKKK8G+Nnx11TwX4lHhzwtDCt1AiSXVzcR7wCwyEVenQgk0Ae80V4n8DvjVqHj3U7jQvEcMX9oRxGeKeBNqyKCAQR2IyK9soAKKxrrxh4asXuEu/EGmQyWq7p43u4w0Q/2lzkV5pf8AxB8T/Em4n0j4VWjW2nBjHceIblSqL6+WD1P6/SgDs/HPxM8OeCYDBqcpvNQlXEWmWoDzS54A29h7n9a8c0v4DXfxB1678SeI9Oi8IWFwN0Gl2Z3SuccM+eEGcZAGTzwvWvWvAnwp0fwb/p1yzavrsp3z6nd/NIW/2c/dH613VAHxRqH7P3xBs9cewt9H+2Rh9qXcUqiJ17Nknj6Gvqv4e+CIfBfw7sPDdw0d40aMblyuVkdyWbAPYZwPYCusooA4vVPhF4H1PSLywHh2xsxeJtee0hWORTnIIYDqCAa4Dw7+y5oGkeIk1DU9YuNUtIZBJFZvAIwSDkB2BO4ewAzXudFADIYY7eFIYEWOONQqIowFA6ACn0UUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFeN2H/J22p/9gdP5CvZK8bsP+TttT/7A6fyFAHslFFFADXdI42eRlRFGSzHAAr5z+LvwuT4keMZNa8Bazp1/qDIsV7Ym6UMCg2h1PpgAEev1rrf2l9S1TT/AIXoumM6Q3F2sd3IjYITBIH0JAFfJuha1e+HtctdV02d4Lm2lWRXQ+hzj3HHQ0AfUvwN+CupeA9UuNe8SSxC/khMENvA+4IpIJLHoScDpVj43/Few0rwxqXh/wAOXr3GtyR7JjaZb7LH/EWYfdODivGfHP7QXizxU9xa6Zcf2Rpco2iGBQJGHqX68+gOK9q+AEHhfXvhe7WujwR3Tu9tqnmHzGnbAJYsedrA5x25HbNAHx+GIcMeTnPPOa+5fhP448P+KPCNhZ6RPax3tpaoLiygUR+UcYJCgAYz6etfPP7QPwzsfA2vWN54bsHt9JvYSH+ZnVJgxyMnoCpXAz61k/AK01W4+MGkyaSHEcJdrtwDtEW0g7vqcAe+KAPtmiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigArxuw/5O21P/sDp/IV7JXjdh/ydtqf/AGB0/kKAPZKKKKAKeraRYa7pc2navax3VpOu2SKQZBrzvSv2fPAuleI5NVSykuEI/d2k774oz3IHf8eleoUUAfLPij9l7X28RTyeF7yxfTZXLoLiRkeIE/dwAc4r3X4YeAYPhz4Mi0eOYXFw7ma6n243yEAHHsMACuxooArajptlq9jJZapaw3dtIMPFMgZT+Bqro3hvRvDySLoemWtiJD8/kRhS31NadFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAV43Yf8AJ22p/wDYHT+Qr2SvG7D/AJO21P8A7A6fyFAHslFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAV43Yf8nban/2B0/kK9krxuw/5O21P/sDp/IUAeyUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFRzzxWtvJPcOI4olLu7HhQBkmvNbX48eG7jUYUey1ODTbifyIdWlt8WzvkjrnOOOuKAPTq8bsP+TttT/wCwOn8hXsgORkcivG7D/k7bU/8AsDp/IUAeyUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAch8V3nj+E/iNrTPnCyfbj9f0zXmPiqOJf2NtPMarvWzsXTjo5mjyfrya92vrKDUtPnsryMSQXEZjkQ91Iwa8rg+BZH2bS77xVqF14YtJxPDpDqoUEEkKW6kDNAGprfinxloPhjw8fDXhRtfknsYzdESbfKbYv/ANevHrXxl45T47XmrR+CWfW3sFjfS/M5SPHD5r6nVVRAqAKqjAA7CuDtvAt/D8cLzxk00JsZ7BbZYgTvDAdfpQByv/CzPix/0S2T/v8Amj/hZnxY/wCiWyf9/wA17PRQB4x/wsz4sf8ARLZP+/5o/wCFmfFj/olsn/f817PRQB4x/wALM+LH/RLZP+/5o/4WZ8WP+iWyf9/zXs9FAHjH/CzPix/0S2T/AL/mj/hZnxY/6JbJ/wB/zXs9FAHjH/CzPix/0S2T/v8Amj/hZnxY/wCiWyf9/wA17PRQB4x/wsz4sf8ARLZP+/5o/wCFmfFj/olsn/f817PRQB4x/wALM+LH/RLZP+/5o/4WZ8WP+iWyf9/zXs9FAHjH/CzPix/0S2T/AL/mj/hZnxY/6JbJ/wB/zXs9FAHgut/G/wCIPhyGCXXPh4tklxKIYmlnPzueiitP/hZnxXP/ADS2T/v+a6r4q+Br/wAc6bpNvps0MLWV+ly5lzyo7D3rvFGFA9qAPGf+FmfFj/olsn/f80f8LM+LH/RLZP8Av+a9nooA8Y/4WZ8WP+iWyf8Af80f8LM+LH/RLZP+/wCa9nooA8Y/4WZ8WP8Aolsn/f8ANH/CzPix/wBEtk/7/mvZ6KAPGP8AhZnxY/6JbJ/3/NH/AAsz4sf9Etk/7/mvZ6KAPGP+FmfFj/olsn/f80f8LM+LH/RLZP8Av+a9nooA8Y/4WZ8WP+iWyf8Af80f8LM+LH/RLZP+/wCa9nooA8Y/4WZ8WP8Aolsn/f8ANH/CzPix/wBEtk/7/mvZ6KAPGP8AhZnxY/6JbJ/3/NR3HxT+KdrbS3Fx8MGjiiQu7mc4VQMk17XVLWbN9R0K/soiFkubaSJS3QFlIH86APFtF+NHxF8RaWmo6J8OftlnISEmjnO0kdav/wDCzPix/wBEtk/7/mu3+FnhK88EeAbTQ9SliluIHdmeLO05Oa7CgDxj/hZnxY/6JbJ/3/NH/CzPix/0S2T/AL/mvZ6KAPGP+FmfFj/olsn/AH/NH/CzPix/0S2T/v8AmvZ6KAPGP+FmfFj/AKJbJ/3/ADR/wsz4sf8ARLZP+/5r2eigDxj/AIWZ8WP+iWyf9/zR/wALM+LH/RLZP+/5r2eigDxj/hZnxY/6JbJ/3/NH/CzPix/0S2T/AL/mvZ6KAPGP+FmfFj/olsn/AH/NH/CzPix/0S2T/v8AmvZ6KAPGP+FmfFj/AKJbJ/3/ADR/wsz4sf8ARLZP+/5r2eigDxj/AIWZ8WP+iWyf9/zWx8N/ifrni7xfqnh/xF4eTRrrT4FldPMLNknGDXp9cF4e8C3+k/F7xD4qnnha01OBI4o1zvUgjr+VAHe0UUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAV53N8cfBcOttp5u7ho0m8h71bdjbrJnGPMxj8eldZ4wnntfA+uz2eftEWnXDxY/viNiP1rxLRrK1b9iyd2iQu1tcSsSOS4uHAP1wBQB9CI6yRq6MGVhkEdCKWua+HM89z8NPD013kTvp8JfJ77RXS0AFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFAHDeN/i14f8B6vb6bqyXc1zPF5qpbQ78LnHNbHgrxpp/jrRJNU0mK5igjnaArcx7G3BVPT0+YV574l+H/jrUvizfeJ9AvtGtkFottbG7UzEDqQU2kAk9+a6X4QeNZvGfhW5bUbSG01LTrt7S8S3XCO64+cD3/mKAO+ooooAKKKKACiiigAooooAKKKKACiiigAooooAK5iP4g6FP8AEAeD7adptT8hpn8sZRAOqk+vtW9qNkupabcWTyyQrPGYzJE2GUEdQexrwvQPC+meEf2o7LTdGiZIRo8ju0jl3kcjlmY8kmgD07xf8TfD3gu9hstTkuJ72Zd621pA0rhf7xA6Ctnwz4n0nxfocWraDdC5tZCVzghkYdVYHkEehrzXwZuu/wBpbxw94gLW1pBHAWHRCBnH5mm/A0tD4u+JNlCgSyg19/IVfugl5AQB9FWgD2KiiigAooooAKKKKACiiigAooooAKKKKACiiigDm/GfjvRPAunw3WuTlTPKsUUMY3SOSeoHoPWrfiLxTp/hjwlceItTLrZW8ayNtGWIYgAAeuWFeN/H7wRpWn6LJ4m/f3Gp3epwL5k8pYQoT9xF6KOK7T4p+E9b8bfC+y0Pw9JaxyTyQPObqUopjUZwMA5O7afwoAl8KfGrw74v1+DSdMtdSSecEo81sVTAGeteiV474f13xL4M+JWkeEfGI0m8g1WBvsFxYW/lmEoPuHgZGB1NexUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAyaKO4geGZA8cilXU9GBGCK8c/4UhrEWnyeGrTxdJH4PknMpsDDmUKW3FA/pmvZqKAILK0h0+xgs7VAkMEYjjUdgBgVPRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAeZ678NPECeL77XvBHik6Q+pKBdwTxeamR/Enoa6XwB4ItPAXhkaXaTPcyyStcXVzJ96aVurfoK6eigAooooAKKKKACiiigAooooAKKKKACiiigAooooAK4eTwDcyfGmHxuL2IW8dgbU22w7ySMZz0xXcUUAed+Kfhxqt34wPijwXrw0XU54Bb3XmQ+ZHMo6HHqK2fh74Gh8CaDNa/anvr68uGur28kGGmlbqfp/8AX9a6uigAooooAKKKKACiiigAooooAKKKKACiiigAooooA4z4oeB7jx94Wi0q0vIrR47uOcySqWBCnpxS+OPAH/CYeG7Czi1KbTtQ02SOe0vIf4JFA6juMgH8BXZUUAec+Gfhtq0XjCHxP4418a3qNnEYbNYofLjhB6nHrXo1FFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFAGb4h0661bw/eWOn6hNp11NGRDdQHDRP1B+meo9K8oX4vanZeEZfD93b+Z4+iuBpkdp/z1kI+Wf/dx82a9d1bVbTRNIutT1KZYLW1jMksjHgAV4I2m+KNRvG+M1vbATwybrfS2iG6SwAwTnrvIyRQB7d4T0q/0XwvZ2Ws6nNqmoKm65upmyXc8nHoo6AegrZrO8P67Y+JdAtNY0mYTWl3GHRh29QfQg5BHqK0aACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigDK8ReGdJ8WaYun6/bNdWglWUw+c8asy9N20jcPY5HtWmkUccKxRoqxqu1UAwAPTFOooAyPD3hbR/CtvcwaBZ/Y4LmdriSJZXZN7dSqsSFHsuB7Vr0UUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQB//Z)

