**INDEX**

[CHAPTER 1 INTRODUCTION 5](#_Toc120138160)

[1.1 Purpose 5](#_Toc120138161)

[1.2 Problem Statement 5](#_Toc120138162)

[1.3 Overview 5](#_Toc120138163)

[1.4 Objective 6](#_Toc120138164)

[1.5 Tools and Technology 6](#_Toc120138165)

[CHAPTER 2 FEASIBLITY STUDY 7](#_Toc120138166)

[2.1 Research Paper-1 7](#_Toc120138167)

[2.2 Research Paper-2 7](#_Toc120138168)

[2.3 Machine Learning Model Structure 8](#_Toc120138169)

[2.4 Implementation Steps 8](#_Toc120138170)

[CHAPTER 3 SYSTEM REQUIREMENTS STUDY 9](#_Toc120138171)

[3.1 Hardware & Software Requirements 9](#_Toc120138172)

[CHAPTER 4 PROTOTYPE 10](#_Toc120138173)

[4.1 Design 10](#_Toc120138174)

[4.2 Designing Tools 10](#_Toc120138175)

[CHAPTER 5 DATABASE STRATEGY 11](#_Toc120138176)

[Database 11](#_Toc120138177)

[5.1 Rejected Databases 11](#_Toc120138178)

* [Car price Assignment (Kaggle) 11](#_Toc120138179)
* [New Used car Dataset (Kaggle) 13](#_Toc120138180)

[5.2 Final Databases 14](#_Toc120138181)

[CHAPTER 6 CAR PRICE PREDICTION MODEL 15](#_Toc120138182)

[6.1 Flowchart 15](#_Toc120138183)

[6.2 Datasets 15](#_Toc120138184)

[6.3 Preprocessing Steps 16](#_Toc120138185)

* [Heatmap 16](#_Toc120138186)
* [Relational Heatmap 17](#_Toc120138187)

[6.4 Methodology 18](#_Toc120138188)

[6.5 Result 21](#_Toc120138189)

CHAPTER 7 LAPTOP PRICE PREDICTION……………………………………..22

[CHAPTER 8 CONCLUSION 21](#_Toc120138190)

* 1. [Conclusion 22](#_Toc120138191)

[Bibliography 22](#_Toc120138192)

[References 22](#_Toc120138193)

# LIST OF FIGURES

[Figure 1 : Implementation Steps 8](#_Toc120132776)

[Figure 2 : Block Diagram for all Classification Process 8](#_Toc120132777)

[Figure 3 : Frontend UI 10](#_Toc120132778)

[Figure 4 : Flowchart 15](#_Toc120132779)

[Figure 5 : Heatmap 16](#_Toc120132780)

[Figure 6 : Relational Heatmap 17](#_Toc120132781)

[Figure 7 : Linear Regression 18](#_Toc120132782)

[Figure 8 : Random Forest 20](#_Toc120132783)

# 

# LIST OF TABLE

Table 1: Hardware and Software Requirement

Table 2: Accuracy Table

# 

# CHAPTER 1

# INTRODUCTION

**Car Price Prediction Model: -**

The used car market is an ever-rising industry, which has almost doubled its market value in the last few years. The emergence of online portals such as Car Dekho, Quikr, Carwale, Cars24, and many others has facilitated the need for both the customer and the seller to be better informed about the trends and patterns that determine the value of the used car in the market. Machine Learning algorithms can be used to predict the retail value of a car, based on a certain set of features. Different websites have different algorithms to generate the retail price of the used cars, and hence there isn't a unified algorithm for determining the price. By training statistical models for redacting the prices, one can easily get a rough estimate of the price without actually entering the details into the desired website. The main objective of this paper is to use three different prediction models to predict the retail price of a used car and compare their levels of accuracy. The data set used for the prediction models was created in 2020 by Car Dekho uploaded now on Kaggle, whose retail prices have been calculated. The dataset primarily comprises of categorical attributes along with two quantitative attributes.

## **Purpose : -**

The Main purpose of creating this project is to create a smart way to identify the prices and values of commodities online by some practical inputs. By using this application on can predict their cars and bike prices online on the basis of machine learning model and also there is no need of any offline broker or agent to identify their price or values.

## **Problem Statement : -**

The prices of new cars in the industry are 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 price of new cars and the incapability of customers to buy new cars due to the lack of funds, used cars sales are on a global increase. There is a need for a used car price prediction system to effectively determine the worthiness of the car using a variety of features. Even though there are websites that offers this service, their prediction method may not be the best. Besides, different models and systems may contribute on predicting power for a used car’s actual market value. It is important to know their actual market value while both buying and selling.

## **Overview :-**

**Required of prediction system:** To effectively determine the worthiness of things by giving accurate result by prediction for better decision.

