Car Resale value Prediction Using Machine Learning Introduction: With difficult economic conditions, it is likely that sales of second-hand imported (reconditioned) cars and used cars will increase. In many developed countries, it is common to lease a car rather than buying it outright. After the lease period is over, the buyer has the possibility to buy the car at its residual value, i.e. its expected resale value. Thus, it is of commercial interest to sellers/financers to be able to predict the salvage value (residual value) of cars with accuracy. In order to predict the resale value of the car, we proposed an intelligent, flexible, and effective system that is based on using regression algorithms. Considering the main factors which would affect the resale value of a vehicle a regression model is to be built that would give the nearest resale value of the vehicle. We will be using various regression algorithms and algorithm with the best accuracy will be taken as a solution, then it will be integrated to the web-based application where the user is project objectives: • Project Flow • Pre-Requisit • Project Folder Structure • Importing Required Libraries • Spitting Data Into Independent & Dependent Variables • Model Building • Application Building Car Resale Value Prediction With difficult economic conditions, it is likely that sales of second-hand imported (reconditioned) cars and used cars will increase. In many developed countries, it is common to lease a car rather than buying it outright. After the lease period is over, the buyer has the possibility to buy the car at its residual value, i.e. its expected resale value. Thus, it is of commercial interest to sellers/financers to be able to predict the salvage value (residual value) of cars with accuracy. In order to predict the resale value of the car, we proposed an intelligent, flexible, and effective system that is based on using regression algorithms. Considering the main factors which would affect the resale value of a vehicle a regression model is to be built that would give the nearest resale value of the vehicle. We will be using various regression algorithms and algorithm with the best accuracy will be taken as a solution, then it will be integrated to the web-based application where the user is notified with the status of his product. Project Objectives : By the end of this project: . You'll be able to understand the problem to classify if it is a regression or a classification kind of problem. .You will be able to know how to pre-process/clean the data using different data pre- processing techniques. .Applying different algorithms according to the dataset .You will be able to know how to evaluate the model. .You will be able to build web applications using the Flask framework. Project Flow : Find below the project flow to be followed while developing the project. • User interacts with the UI (User Interface) to enter the input features .Entered input features are analyzed by the model which is integrated • Once the model analyses the input, the prediction is showcased on the UI • To accomplish this, we have to complete all the activities and tasks listed below • Download the dataset • Preprocess or clean

the data. • Analyze the pre-processed data. • Train the machine with preprocessed data using an appropriate machine learning algorithm. • Save the model and its dependencies. • Build a Web application using Flask that integrates with the model built. Pre-Requisite: • In order to develop this project we need to install the following software/packages: • Step 1: • Anaconda Navigator : • Anaconda Navigator is a free and open-source distribution of the Python and R programming languages for data science and machine learning related applications. It can be installed on Windows, Linux, and macOS.Conda is an open-source, cross-platform, package management system. Anaconda comes with great tools like JupyterLab, Jupyter Notebook, QtConsole, Spyder, Glueviz, Orange, Rstudio, Visual Studio Code. • For this project, we will be using Jupyter notebook and Spyder Step 2: To build Machine learning models you must require the following packages Sklearn: Scikit-learn is a library in Python that provides many unsupervised and supervised learning algorithms. NumPy: NumPy is a Python package that stands for 'Numerical Python'. It is the core library for scientific computing, which contains a powerful n-dimensional array object Pandas: pandas is a fast, powerful, flexible, and easy to use open-source data analysis and manipulation tool, built on top of the Python programming language. Matplotlib: It provides an object-oriented API for embedding plots into applications using general-purpose GUI toolkits Flask: Web framework used for building Web applications. If you are using anaconda navigator, follow the below steps to download the required packages: Open anaconda prompt. Type "pip install numpy" and click enter. Type "pip install pandas" and click enter. Type "pip install matplotlib" and click enter. Type "pip install scikit-learn" and click enter. Type "pip install Flask" and click enter. If you are using Pycharm IDE, you can install the packages through the command prompt and follow the same syntax as above. Project Folder Structure : • Let us introduce you to the main project folder downloaded by you in prerequisites. ."Resale value prediction final.py" has all the mastery model building architecture, that Collects Data, Import necessary packages, Pre-process images, and passes on to Network Model and Saves Model Weights into "resale_model.sav". ."Resale_flask.py" takes weights and Inputs from "User Interface" to Predict output. .The .npy format is the standard binary file format in NumPy for persisting a single arbitrary NumPy array on disk. The format stores all of the shape and dtype information are necessary to reconstruct the array correctly even on another machine with a different architecture. Collect Dataset : • The dataset which contains a set of features through which the resale price of the car can be identified is to be collected. You can collect datasets from different open sources like kaggle.com, data.gov, UCI machine learning repository, etc. • Please download the dataset in the data.txt file in the project folder provided to you in the prerequisites. Pre-Process The Data : • In this milestone, we will