Linear regression performs the task to predict a dependent variable value (y) based on a given independent variable (x). So, this regression technique finds out a linear relationship between x (input) and y(output). Hence, the name is Linear Regression.  
In the figure above, X (input) is the work experience and Y (output) is the salary of a person. The regression line is the best fit line for our model.

**Hypothesis function for Linear Regression :**  
Chart

Description automatically generated with medium confidence

While training the model we are given :  
**x:** input training data (univariate – one input variable(parameter))  
**y:** labels to data (supervised learning)

When training the model – it fits the best line to predict the value of y for a given value of x. The model gets the best regression fit line by finding the best θ1 and θ2 values.  
**θ1:** intercept  
**θ2:** coefficient of x

Once we find the best θ1 and θ2 values, we get the best fit line. So when we are finally using our model for prediction, it will predict the value of y for the input value of x.

**How to update θ1 and θ2 values to get the best fit line ?**

**Cost Function (J):**  
By achieving the best-fit regression line, the model aims to predict y value such that the error difference between predicted value and true value is minimum. So, it is very important to update the θ1 and θ2 values, to reach the best value that minimize the error between predicted y value (pred) and true y value (y).

![A picture containing text, clock, watch

Description automatically generated](data:image/jpeg;base64,/9j/4AAQSkZJRgABAQEAYABgAAD/4RDaRXhpZgAATU0AKgAAAAgABAE7AAIAAAAFAAAISodpAAQAAAABAAAIUJydAAEAAAAKAAAQyOocAAcAAAgMAAAAPgAAAAAc6gAAAAgAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAGRlbGwAAAAFkAMAAgAAABQAABCekAQAAgAAABQAABCykpEAAgAAAAM4NAAAkpIAAgAAAAM4NAAA6hwABwAACAwAAAiSAAAAABzqAAAACAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAMjAxODowOToxMSAyMjo0MjozMgAyMDE4OjA5OjExIDIyOjQyOjMyAAAAZABlAGwAbAAAAP/hCxdodHRwOi8vbnMuYWRvYmUuY29tL3hhcC8xLjAvADw/eHBhY2tldCBiZWdpbj0n77u/JyBpZD0nVzVNME1wQ2VoaUh6cmVTek5UY3prYzlkJz8+DQo8eDp4bXBtZXRhIHhtbG5zOng9ImFkb2JlOm5zOm1ldGEvIj48cmRmOlJERiB4bWxuczpyZGY9Imh0dHA6Ly93d3cudzMub3JnLzE5OTkvMDIvMjItcmRmLXN5bnRheC1ucyMiPjxyZGY6RGVzY3JpcHRpb24gcmRmOmFib3V0PSJ1dWlkOmZhZjViZGQ1LWJhM2QtMTFkYS1hZDMxLWQzM2Q3NTE4MmYxYiIgeG1sbnM6ZGM9Imh0dHA6Ly9wdXJsLm9yZy9kYy9lbGVtZW50cy8xLjEvIi8+PHJkZjpEZXNjcmlwdGlvbiByZGY6YWJvdXQ9InV1aWQ6ZmFmNWJkZDUtYmEzZC0xMWRhLWFkMzEtZDMzZDc1MTgyZjFiIiB4bWxuczp4bXA9Imh0dHA6Ly9ucy5hZG9iZS5jb20veGFwLzEuMC8iPjx4bXA6Q3JlYXRlRGF0ZT4yMDE4LTA5LTExVDIyOjQyOjMyLjg0MDwveG1wOkNyZWF0ZURhdGU+PC9yZGY6RGVzY3JpcHRpb24+PHJkZjpEZXNjcmlwdGlvbiByZGY6YWJvdXQ9InV1aWQ6ZmFmNWJkZDUtYmEzZC0xMWRhLWFkMzEtZDMzZDc1MTgyZjFiIiB4bWxuczpkYz0iaHR0cDovL3B1cmwub3JnL2RjL2VsZW1lbnRzLzEuMS8iPjxkYzpjcmVhdG9yPjxyZGY6U2VxIHhtbG5zOnJkZj0iaHR0cDovL3d3dy53My5vcmcvMTk5OS8wMi8yMi1yZGYtc3ludGF4LW5zIyI+PHJkZjpsaT5kZWxsPC9yZGY6bGk+PC9yZGY6U2VxPg0KCQkJPC9kYzpjcmVhdG9yPjwvcmRmOkRlc2NyaXB0aW9uPjwvcmRmOlJERj48L3g6eG1wbWV0YT4NCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgPD94cGFja2V0IGVuZD0ndyc/Pv/bAEMABwUFBgUEBwYFBggHBwgKEQsKCQkKFQ8QDBEYFRoZGBUYFxseJyEbHSUdFxgiLiIlKCkrLCsaIC8zLyoyJyorKv/bAEMBBwgICgkKFAsLFCocGBwqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKv/AABEIAFsBUQMBIgACEQEDEQH/xAAfAAABBQEBAQEBAQAAAAAAAAAAAQIDBAUGBwgJCgv/xAC1EAACAQMDAgQDBQUEBAAAAX0BAgMABBEFEiExQQYTUWEHInEUMoGRoQgjQrHBFVLR8CQzYnKCCQoWFxgZGiUmJygpKjQ1Njc4OTpDREVGR0hJSlNUVVZXWFlaY2RlZmdoaWpzdHV2d3h5eoOEhYaHiImKkpOUlZaXmJmaoqOkpaanqKmqsrO0tba3uLm6wsPExcbHyMnK0tPU1dbX2Nna4eLj5OXm5+jp6vHy8/T19vf4+fr/xAAfAQADAQEBAQEBAQEBAAAAAAAAAQIDBAUGBwgJCgv/xAC1EQACAQIEBAMEBwUEBAABAncAAQIDEQQFITEGEkFRB2FxEyIygQgUQpGhscEJIzNS8BVictEKFiQ04SXxFxgZGiYnKCkqNTY3ODk6Q0RFRkdISUpTVFVWV1hZWmNkZWZnaGlqc3R1dnd4eXqCg4SFhoeIiYqSk5SVlpeYmZqio6Slpqeoqaqys7S1tre4ubrCw8TFxsfIycrS09TV1tfY2dri4+Tl5ufo6ery8/T19vf4+fr/2gAMAwEAAhEDEQA/APpGiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKxvEHi/w/4Vtmn8Qava2IVd+ySQb2H+yg+ZvwFa0MqTwJNEdySKGU46gjIoAfRRRQAUVwXxcutUsfCaXOj6rNp8/nxwwC3OHlnkdUjBPdRlmK99oHTNUvDl5qGnfFi70a68QXmo2sWnQrdNeMpEl85ZgI1AAT92jHauBj35oA9KooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKM0AFFFFABRRRQAUUUUAFFHXpRQAUUUUAeUftA6FpMnwt1nVpNMs31JEiVbxoFMqjzBwHxkDk9+9emaT/yBbL/r3j/9BFedfHG4u9U8Cal4a0nRNX1C+ukjeNrWxeSL74JBcDGeOnXpXZ+EtaGr6Oif2dqVg9rHHE6ahZvASdo+7uHzD3HFAG7RRXyl8UIPEXw2+Jtvdanrevah4Xv5zMkaalNGdufniBDDDLnj1GPegD6h1HSbHVhbDUbZLgWs63MIf+CRfusPcZqBPDukx68+tJZRjUJBhp8nJ425xnGcDGcZxxXI6P4E8Ma9o1rqmlaz4gntLqMSRSLrt1yD/wAD6+1WZPhRo8ikLrHiWMkdV1y4yPzY0AdxRXiXi74M+J7WA6j4D8c6813bkSJY39/I6ykHOA2f0YEH2r1Twlr3/CS+FbHVGiMM00eJ4TwYpVO2RD7hgR+FAGzRVC21uxu9au9KtpTLdWSK1wFUlYt33VLdNxHOOuOav0AFFFFABRRRQAUVUvdW07TiBqF/a2hbp58ypn8zVugAooooAKKKKACiiigAoqhe63YadqdjYXs3kzagzJbblIWR1Gdm7puxyB1ODjpWd4w1HWNP06zXw/ZyXFxd30VtJJHD5v2aJid0pXIyBj14zn2oA6CiuG8EeKNXv/GHibwzrrwXcuiyRGO9gi8sSpIu4BlyQGHTiu5oAKKKKACoL++ttM0+e+v5lgtreNpZZX6IoGSanqhremWOsaPPZ6vEJbNwGlQsQGCkNzjtkCgDm7T4seE755IbO7upryNyj2KWMxuFwAcmPbuC4I+Y8c468UR/Frwfc6d9tsdQmvYliM0otbOaVoEBIJkVVynQ/eweOK4f4Y3iaV8O/GHxFuYl8/U7q6vI3I5aKMkIufTdurP04t4H/ZUe6igzqmux8vgbpJbltqsx9lI6+lAHsFz4w0S28O2mtteebY3uwWrwRtI05cZUKigsSfTHGDnpXN+Dbvwn4c8G3fiLSdXvr/TtVvWnM1yGkkMrPs8pEChvvDG3Ga5bW9U0/wAK/BZzYJJPf+F4F0y0ugcp9rkiVHdccHbvI56HIqzpH9jeHfg7PbiFppfBsZmNwWxG9+Yi7EEH5irSkHPQ/SgDqovi54PuNOa9tL+4uooozLcC3sppGtlBIJlAXKdD1x0rbvPFujWWg2msPdmWzvtn2RoI2ka4LjKhFUEkkc9OgOa8dRW8C/sp28NpBt1LxBGsWcYLy3JxuY+oj/kK0/FGr6boHwaaTR0lnvfDsQ0SwulbKG4eNIndMHBK5YZPQggUAdV4Ou/CfhjwLJruk6tfX+mateNcLPcb5ZZJXbZ5aIFDZyuNuM8GrMXxe8HXGlnUbPUJ7u2jjMs7W9lNJ9mUHBMmF+ToeDzxnpXJ2v8AYug/Bm4jht2kfwVG3lXOf3bXxhO9lwfm2tMwPvn0rGuoD4E/ZMgtLWFYr/WreOIjGC8lyRuz7iMkfgKAPSrj4q+ELaEXLam8lluRHvYbaV7eNnxtDSBdoPI78d8VZ8b+KNA0TRZrXX7+4tkvbaT5rOJ5JUj24aQbFO0DI+Y8ZxXnvxG0e1034f8Ag74bWCLaLq1zDBM6gKqRRAPM5P8AeJOfc5pPilqtnr/hrRbfRYZILnxbMumC5lbGyyjkLMw7ANgHPdTz0oA9E8PjQPBfw4tJYNTd9DtLUTrfXUm4vG3zBycDOc8ADuABVTQ/iz4N8RrF/Y+ri4lmuVtUt/KdZS577CN23GSWxgAcmqGp+MfCGm/C+3vrixW40YMLfTrJ4g/2rymxGUU5G35AwY9AM1m+Fte8OXXi+y1C71fT/EHivVVaJE0tllj023ALFeD8oHALN8zE8DHAAO21nxnoehajDp17dPJqEy747O1geeZl/vbEBIHucCrmh69p/iLTje6VK0kSyPC4kjaN43U4ZWVgCCD2IrxzwBo9zqvxB+IGrXXiO40m+XUjbSeWsPmeSo+U5kVioxgDjHy16V4RtdEm8L3Ft4X1O6mtzdSGXUQ++SebdueQOy7XyeNwGPSgDqqKxv7An/6GDVv++4v/AI3UkGizQ3CStrepTBGBMcjR7W9jhAcfjQBq1z3jjwdp/jvwnd6HqYAWYboZtuWhkH3XH0/UEiuhooA+Yfg/4y1D4XePLv4d+M5Gjs5J9lvIx+SGU9GB/uPx9CQeOa+nq8O/aB8H2PjDUtB0/SE3+LbiTZEI1/5dRks0p/hRScgnvkDrXsHh7TrnSPDen6ffX0moXNrbpFLdSjDSsBgsf859SetAGjWT4m1ZPDnhLVdX2qBZWstxjHBIUn+da1cd8W4XuPhD4mjiUs39nyNgD0GT/KgCL4b2CeGfhjbahrNxi5u4Tqmp3Ux5LyDzGLH/AGRgf8BpYPi34Ruppbe1u7ua8i5ezj0+dp1XaG3FAmduCDu6cjvWrY2mneI/h5ZW93GJdOvNPhLIrFQyFFOMjHGK8p+HPibTdM8L+PvHt1PAZrrUphEm8FiqLiKMDryWwBQB6Rd/FLwjZaLZ6tJqhksbyMyxywwSPtQMFLOAuUAZgDuxzxU2ifEfwv4i14aPpGoNPdtAbiMGCRVljBwWRioDDPcHntXhE3hKXRPgn4a0a8tmi1vxTfQ2kitkMluZzME29uSCfrz0r32XwnpOnXEut6bpsa6vb6cbSCdQchFX5UA6dh2oAi1D4h+HNO1C5sjdzXdzZqWuo7G1luPs4HXeUUhfoefart54v0HT/DMPiC81OGLS50V4bg5/eBvuhRjJJ9AM15D8J/D9t/wqM6rqnjG702HUZJ59UWJ4YiH3srBpCpcHCjjIPPGKs3kWm6z8Wvh5ptqjQ+FbDSn1CwiuI2jEkiZVMh8HKgI3PPfvQA3xvqmiePfip4I0Sws3N1De/bb17mzeGZYYl3ovzqCVbJ/Kvca8R0/xHpmp/tBaz4hilF7a6ZDbaLafZ8MZJ5n+Yr6hdr5I7Zrs/BmuWnifxtrt8dGlsNQsoLe0eV52fdGwMqoVIARxv5HOM4z2oA7uioL23kurKWCG5ltJJFKrPCFLxn1G4EZ+oNc//wAIpq3/AEO2t/8Afu1/+M0AdPUV1dQWVnNdXcqwwQIZJJHOAigZJPsBWNp3h7UbK/juLjxTqt9Gmd1vOluEfgjnbGD78EdK0tW0u01nSprDUYzLbTACSMMV3AEHGR246d6AM3w7400TxVPewaPcSvNY+WZ45rd4WQOu5Gw4HDDkGruheINM8TaSmp6HdC6s3d41lCMoJVirY3AHqDz3rwzQPEqaR8MfiH42u7qOHVtWvp7eFWcB0KDy4UA/2d3A9BWz4fey0rQ/AfgfUtCuLhWkhmE4maHZceW05YLj94q5+Y5wCw6kHAB6L8Q/DyeJfAupWeTHcxx/abSZeGinj+eNgf8AeA/AmsE+PNZu/gzpfiPw7pE+r6tfQRoIbePf5cmCHdhkZCsrcdzgV3moyLDpd1JJwiQuzfQKa4j4HQvD8GtD8xNnmCaVQf7rzOyn8iKAIfhSLyNNQjuPDeoaQp2Sz3eqoBc6hcMW3yMQcYACgAdM4HFei0UUAFFFFABTZESWNo5FDo4KsrDIIPUGnUUAZ/8AYOk/2F/Yv9mWg0vy/L+xCFfK2/3dmMYp93o2mX2lDTL3T7W4sAqqLWWFWjAX7o2kY4wMemKu0UAZzeHdGfRDoz6TZHTCMGzNuvldc/cxjrz9aX/hH9H/ALEbRv7Ls/7MZdrWfkL5RGc424x15rQooApX2jaZqmmjT9S0+1u7IbcW80KvGNv3flIxxjj0qOTw9o02ijR5dKsm0wAAWZt18oAHI+TGOvNaNFAGe3h/R30M6M2l2Z0wrtNkYF8ojOcbMY681JfaPpupwwQ6jYW91FbyLLCk0QZY3X7rAHoR2q5RQBn6toOka9DHFrml2eoxxNujW6gWQIcYJG4HHHFN1Lw7ous2cNpq2k2V7b25BhiuLdXWM4x8oI44447VpUUAVDpdg01rKbK3MlmjJbN5YzCrAAhfQEADj0qvpnhrQ9Fup7nR9HsbGe4JaaW2t0jaQ9eSBzWnRQBj6h4R8O6tffbdT0PT7u6ICmaa2R2YDoCSOR9a1YoY7eFYoI1ijQYVEUAKPQAU+kZdylckZGMigDl5PiX4OhvLu1m1+0jmswxlVyRna21gpx85DcYXJzxWhoPizRPEz3UejXyzy2bBLiFkZJIiem5WAIz9K8z8MeG9Nv8A493yW1uH07wdp0VpbCT5sXMpMjPnu3LZPqc0vw11m3u/iN4n1+48zPiDVDp+nIiZ3pbRndIfReAM+vFAHslc1418ZQeEtNiEUJvdXvn8nTdOQ/Pcy/0UZBZugFdLXjWt/Cv4gan4+n8VWfjazs7oq0Nsost4toSfuqGyAeOT1PPrQB23gbwbLoIutY16YXviTVSJL+66qnpDH6IvQevX0x19eQ/8IH8X/wDoqEH/AILk/wAKZJ8Pfi7KhVvijGAf7tiqn8wKAPVNY1jT9A0mfU9Yuo7Szt13SSyHAH+J9u9ZmgaknjbwUl5eWRt7XVIpAkLNlmgYkKx4GCy4OO2eprxrVP2ePF/iS7t28U+P21KCOQFkkSQ4HfaCcA4zzivoC1tobK0htbWNY4YUEcaKOFUDAH5UAeVWK+LfD3w81TwRp+l3GoarZRyWum3e4RxPauGEUu9uNyDClOuQO3Na3w08BadpfhHS4tb8J2dpq2noIzPPFDLJI45MgZc92bGTkV6JRQBVutLsL66tbm8s4LiezcvbSyxhmhYjBKk9Dj0q1RRQBif8IZ4YOoNfnw9phumfzGm+yJuL/wB7OOvv1q3q2gaRr0UcetaXaX6RNujW5gWTYfUZHFaFFAGTB4V8P2upR6ha6Jp8N5EgSOeO2RXRQMAAgccEisjwh4Il8L6pql5LqrXYv7ma4KCMplpHBLP8xDMAqqDgYAPHJrraKACiiigAqhrw1M+HdQGgmMap9mk+xmX7ol2nZn2zir9FAHhP/CudX8dR6LZ+JfDFnpvlutzr2ry28EVxfSj/AJZxrEThT0L8EgdB0PpeueDJtW8YafrUGqG0js7Y2xhSIlgpkV2MbhhsLBQhOD8pOMZrq6KAOP8AiM2q32gjw54eik+260TbPchD5drAcCWRm6A7SQB1JPHSul0rTYNH0ez02zXbb2cCQRjHRVAA/lVuigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAzdL8P6Xot3qF1plosE+pT/aLtwxJlk9Tk/oKq6P4M8P6Bqc+oaTpsdtdT7gzhmOAzbmCgkhQW5IXGTW5RQAVi+KdWn0zSSul3WmQ6rcMI7KPUpvLjlckfL1BY4zwOa2q8x+K9nNJrXh57XRrrUC9yJJ5IEZ/lg/exw5GRH5ku3LcDC89KAPTQflG7Gcc4oz6cmuO1fw3e6lqkt0fDvh658wL+9urmTzCdoznEZHByBz0A+lP0Pw3dadq8VzL4f0CzRAczWc8jSrx2BjA/WgB3h3xVq2qeOtb0LVdNt7FNOt7eaLypzM0nmF+ScAD7o4x+Jrra840HUg3xk1+7ax1KO2vrS0t7eeTT5lRnjMm4biuABkcnivR6ACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooA//Z)