**Target Business:** Online Used-car dealers.

**Target Customers:** Buyers who would like to purchase used cars via online portal.

## **Objective:-**

The main objective of the project is to create an online predictor which can predict different prices, size, occurrences, probability that allows users to find out best prices or accuracy of the things on which they are searching for.

## **Tools and Technology:**

* Python.
* Google Colab.
* Machine Learning.
* Flask.
* Visual Studio Code.
* Github.

# 

# CHAPTER 2

# FEASIBLITY STUDY

## **2.1** **Research Paper-1**

The first paper is Predicting the worth of Used automobile exploitation Machine Learning techniques. In this paper, they investigate the application of supervised machine learning techniques to predict the worth of used cars in Mauritius. The predictions are supported historical knowledge collected from daily newspapers. Different techniques like multiple regression toward the mean analysis-nearest neighbours, native Bayes and call trees have been accustomed create the predictions.

Automobile value Prediction victimization Machine Learning Techniques considerable range of distinct attributes square measure examined for the reliable and correct prediction. to create a model for predicting the worth of used cars in Bosnia-Herzegovina and Herzegovina,[1] they need applied machine learning techniques (Artificial Neural Network, Support Vector Machine and Random Forest).

The worth analysis model in used car system supported neural networks. during this paper, the price analysis model supported massive information analysis is proposed, that takes advantage of wide circulated vehicle data and an oversized variety of car group action information to analyse the value information for every sort of vehicles by victimization the optimized neural network rule. It aims to ascertain a second-hand automobile worth analysis model to induce the value that best matches the automobile.

A wide variety of approaches, algorithms, statistical software, and validation strategies were employed in the application of machine learning methods to inform provider for decision making. There is a need to ensure that multiple machine learning approaches are used, the model selection strategy is clearly defined, and both internal and external validation are necessary to be sure that decisions for predictions are being made with the highest quality evidence.

## **2.2 Research Paper-2**

Overfitting and underfitting come into picture when we create our statistical models. The models might be too biased to the training data and might not perform well on the test data set. This is called overfitting. Likewise, the models might not take into consideration all the variance present in the population and perform poorly on a test data set. This is called underfitting. A perfect balance needs to be achieved between these two, which leads to the concept of Bias-Variance tradeoff.

Pierre Geurts [2] has introduced and explained how bias-variance tradeoff is achieved in both regression and classification. The selection of variables/attribute plays a vital role in influencing both the bias and variance of the statistical model. Robert Tibshirani

[3] proposed a new method called Lasso, which minimizes the residual sum of squares. This returns a subset of attributes which need to be included in multiple regression to get the minimal error rate. Similarly, decision trees suffer from overfitting if they are not pruned/shrunk. Trevor Hastie and Daryl Pregibon. [4] have explained the concept of pruning in their research paper. Moreover, hypothesis testing using ANOVA is needed to verify whether the different groups of errors really differ from each other. This is explained by TK Kim and Tae Kyun in their paper [5]. A Post-Hoc test needs to be performed along with ANOVA if the number of groups exceeds two. Tukey’s Test has been explored by Haynes W. in his research paper.