be preprocessing the dataset that is collected. Preprocessing includes: 1. Handling the null values. 2. Handling the categorical values if any. 3. Normalize the data if required. 4. Identify the dependent and independent variables. 5. Split the dataset into train and test sets. Import Required Libraries : • The libraries can be imported using the import keyword. Read The Datasets : • The dataset is read as a data frame (df in our application) using the pandas library (pd is the alias name given to the pandas package). Cleaning The Dataset : • In this activity, the dataset is being cleaned. The dataset contains columns that have almost only one type of data. All the other entries expect the seller (gewerblich) containing only 3 entries are the same hence the column is dropped or removed from the dataset. Similarly, all the entries except the offer type (Gesuch) containing only 12 entries are same hence offer type column is also dropped. Suspicious data and data that is not in the range specified are not considered by only taking that is found to be valid. Like the Power should be between 50ps to 900ps and Year of Registration between 1950 and 2017. All the remaining entries are neglected. Columns that introduce bias or are the same for all the cars are removed. A copy of the dataset is made by deleting the rows that have the same values across all the columns. The first of such rows are stored and the remaining are omitted. The dataset contains a few German words. Hence replacing the German words with English words. Removing the outliers (the objects that deviate from the rest of the objects. They can be caused by measurement or execution error) and filling NaN values using the fillna() function. Saving the cleaned dataset for future use. Label encoding the categorical data. Splitting Data Into Independent And Dependent Variables : • In this activity, the dependent and independent variables are to be identified. The first column (Result) in the cleaned dataset is the dependent variable which is dependent on the remaining different factors. The independent columns are considered as x and the dependent column as y. In this activity, the dependent and independent variables are to be identified. The first column (Result) in the cleaned dataset is the dependent variable which is dependent on the remaining different factors. The independent columns are considered as x and the dependent column. Pre-Process The Data In this milestone, we will be preprocessing the dataset that is collected. Preprocessing includes: 1. Handling the null values. 2. Handling the categorical values if any. 3. Normalize the data if required. 4. Identify the dependent and independent variables. 5. Split the dataset into train and test sets. Model Building : There are several Machine learning algorithms to be used depending on the data you are going to process such as images, sound, text, and numerical values. The algorithms can be chosen according to the objective. As the dataset which we are using is a REgression dataset so you can use the following algorithms 1. Multi Linear Regression 2. Random Forest Regression / Classification 3. Decision Tree Regression /

Classification 4. K-Nearest Neighbors 5. Support Vector Machine You will need to train the datasets to run smoothly and see an incremental improvement in the prediction rate. Choose The Appropriate Model : • We will be initially considering the Random Forest Regressor model and fit the data. Check The Metrics Of The Model : • Here we will be evaluating the model built. We will be using the test set for evaluation. The test set is given to the model for prediction and prediction values are stored in another variable called y_pred. The r2 score of the model is calculated and its performance is estimated. Save The Model : • The finalized model is now to be saved. We will be saving the model as a sav file. Application Building : • After the model is built, we will be integrating it into a web application so that normal users can also use it to know the resale price of the care. In the application, the user provides the parameter values affecting the resale value. Application Building After the model is built, we will be integrating it into a web application so that normal users can also use it to know the resale price of the care. In the application, the user provides the parameter values affecting the resale value. Build The Python Flask App In the flask application, the user values are taken from the HTML page. These factors are then given to the model to know the resale value of a car. Step 1: Import required libraries Step 2: Load the model and initialize Flask app Step 3: Configure app.py to fetch the parameter values from the UI, and return the prediction. Step 4: Run the app Build An HTML Page We Build an HTML page to take the values from the user in a form and upon clicking on the button for submission it has to redirect to URL for "y_predict" which returns the predicted resale value. The output is to be then displayed on the page. The HTML pages are put under the templates folder and any style sheets if present is put in the static folder. Execute And Test Your Model Step 1: Execute the python code and after the module is running, open the localhost and click on the button to check the prediction.