![A picture containing text, clock, watch

Description automatically generated](data:image/jpeg;base64,/9j/4AAQSkZJRgABAQEAYABgAAD/4RDaRXhpZgAATU0AKgAAAAgABAE7AAIAAAAFAAAISodpAAQAAAABAAAIUJydAAEAAAAKAAAQyOocAAcAAAgMAAAAPgAAAAAc6gAAAAgAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAGRlbGwAAAAFkAMAAgAAABQAABCekAQAAgAAABQAABCykpEAAgAAAAMwMAAAkpIAAgAAAAMwMAAA6hwABwAACAwAAAiSAAAAABzqAAAACAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAMjAxODowOToxMSAyMjo0Mjo1MQAyMDE4OjA5OjExIDIyOjQyOjUxAAAAZABlAGwAbAAAAP/hCxNodHRwOi8vbnMuYWRvYmUuY29tL3hhcC8xLjAvADw/eHBhY2tldCBiZWdpbj0n77u/JyBpZD0nVzVNME1wQ2VoaUh6cmVTek5UY3prYzlkJz8+DQo8eDp4bXBtZXRhIHhtbG5zOng9ImFkb2JlOm5zOm1ldGEvIj48cmRmOlJERiB4bWxuczpyZGY9Imh0dHA6Ly93d3cudzMub3JnLzE5OTkvMDIvMjItcmRmLXN5bnRheC1ucyMiPjxyZGY6RGVzY3JpcHRpb24gcmRmOmFib3V0PSJ1dWlkOmZhZjViZGQ1LWJhM2QtMTFkYS1hZDMxLWQzM2Q3NTE4MmYxYiIgeG1sbnM6ZGM9Imh0dHA6Ly9wdXJsLm9yZy9kYy9lbGVtZW50cy8xLjEvIi8+PHJkZjpEZXNjcmlwdGlvbiByZGY6YWJvdXQ9InV1aWQ6ZmFmNWJkZDUtYmEzZC0xMWRhLWFkMzEtZDMzZDc1MTgyZjFiIiB4bWxuczp4bXA9Imh0dHA6Ly9ucy5hZG9iZS5jb20veGFwLzEuMC8iPjx4bXA6Q3JlYXRlRGF0ZT4yMDE4LTA5LTExVDIyOjQyOjUxPC94bXA6Q3JlYXRlRGF0ZT48L3JkZjpEZXNjcmlwdGlvbj48cmRmOkRlc2NyaXB0aW9uIHJkZjphYm91dD0idXVpZDpmYWY1YmRkNS1iYTNkLTExZGEtYWQzMS1kMzNkNzUxODJmMWIiIHhtbG5zOmRjPSJodHRwOi8vcHVybC5vcmcvZGMvZWxlbWVudHMvMS4xLyI+PGRjOmNyZWF0b3I+PHJkZjpTZXEgeG1sbnM6cmRmPSJodHRwOi8vd3d3LnczLm9yZy8xOTk5LzAyLzIyLXJkZi1zeW50YXgtbnMjIj48cmRmOmxpPmRlbGw8L3JkZjpsaT48L3JkZjpTZXE+DQoJCQk8L2RjOmNyZWF0b3I+PC9yZGY6RGVzY3JpcHRpb24+PC9yZGY6UkRGPjwveDp4bXBtZXRhPg0KICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICA8P3hwYWNrZXQgZW5kPSd3Jz8+/9sAQwAHBQUGBQQHBgUGCAcHCAoRCwoJCQoVDxAMERgVGhkYFRgXGx4nIRsdJR0XGCIuIiUoKSssKxogLzMvKjInKisq/9sAQwEHCAgKCQoUCwsUKhwYHCoqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioq/8AAEQgAWQEiAwEiAAIRAQMRAf/EAB8AAAEFAQEBAQEBAAAAAAAAAAABAgMEBQYHCAkKC//EALUQAAIBAwMCBAMFBQQEAAABfQECAwAEEQUSITFBBhNRYQcicRQygZGhCCNCscEVUtHwJDNicoIJChYXGBkaJSYnKCkqNDU2Nzg5OkNERUZHSElKU1RVVldYWVpjZGVmZ2hpanN0dXZ3eHl6g4SFhoeIiYqSk5SVlpeYmZqio6Slpqeoqaqys7S1tre4ubrCw8TFxsfIycrS09TV1tfY2drh4uPk5ebn6Onq8fLz9PX29/j5+v/EAB8BAAMBAQEBAQEBAQEAAAAAAAABAgMEBQYHCAkKC//EALURAAIBAgQEAwQHBQQEAAECdwABAgMRBAUhMQYSQVEHYXETIjKBCBRCkaGxwQkjM1LwFWJy0QoWJDThJfEXGBkaJicoKSo1Njc4OTpDREVGR0hJSlNUVVZXWFlaY2RlZmdoaWpzdHV2d3h5eoKDhIWGh4iJipKTlJWWl5iZmqKjpKWmp6ipqrKztLW2t7i5usLDxMXGx8jJytLT1NXW19jZ2uLj5OXm5+jp6vLz9PX29/j5+v/aAAwDAQACEQMRAD8A+kaKKKACiiigAooooAKKKKACiisnxF4n0rwrp63mt3Dwws2xfLgeVmbGcBUBNAGtRXO+CPGdj478PHWdKhmitjPJConADNsOM47Z9K6KgAoorN8QwQXPh2+jvbiW2tfJLTyRNtYRjlhntkAjPvQBpUV87W9lZ6L4W8Cai0U1vf6hqLaiZjIxNpaoHk+zxAnChlKqEHDE+9fQ8bb41faV3AHa3UexoAdRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRWRr3irRfDEccmvXy2aSBirurEYXGSSAcdR1oA16KyPD/irRfFVvJceH7+O+hjIDSRq23Jz0JHPQ9K16ACiiigAqO4H+jyf7h/lUlZHiS91ey05ToOi/2vPI2xovtSQbFIPzZbr24HPNAHCfs9f8kub/ALCV1/6HXqVeW/B/SPF3hDSV0DXvDkcNs9zNcNfx38bhN53Bdg5PPGa9SoAgmvrS3fZPdQxMP4XkAP61E2qaY6lWvrRlIwQZlwf1rxb9ov4XNr+mf8JfocDPqVhHi7iQZM0I/iA9V5+o+gq18FdQ8GfELwoI73w5o6a1p6rHdxi0T94OglAx0Pf0P4UAeuNe6O6or3NiwjOUBkT5fp6VJ/a2nf8AP/a/9/l/xrHb4e+Dm+94Y0o/9uif4VzviP4E+AfENm8Y0WPTJzytzp/7plP0+6fxBoA9DSRJY1eNldGGVZTkEfWnVwfwr0TV/B+k3nhLV3e6g0uXdp99tIWaCTLBfZlbcCO3HbFdFfeJre18T2OgW8El3f3SmaRIiALaEcea5PQE8AdSfoaANqiiigAooooAKKKztd8QaZ4a003+t3QtbVW2mQozAHBPYH0NAGjRVHRdasPEOj2+qaRP9osrkFopdjLvAJGcEA9RV6gAooooAKKKKACiiqGuam2jaLc6ilnNe/Zl3tBBjeyg/MVB6kDJx3xigC/RWXFqy634VOp+GJ4rr7VatJZSE4R2KnbnPTnGc9K80sp/