## **2.3 Machine Learning Model Structure**

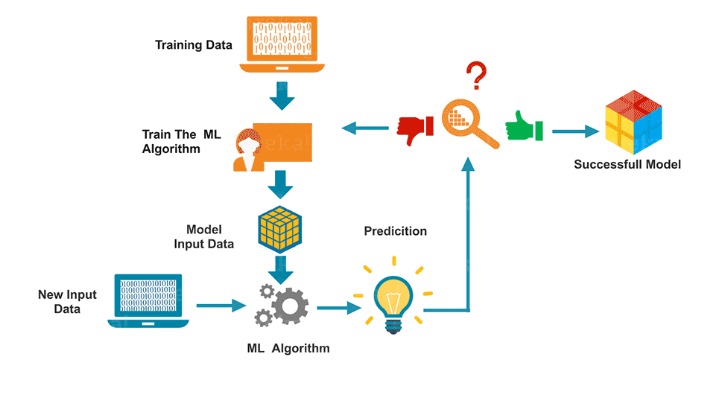


Figure 1 Machine Learning Structure

## **2.4 Implementation Steps**

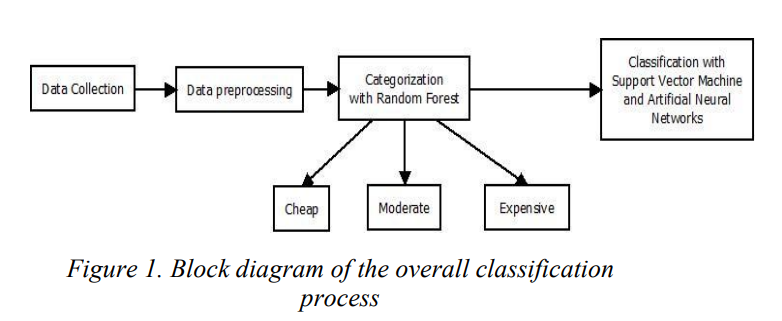
****

Figure 2 Block Diagram for all Classification Process

# 

# CHAPTER 3

# SYSTEM REQUIREMENTS STUDY

## **3.1 Hardware & Software Requirements**

|  |  |  |
| --- | --- | --- |
|  | **Hardware** | **Software** |
| **Developers** | 1. 8 GB RAM 2. 256 GB Storage 3. Intel i5 10th Gen + Processor | 1. Collab or Python 2. Pycharm IDE 3. Vscode |
| **Users** | 1. Windows PC, Mobiles, Tablets. 2. Min 4 GB RAM | 1. Chrome, Edge, Firefox. |

Table 1 : Hardware and Software Requirement

# 

# CHAPTER 4

# PROTOTYPE

## **4.1 Design**

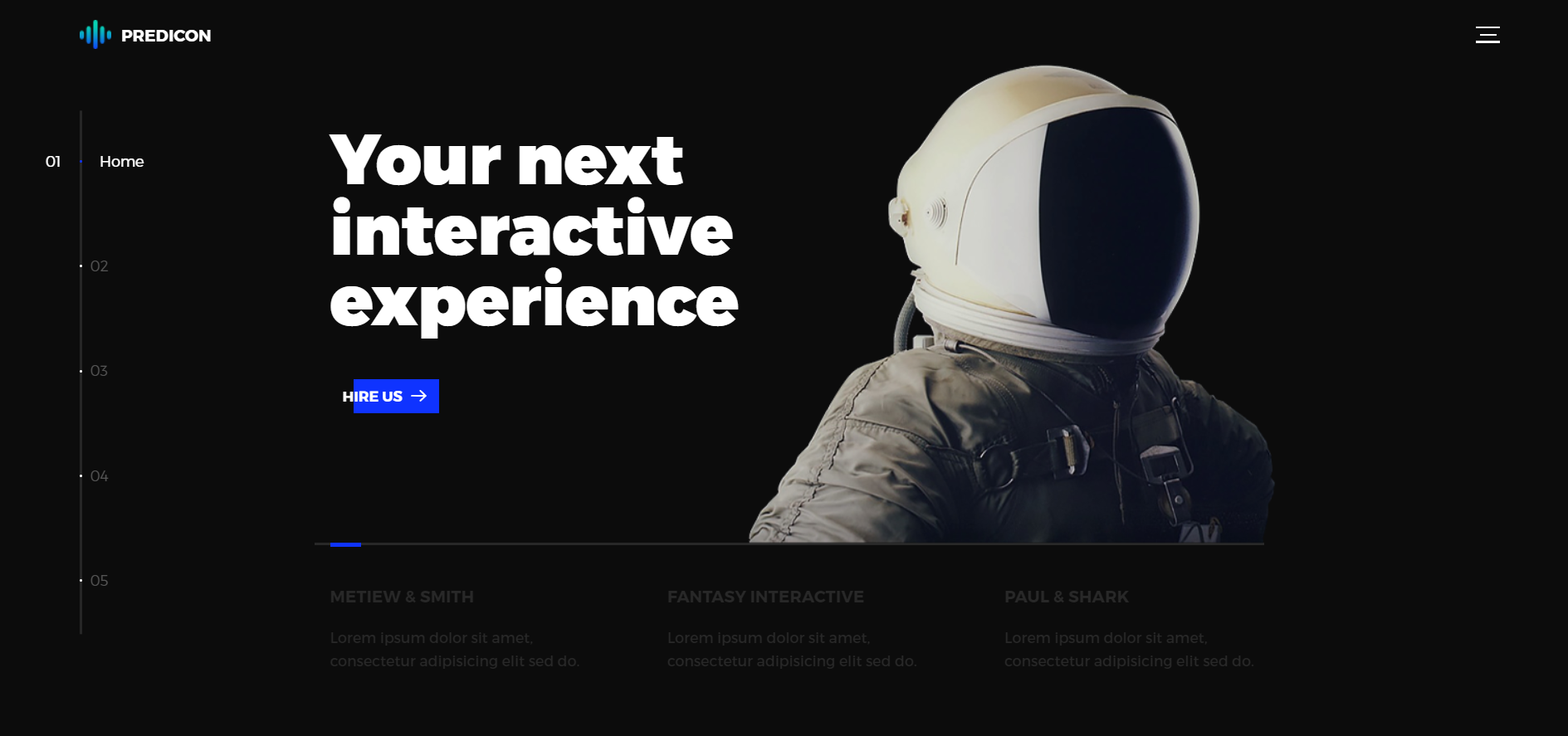


Figure 3 Frontend UI

## **4.2 Designing Tools**

|  |
| --- |
| **HTML5** |
| **CSS 3.0** |
| **BOOTSTRAP** |
| **Javascript** |
| **Flask** |

# 

# CHAPTER 5

# DATABASE STRATEGY

## **Database:-**

## **5.1 Rejected Databases**

### **A) Car price Assignment (Kaggle)**

**Reason’s Why Rejected**

* We rejected this dataset because it doesn’t have the cars showroom price. So without the comparison between cars road price and showroom price we cannot predict the price of cars by just cars showroom price.
* In this Dataset we don’t get accuracy as per mark.
* This dataset contain so 26 columns which includes various unwanted columns and by removing those columns affects the accuracy and coorelation for heatmap

**Rows And Columns**

|  |  |
| --- | --- |
| No. of Rows | 205 |
| No. of Cols. | 26 |

**A) Car price assignment features**

|  |
| --- |
| car\_ID |
| symboling |
| CarName |
| Fueltype |
| aspiration |
| doornumber |
| carbody |
| drivewheel |
| enginelocation |
| wheelbase |
| carlength |
| curbweight |
| enginetype |
| cylindernumber |
| enginesize |
| boreratio |
| stroke |
| compression |
| horsepower |
| highwaympg |
| peakrpm |
| Price |
| citympg |

### **B. New Used car Dataset (Kaggle) :-**



**Reason’s Why Rejected**

* This Dataset contain foreign countries cars descriptions and models that’s why that dataset dosen’t fit in our model.
* This Dataset doesn’t contain our requirements to predict the car prices and dosen’t give Correlation for Heatmap.

**New Used car Dataset Columns Name**

|  |
| --- |
| Car Name |
| Mileages |
| Ratings |
| Reviews |
| used/certified |
| price drop |

**Rows And Columns :**

|  |  |
| --- | --- |
| No. of Rows | 9330 |
| No. of Cols. | 6 |

******5.1 Final Databases**

**Reason Why Accepted**

* This Dataset mainly contains cars descriptions and models which perfectly fits into our model.
* Pricing of cars like Showroom price and On-Road price are given perfectly in each rows so that we can correlate.

# CHAPTER 6

# CAR PRICE PREDICTION MODEL

## **6.1 Flowchart**

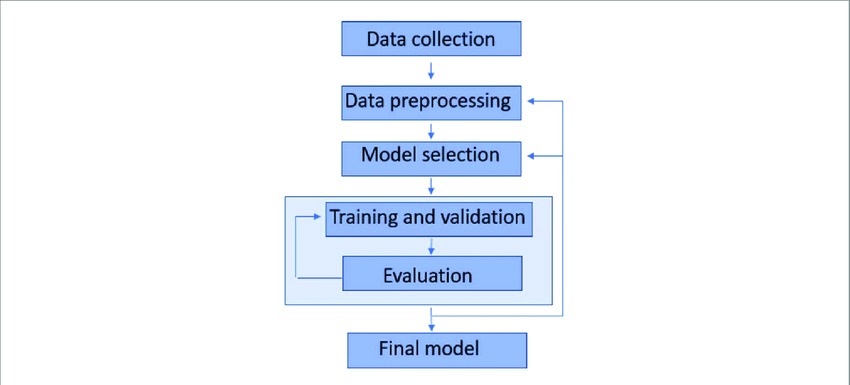


Figure 4 Flowchart

## **6.2 Datasets**



**Proposed dataset information**

|  |
| --- |
| full\_name |
| selling\_price |
| new\_price |
| Year |
| seller\_type |
| km\_driven |
| owner\_type |
| fuel\_type |
| transmission\_type |
| Mileage engine |
| Mileage engine |
| seatstransmission\_type |