Evh34g+C9OuNWvLy91a0lfXdPmuPPjiKqD5qnHyDcSMLgcACgD1+iiigAooooAKKKKACvNvj3rUmk/CPUYLY/wCk6o6WEQHfzG+Yf98hh+Na3xU8Yan4H8F/2tothFeT/aY4W89iscKsSN7njAzgZyANwrhfFGoQ/Ff4g+FND0CWLUNP0iddR1m7tX8y3jYAFYw44YnDD8fY4APVvCWjx+HvBukaREMLZ2cUR9yFGT+Jyadr/ibSfDFvDPrd01tHM+xGELyZbGcfIDjgd61aztYj1mSGMaBcWUEob94byFpARjsFYc0Ac9/wtnwX/wBBhv8AwCn/APiKv6N4+8N+INSWw0jUGuLllLBDbSpwOvLKBUX2bx3/ANBLQP8AwBm/+OVc0uHxSl8p1m90ma1wdyWtrIjk9uWcj9KANyiisD/hCdB/59Jf/Aub/wCKoA36KwP+EJ0D/n0l/wDAub/4qtyKJIIUijGEjUKoJzgDgcmgBxAZSGGQRgg96+WfiN4b1D4IfE6z8a+E4m/se7lO+EE7EJ5eBiOityV9Mf7NfU9cJ8XL/Sz4Lm0G9tP7S1DWgbbT9PRsSTS9mHoEOGLdB+NAHSeFvEun+L/DVnrekSb7a6TcBnlG7qfcHiteuH+E3w7j+HHg1NOa4e4vbhhPdvvJQSEYwg6ADpnqe/bHcUAFecfClP7W1LxZ4ruPnn1DV5LWJic7YIMIij8dx/KvR687+DMnl+GtZ02TibTtdvYZFPXmTeD+IcUAR3fxjtIPEkGkWmg6jqD30k0Ng9sUP2l4n2OcEjam7PznjCk9KuRfFG3+1appl3pNxb65p1xBb/2d5yOZmnwYyrg4wRyc9ADmuGbxP4c0r9pHU7nWbq302w0PShaWxK7YxK37xlGBjcQzcdTzTtLj1DT9K+IHxQ1uza2k1BPM0qG5jw8ccaFIpCp5UnI9+vrQBtxfHvTpbW1uh4f1EW02oDTZJy6bVnJPCHPzgAZJHqK67XvHlnpPiO18OafZ3Gr65cp5q2VsVXy4/wC/I7EKo/HPt0qj8J/DkWjfCfw9Z3dshlEAuWEiZKyOd+eeh+avPrnTvD2nfHHxdefEPUrnSEvIrd9LuP7QmtFniCYkUOjLuwQvy5/CgD1DS/Hdlc6frlzrMDaQdBlMV8JZFkVPkDghl65DDjrnjGa88+JXxOvb/wCFM0uneH761tPEI+xWF5NIgLbzjJQHcoZN2Kg8f2ljqHwRvLbwhot9p2hx38Ek87RMrXdvuBknwxMjgcHc3JAz05qt8RPGXhzWdc8F2Wm3Bk8Oade/aJ72OJvsrNFHujgV8YZzjG0dNwoA9i0SCw8K+GtE0aW4htvLgis7dZZApldU+6ufvMdpOBz1rarzS08T65qfxD8PaXq+lWMXnWb372zxCSS0ARQJQ5Pyku7R4x/Ceea9LoAKK5O5+Gnh67u5rmY6t5kztI+zWrtFyTk4USgAew4FRf8ACrPDX97Wf/B7ef8Ax2gDsa4i0+J1neeNNP0KPTrjydTkuYrO+3qUmNuP3hC53Bc5Abvj05rtUQRxqi5woAGTk/ma8Vv9W0bwl+0DqGp+KpodJ0+w0NI9JDJsSQFsyCMDgtksMDnmgD0vTfGNhqfirXtEiR0bQkha6uXIEeZFLYBz2A5zW3aXltqFnFd2NxFc20yh45oXDo6nuCOCK+fvC2varpfwl8R+N7KCF5de1S5uWa6j81SgykcWzILbnITHYZPbFe9aNE0Gh2Mb2sNo626BreBQscR2jKqBwADxQBwPw8nl0jXPHPhTTokI0q9+1afC7bUVLhC6x57KHU/QGq3g7QPiHZ+JVvPEVroyG6nMuo6jBcvJPMihvLhVSoCoMgYHoT1JNXPAiPdfF/4hakg/0cTWlmG7GSONiw/Dev516RQAUUUUAFFFFABRRRQAjosiFJFDKwwVYZBFMht4baPy7eJIkznbGoUfkKkooAKKKKACiiigArifFPxU0LwjrS2OqQ3rRI8Ud1ewxBobVpMlFc53EkKThQcDHrXbV5N8QtMs9Z+Kvg3w1Bbxqs95JrWpbV/1ghQKm71yV2/hQB0tj8TLG68Uafot5o2raY+qrI+n3F5Cix3ITBOAHLLkEEblHB5xXaV4r4x1oJ8e4Lq4hM6eG9NH2OzDhZLy6uW2oiA9fc9tvNe0qSUBYbWI5Gc4NAGL4t8V6f4O0GTU9SLOciO3t4hukuZTwsaDuSfy6msLwT4Vvm1GXxh4yVJPEN8mIoRymmwHkQJ7/wB5u5qHx38JdP8AiBrFpqGq67rVqbIf6NDZTxokLZyXXKEhjgc57CsX/hQVj/0PPjP/AMGa/wDxFAHrFUNW1zS9Bsnu9Z1C3sYEGWeeQKP1615q37P+nt97xt4xP11Jf/iKy5v2WvCNxK0k+ueIJXY5ZnuIiSfcmKgDuvh/43fx9Jq2p2EWzQoZxbWLuhEk5UfvJDz90kgAdeOfQQT+HtX8O+PrzWvC9vHcWuvRql9A7hVt7hQdk+MjKkfKwHPcZrofCXhfT/Bnhez0HSBJ9ltFIVpSC7kkksxAAySSelbNAHA/DvwXrvhTUNbn1y706+OsXj30ssCOHEjYwo3cbAM47813xAYYIyPQ0UUAFIyK+N6hsdMjOKWigApvlptC7FwDkDHSnUUAc3deBtNvPGyeJpprk3KxRxtb7l8p/LJZGI27sgtnAYDIBIOBXSUUUAFFFFABXlc3w88aynxHp48SwDT9cvWlN5IzyT21u3WGKMjahwSu7PTsDzXqlFAHL3Xw/wBJm8Gad4atJLiwtNNeGW1ktim9HiO5W+dWUnPJyDkmr91BceHvCLW/hqwa9ubeIR2sDygb2JxudmPTJ3MevBxk8Vs0UAc74G8LN4S8MJY3Nz9sv5pXur67xjz7hzl3/PgewFdFRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAVGbaA3QuTDGZ1XYJdo3BfTPXFSUUAQPZWsl0lzJbQtcIMLK0YLL9D1FT0UUAUdY1iz0PTmvdRlMcQZUG1C7OzHCqqjkkk9BWBonivSdP8LWF54h8X2VwNQ3zW93eeXZ+ahbIUISPughfX1q349sNR1DwPqcOg28c+qmBhaBioKuRtLKW4DBWbBJHNZ+g6Br8PhPS7JLi10QWkZiS1+yLcFYhgRhm3434BLbcjLcdMkA0F+IXg1mCr4s0QsTgAajFz/49WZ4j1fXdM+Ifha1g1CD+ytUupYpbZbYbyFgZxmQk8bhngD61oLoXiEMCfEdvjPI/spP/AIqsTxpb+Irrxt4cvNK8NXF9aaLcSTyzJdQR+cHhZMIHcHIJ7gCgD0CimxMzxIzoY2ZQShIJU+nHFOoAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigD/2Q==)