|  |  |
| --- | --- |
| No of Rows. | 19974 |
| No of Cols. | 12 |

## **6.3 Preprocessing Steps**

### **A. Heatmap**

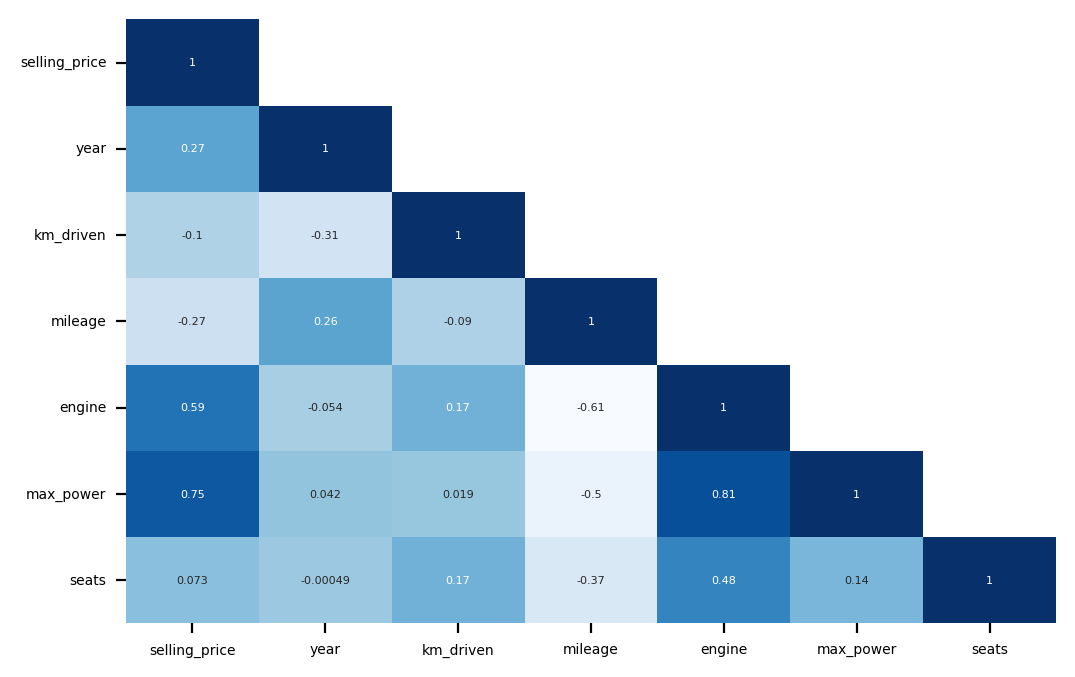
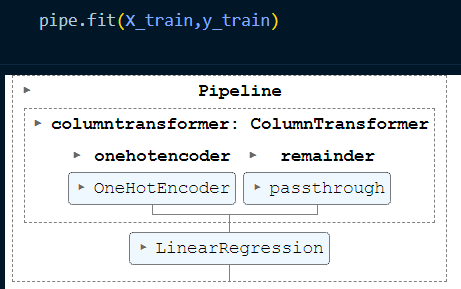


Figure 5 Heatmap

### **B. Relational Heatmap**

Figure 6 Relational Heatmap

****

## **6.4 Methodology**

We utilized several classic and state-of-the-art methods, including ensemble learning techniques, with a 90% - 10% split for the training and test data. To reduce the time required for training, we used 500 thousand examples from our dataset. Linear Regression, Random Forest and Gradient Boost were our baseline methods. For most of the model implementations, the open-source Scikit-Learn package was used.

**Linear Regression**

Linear Regression was chosen as the first model due to its simplicity and comparatively small training time. The features, without any feature mapping, were used directly as the feature vectors. No regularization was used since the results clearly showed low variance.

**Explanation**

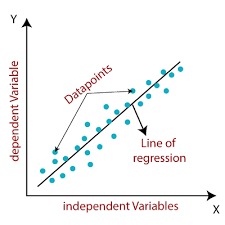


Figure 7 Linear Regression

**Equation**

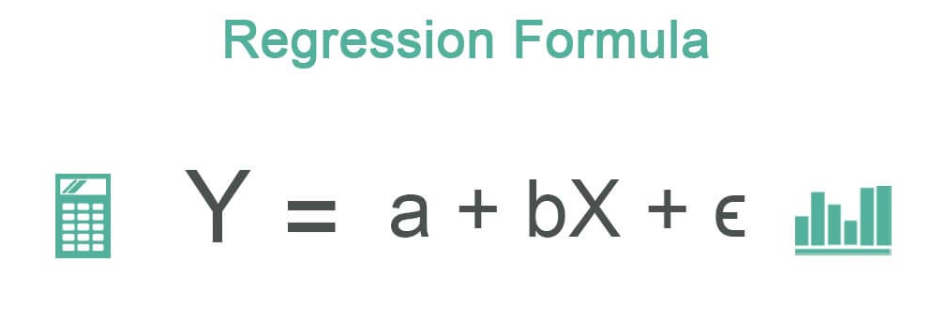


Figure 8 Linear Regression Formula

**Random Forest**

Forest is an ensemble learning based regression model. It uses a model called decision tree, specifically as the name suggests, multiple decision trees to generate the ensemble model which collectively produces a prediction. The benefit of this model is that the trees are produced in parallel and are relatively uncorrelated, thus producing good results as each tree is not prone to individual errors of other trees. This uncorrelated behavior is partly ensured by the use of Bootstrap Aggregation or bagging providing the randomness required to produce robust and uncorrelated trees. This model was hence chosen to account for the large number of features in the dataset and compare a bagging technique with the following gradient boosting methods.

**Explanation**

When using the Random Forest Algorithm to solve regression problems, you are using the mean squared error (MSE) to how your data branches from each node.