Cost function(J) of Linear Regression is the **Root Mean Squared Error (RMSE)** between predicted y value (pred) and true y value (y).

[Gradient Descent](https://www.geeksforgeeks.org/gradient-descent-in-linear-regression/)**:**  
To update θ1 and θ2 values in order to reduce Cost function (minimizing RMSE value) and achieving the best fit line the model uses Gradient Descent. The idea is to start with random θ1 and θ2 values and then iteratively updating the values, reaching minimum cost.

**Code**

## **Prediction Analysis.ipynb :**

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import matplotlib as mpl

%matplotlib inline

mpl.style.use(‘ggplot’)

car=pd.read\_csv(‘quikr\_car.csv’)

car.head()

car.shape

car.info()

backup=car.copy()

car=car[car[‘year’].str.isnumeric()]

car[‘year’]=car[‘year’].astype(int)

car=car[car[‘Price’]!=’Ask For Price’]

car[‘Price’]=car[‘Price’].str.replace(‘,’,’’).astype(int)

car[‘kms\_driven’]=car[‘kms\_driven’].str.split().str.get(0).str.replace(‘,’,’’)

car=car[car[‘kms\_driven’].str.isnumeric()]

car[‘kms\_driven’]=car[‘kms\_driven’].astype(int)

car=car[~car[‘fuel\_type’].isna()]

car.shape

car[‘name’]=car[‘name’].str.split().str.slice(start=0,stop=3).str.join(‘ ‘)

car=car.reset\_index(drop=True)

car

car.to\_csv(‘Cleaned\_Car\_data.csv’)

car.info()

car.describe(include=’all’)

car=car[car[‘Price’]<6000000]

car[‘company’].unique()

import seaborn as sns

plt.subplots(figsize=(25,12))

ax=sns.boxplot(x=’company’,y=’Price’,data=car)

ax.set\_xticklabels(ax.get\_xticklabels(),rotation=40,ha=’right’)

plt.show()

plt.subplots(figsize=(20,10))

ax=sns.swarmplot(x=’year’,y=’Price’,data=car)

ax.set\_xticklabels(ax.get\_xticklabels(),rotation=40,ha=’right’)

plt.show()

sns.relplot(x=’kms\_driven’,y=’Price’,data=car,height=10,aspect=1.5)

plt.subplots(figsize=(20,10))

sns.boxplot(x=’fuel\_type’,y=’Price’,data=car)

ax=sns.relplot(x=’company’,y=’Price’,data=car,hue=’fuel\_type’,size=’year’,height=10,aspect=2)

ax.set\_xticklabels(rotation=40,ha=’right’)