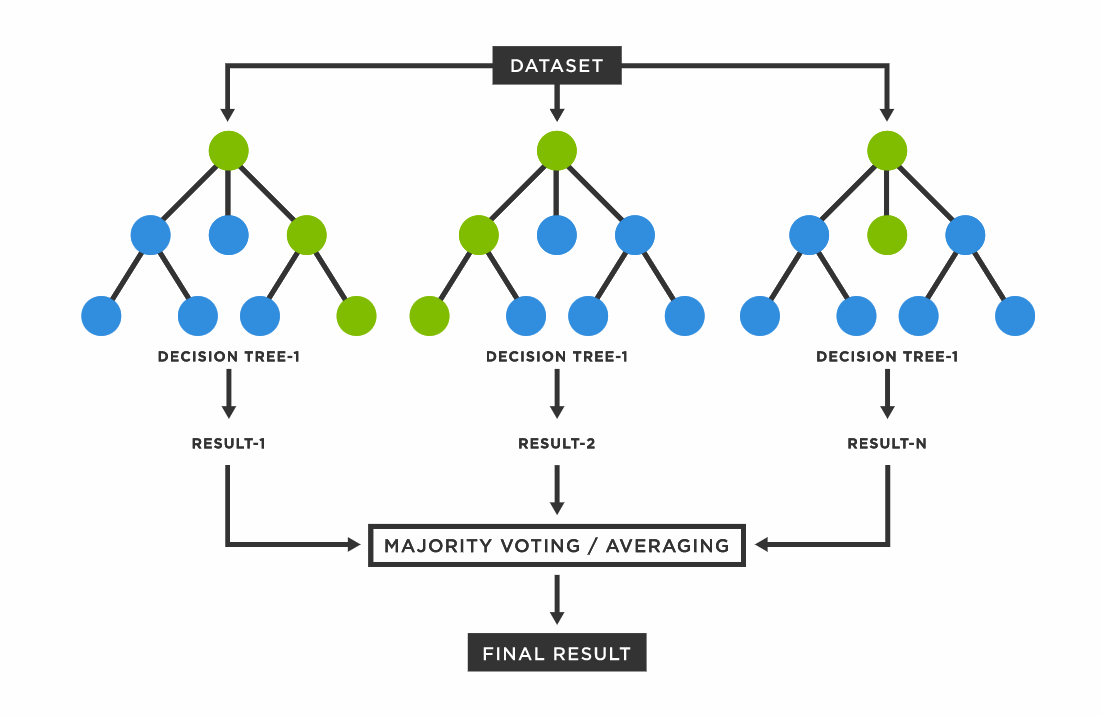


Figure 9 Random Forest

**Equation**

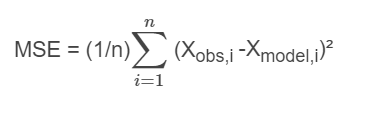
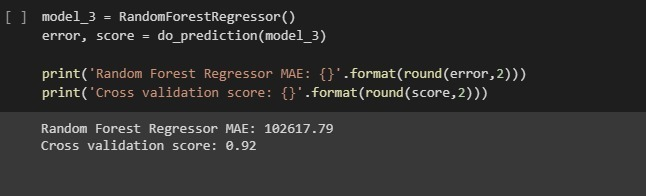


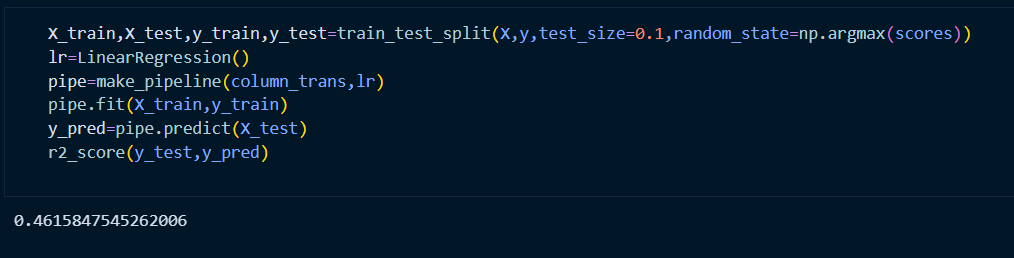
Figure 10 Mean Squared Error Equation

## **6.5 Result**

**A. Accuracy of Random Forest.**



**B. Accuracy of Linear Regression.**

****

|  |  |
| --- | --- |
| **Accuracy of Linear Regression.** | **0.93** |
| **Accuracy of Random Forest.** | **0.92** |

Table 2: Accuracy Table

**Q.Why the Accuracy of Random Forest is Greater than Linear Regression?**

Linear Models have very few parameters, Random Forests a lot more. That means that Random Forests will overfit more easily than a Linear Regression.

# CHAPTER 7

# LAPTOP PRICE PREDICTION MODEL

**5.Introduction**

Laptop price prediction especially when the laptop is coming direct from the factory to Electronic Market/ Stores, is both a critical and important task. The mad rush that we saw in 2020 for laptops to support remote work and learning is no longer there. In India, demand of Laptops soared after the Nationwide lockdown, leading to 4.1-Million-unit shipments in the June quarter of 2021, the highest in the five years. Accurate Laptop price prediction involves expert knowledge, because price usually depends on many distinctive features and factors. Typically, most significant ones are brand and model, RAM, ROM, GPU, CPU, etc. In this paper, we applied different methods and techniques in order to achieve higher precision of the used laptop price prediction.