X=car[[‘name’,’company’,’year’,’kms\_driven’,’fuel\_type’]]

y=car[‘Price’]

X

y

from sklearn.model\_selection import train\_test\_split

X\_train,X\_test,y\_train,y\_test=train\_test\_split(X,y,test\_size=0.2)

from sklearn.linear\_model import LinearRegression

from sklearn.preprocessing import OneHotEncoder

from sklearn.compose import make\_column\_transformer

from sklearn.pipeline import make\_pipeline

from sklearn.metrics import r2\_score

ohe=OneHotEncoder()

ohe.fit(X[[‘name’,’company’,’fuel\_type’]])

column\_trans=make\_column\_transformer((OneHotEncoder(categories=ohe.categories\_),[‘name’,’company’,’fuel\_type’]),remainder=’passthrough’)

lr=LinearRegression()

pipe=make\_pipeline(column\_trans,lr)

pipe.fit(X\_train,y\_train)

y\_pred=pipe.predict(X\_test)

r2\_score(y\_test,y\_pred)

scores=[]

for i in range(1000):

X\_train,X\_test,y\_train,y\_test=train\_test\_split(X,y,test\_size=0.1,random\_state=i)

lr=LinearRegression()

pipe=make\_pipeline(column\_trans,lr)

pipe.fit(X\_train,y\_train)

y\_pred=pipe.predict(X\_test)

scores.append(r2\_score(y\_test,y\_pred))

np.argmax(scores)

scores[np.argmax(scores)]

pipe.predict(pd.DataFrame(columns=X\_test.columns,data=np.array([‘Maruti Suzuki Swift’,’Maruti’,2019,100,’Petrol’]).reshape(1,5)))

X\_train,X\_test,y\_train,y\_test=train\_test\_split(X,y,test\_size=0.1,random\_state=np.argmax(scores))

lr=LinearRegression()

pipe=make\_pipeline(column\_trans,lr)

pipe.fit(X\_train,y\_train)

y\_pred=pipe.predict(X\_test)

r2\_score(y\_test,y\_pred)

pipe.predict(pd.DataFrame(columns=[‘name’,’company’,’year’,’kms\_driven’,’fuel\_type’],data=np.array([‘Maruti Suzuki Swift’,’Maruti’,2019,100,’Petrol’]).reshape(1,5)))

import pickle

pickle.dump(pipe,open(‘LinearRegressionModel.pkl’,’wb’))

pipe.steps[0][1].transformers[0][1].categories[0]

pipe.steps[0][1].transformers[0][1].categories[1]

## **app.py:**

from flask import Flask,render\_template,request,redirect

from flask\_cors import CORS,cross\_origin

import pickle

import pandas as pd

import numpy as np

app=Flask(\_\_name\_\_)

cors=CORS(app)

model=pickle.load(open('LinearRegressionModel.pkl','rb'))

car=pd.read\_csv('Cleaned\_Car\_data.csv')

@app.route('/',methods=['GET','POST'])

def index():

companies=sorted(car['company'].unique())

car\_models=sorted(car['name'].unique())

year=sorted(car['year'].unique(),reverse=True)

fuel\_type=car['fuel\_type'].unique()

transmission=car['transmission'].unique()

companies.insert(0,'Select Company')

return render\_template('index.html',companies=companies, car\_models=car\_models, years=year, fuel\_types=fuel\_type, transmission=transmission)

@app.route('/predict',methods=['POST'])

@cross\_origin()

def predict():

company=request.form.get('company')

car\_model=request.form.get('car\_models')

year=request.form.get('year')

fuel\_type=request.form.get('fuel\_type')

driven=request.form.get('kilo\_driven')

transmission=request.form.get('transmission')

prediction=model.predict(pd.DataFrame(columns=['name', 'company', 'year', 'kms\_driven', 'fuel\_type', 'transmission'],

data=np.array([car\_model, company, year, driven, fuel\_type, transmission]).reshape(1, 6)))

print(prediction)

return str(np.round(prediction[0],2))

df = pd.read\_csv('C:/Users/Amay\Desktop/MINI PROJECT/Prediction Model For Car Price/templates/Car\_data.csv')

df.to\_csv('C:/Users/Amay\Desktop/MINI PROJECT/Prediction Model For Car Price/templates/Car\_data.csv', index=None)

@app.route('/carrec')

def carrec():

# converting csv to html

data = pd.read\_csv('C:/Users/Amay\Desktop/MINI PROJECT/Prediction Model For Car Price/templates/Car\_data.csv')

data.to\_csv('C:/Users/Amay\Desktop/MINI PROJECT/Prediction Model For Car Price/templates/Car\_data.csv', index=None)

return render\_template('carrec.html', tables=[data.to\_html()], titles=[''])

if \_\_name\_\_=='\_\_main\_\_':

app.run()

**Output**

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Text

Description automatically generated with low confidence

### Checking relationship of Company with Price

Chart, box and whisker chart

Description automatically generated

### Checking relationship of Year with Price

Chart, scatter chart

Description automatically generated

### Checking relationship of kms\_driven with Price

Chart, scatter chart

Description automatically generated

### Checking relationship of Fuel Type with Price

Chart, box and whisker chart

Description automatically generated

### Relationship of Price with Fuel Type, Year and Company mixed

Chart, scatter chart

Description automatically generated

Graphical user interface

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

A picture containing background pattern

Description automatically generated

Background pattern

Description automatically generated with medium confidence

Graphical user interface, text, application, Word

Description automatically generated

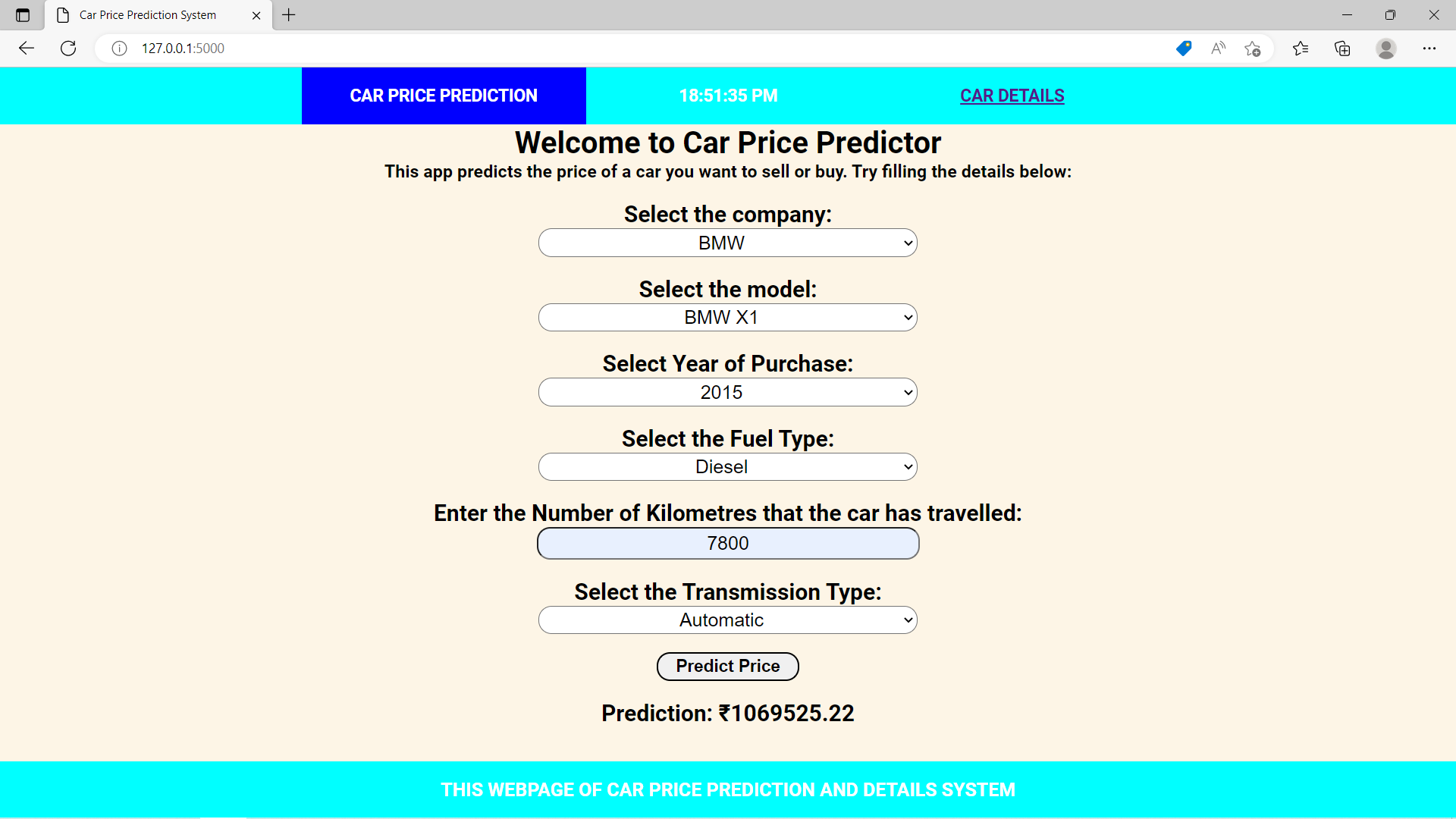
Graphical user interface, text, application

Description automatically generated

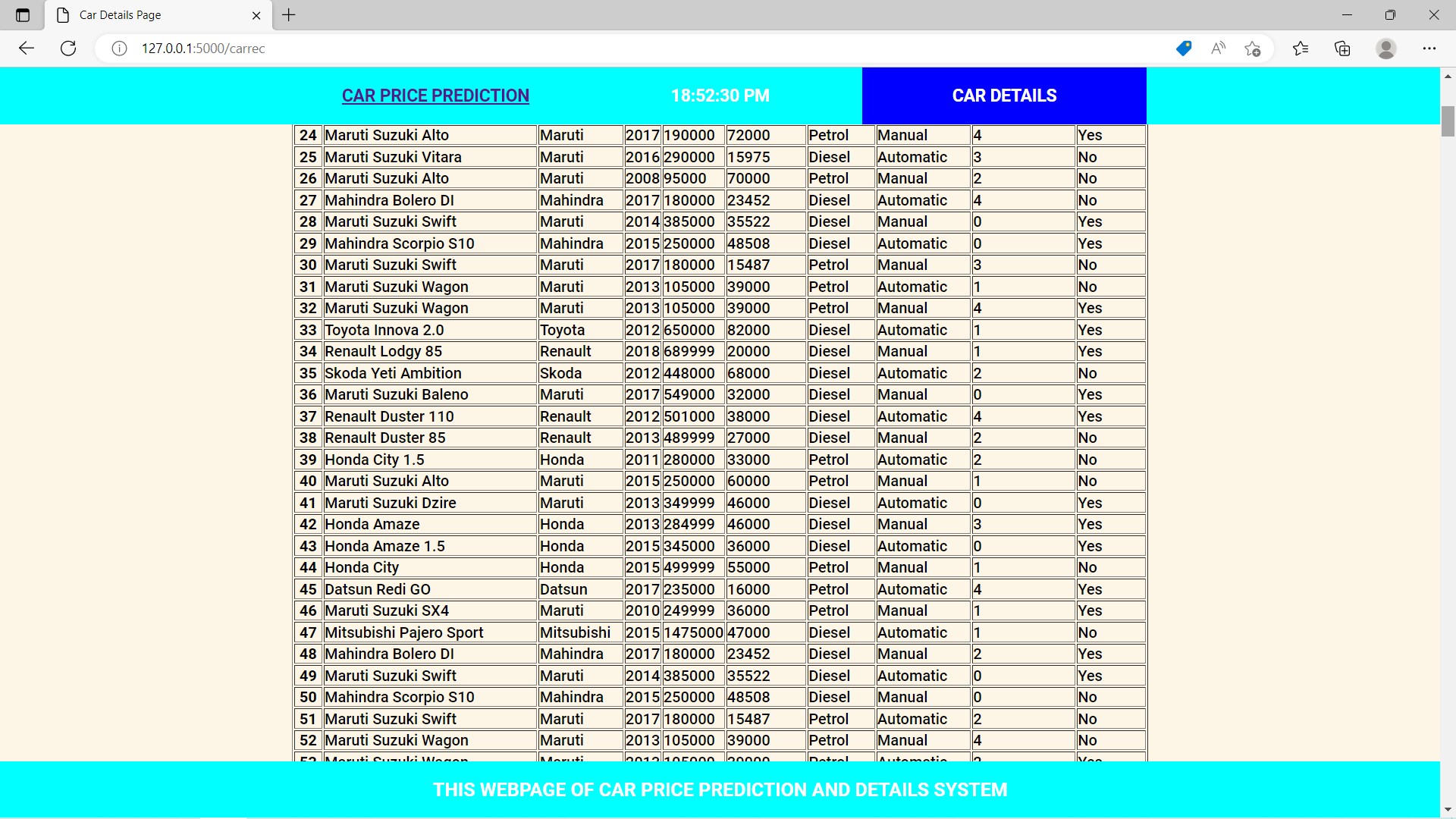
A screenshot of a computer

Description automatically generated

**Prediction Page:**



**Detail Page:**



**Conclusion**

Car price prediction can be a challenging task due to the high number of attributes that should be considered for the accurate prediction. The major step in the prediction process is collection and preprocessing of the data.

Data cleaning is one of the processes that increases prediction performance, yet insufficient for the cases of complex data sets as the one in this research.

Applying single machine algorithm on the data set accuracy was greater than 90%.

Although, this system has achieved astonishing performance in car price prediction problem our aim for the future research is to test this system to work successfully with data set.

By using this web application, we can predict the price of car and also display the details of car.

**Future Scope**

* In future this machine learning model may bind with various website which can provide real time data for price prediction.
* Also we may add large historical data of car price which can help to improve accuracy of the machine learning model.
* For better performance, we plan to judiciously design deep learning network structures, use adaptive learning rates and train on clusters of data rather than the whole dataset.
* Also we may add the recommendation page where we have add only car model name and it will display the result related to it.
* We can host this application on servers such as Heroku, etc.

**References**

* **Article**

<https://www.irjet.net/archives/V8/i4/IRJET-V8I4278.pdf>

<https://www.temjournal.com/content/81/TEMJournalFebruary2019_113_118.pdf>

<https://www.researchgate.net/publication/317608326_Vehicle_Price_Prediction_System_using_Machine_Learning_Techniques>

* **Website**

<https://www.analyticsvidhya.com/blog/2021/05/build-and-deploy-a-car-price-prediction-system/>

<https://lingcure.org/index.php/journal/article/download/1660/546>

* **Blog**

<https://towardsdatascience.com/used-car-price-prediction-using-machine-learning-e3be02d977b2>

<https://www.datascience2000.in/2021/05/car-price-prediction-in-machine-learning.html>

* **YouTube Links**

<https://www.youtube.com/watch?v=L3OtLaCbJC8>

<https://www.youtube.com/watch?v=6qxUcdKd43I>