The expansion of e-commerce businesses and the increased volume of products on e-commerce websites make product classification an intricate task. Product classification deals with the correct placement of the products in the relevant categories. It is a key feature for ecommerce websites that also facilitates marketing. Product categorization tends to increase conversion rates and return higher sales to the company. One of the major benefits of product classification is the improvement of website usability and navigation experience to the user. Users can look for the desired product quickly and easily. A high level of website usability influences user behavior in a positive way [1]. Users visit multiple websites before making the final purchase decision and are more likely to purchase from the website offering an optimal price and a better shopping experience [2].

For the automation of classification tasks, Machine Learning (ML) offers promising methods and algorithms. Classification in machine learning is a learning problem where a system learns to predict class labels on a set of data points. As a supervised learning problem, target class labels are also provided as an input to the classification algorithm. Classification can either be binary or multiclass. In binary classification, there are two classes to be predicted while multiclass classification problems involve predicting more than two classes. Some of the machine learning algorithms available for classification include Support Vector Machines (SVM), Decision Tree, Naïve Bayes, K-Nearest Neighbor, Multi-Layer Perceptron (MLP) [3]. The classification algorithms perform differently on different datasets. The performance of the classifier depends on the application, choice of features as well as nature of the dataset. This research work focuses on the multiclass classification task for a laptop products dataset. Here, the term ‘Laptop Products Classification’ refers to the categorization of laptop products in three classes namely, ‘Budget’, ‘Mid-range’ and ‘Flagship’ products. The inputs to the classifier are features like laptop company, product type, size, weight, RAM, and price. The classifier predicts whether the given product belongs to ‘Budget’, ‘Midrange’ or, the ‘Flagship’ class. Various machine learning classifiers including Support Vector Machines (SVM), Multinomial Logistic Regression, Decision Tree, and Artificial Neural Network (ANN) are used to predict the class of laptop products. The performance of classifiers is compared in terms of classification accuracy, recall, precision, and F1- score. The significance of this work is the better management of laptop products on an e-commerce website. From the user's point of view, it helps customers to find the required laptop product easily, efficiently, and according to their financial budget. It eliminates the need to scroll through hundreds of products to find the required one. For the business, it is beneficial because the smooth navigation and shopping experience are likely to bring the customer again to the website in the future, increasing the business sales. It also helps the business to manage the products more effectively resulting in increased productivity. The same research idea is also applicable to the categorization of smartphone products, tablets, and smartwatches, etc. Section II of the paper covers the literature review and discusses some of the related work. Section III discusses the research methodology and experiment details. Section IV presents data visualization. Section V reveals the results of classification and discusses the results. Section VI presents the conclusion of the paper and provides directions for future work.

**6.Project Scope:**

The project scope for laptop price prediction can be defined as follows:

Data collection: Collecting data related to laptops, such as brand, model, specifications, price, and customer reviews.

Data preprocessing: Cleaning and processing the collected data, removing any missing or irrelevant data, and preparing it for analysis.

Feature engineering: Selecting the most relevant features that can be used to predict the price of a laptop. These may include the brand, model, screen size, processor speed, RAM, storage capacity, graphics card, and customer ratings.

Model development: Developing a machine learning model that can predict the price of a laptop based on the selected features. This may involve exploring various algorithms, such as linear regression, decision trees, random forests, or neural networks.

Model evaluation: Evaluating the accuracy of the developed model and fine-tuning it to improve its performance.

Deployment: Deploying the model in a web application or mobile app, allowing users to predict the price of a laptop based on its specifications.

Maintenance: Maintaining the model and updating it regularly to ensure that it remains accurate and relevant over time.

**7.Literature Survey :**

Machine Learning (ML) is a field that is based on concepts and principles from multiple disciplines including Mathematics, Computer Science, Statistics, Cognitive Science, and Optimization Theory [4]. ML tasks are categorized into supervised learning, unsupervised learning, and reinforcement learning. From the supervised learning category, classification and regression are well-known tasks. In classification, the output is discrete e.g., class labels while output in regression takes on continuous values [3]. Researchers in [5] implemented the Multinomial Naïve Bayes algorithm for catalog classification. The products are categorized into classes like ‘Electronics’ and then subclasses like ‘printer’. The number of products is about 40,000 collected from different databases including Amazon, Flipkart, etc. An overall number of 1000 classes for 40,000 products in the system. Multinomial Naïve Bayes is mostly applied for document classification. The foundation of the Naïve Bayes Classifier is the Bayes theorem based on probability. Naïve Bayes assumption about features is that the features are independent. With X as a feature vector of size n, and y as the class label variable, Naïve Bayes predicts the class label as,

The work of [6] is based on the use of the Naïve Bayes classifier and the Decision Tree classifier to predict the classes of mobile phones with given features as ‘Economical’ or ‘Expensive’. The researchers collected the dataset from GSMArena.com. The features collected in the dataset include display size, weight, thickness, internal memory, camera, video quality, RAM, and battery. Two feature selection algorithms InfoGainEval and WrapperattributEval were applied to select the features that are most important in predicting the output class. The results are compared across the classifiers in terms of accuracy achieved with the selection of minimum features. The Decision Tree algorithm is a popular supervised learning algorithm that works well with classification and regression tasks. The algorithm models a tree-like flowchart that has a root node, decision nodes, branches, and leaf nodes. The algorithm divides the data into small parts to identify the patterns that can be used for making a prediction. The learning strategy behind decision trees is the divide and conquer strategy. The entire dataset is at the root node. The algorithm chooses the feature that best predicts the target class. The entries are divided into groups of feature values. This decision creates the first set of branches. The divide and conquer process continues on the nodes until a stop criterion is reached [7]. The popular decision tree algorithms include ID3, C4.5, and CART algorithm [8]. [9] worked on product categorization on a dataset collected from Amazon distributers, using machine learning classifiers including Naïve Bayes, K-Nearest Neighbor, and Tree Classifier. Features for each item were determined using the bag-of-words model. The features set was processed using the standard pre-processing techniques like stop word removal, punctuation and number removal, lowercasing, and lemmatization. After feature processing, feature importance was determined using a modified MI formula and finally, the features were selected using the forward and backward search strategies. Naïve Bayes finished with 76.9% accuracy, KNN resulted in 69.4% accuracy, and the tree classifier performed the best with 86% accuracy but the execution took a long time (8 hours) to complete as compared to Naïve Bayes (3 seconds) and KNN (4 minutes)

**8.Feasibility Analysis:**

Technical feasibility: Laptop price prediction is technically feasible as it involves using machine learning algorithms to analyze data related to various laptop features and predict their prices. There are many tools and platforms available that can be used to develop and deploy machine learning models for this purpose.

Time schedule feasibility: The time required for developing a laptop price prediction model depends on the complexity of the project, the amount of data available, and the expertise of the developers. However, with the availability of pre-built libraries and frameworks for machine learning, the development time can be reduced significantly.

Operational feasibility: Laptop price prediction can be operationally feasible if the necessary data is available and can be collected efficiently. It also requires the availability of appropriate computing resources for training and deploying the machine learning models.

Implementation feasibility: The implementation of a laptop price prediction system requires the integration of various components, including data collection, preprocessing, feature engineering, machine learning algorithms, and deployment. However, with the availability of cloud-based services and pre-built tools, the implementation can be done with relative ease.

Economic feasibility: The economic feasibility of a laptop price prediction system depends on the cost of data collection, data preprocessing, machine learning algorithms, and computing resources required for training and deployment. However, the potential benefits of such a system, such as improved price competitiveness and customer satisfaction, can justify the investment. Moreover, as the costs of computing resources and machine learning tools are decreasing over time, the economic feasibility is likely to improve in the future.

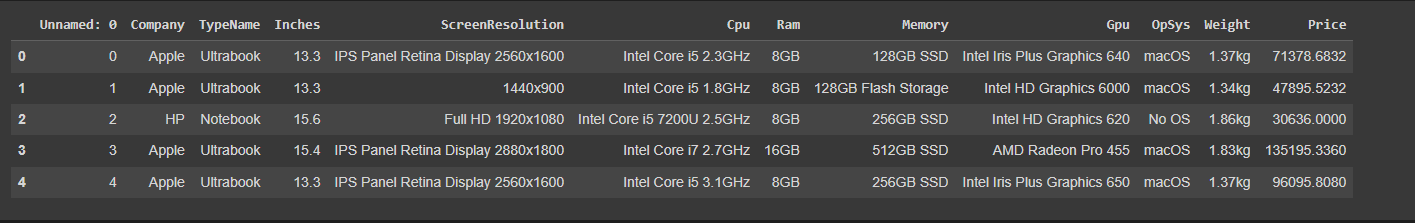
**9**.**Software and Hardware requirement:**

|  |  |  |
| --- | --- | --- |
|  | **Hardware** | **Software** |
| **Developers** | **1. 8 GB RAM**  **2. 256 GB Storage**  **3. Intel i5 10 Gen + Processor** | **1. Collab or Python**  **2. Pycharm IDE**  **3. Vscode** |
| **Users** | **1. Windows PC, Mobiles, Tablets.**  **2. Min 4 GB RAM** | **1. Chrome, Edge, Firefox.** |

**10.Process Model:**

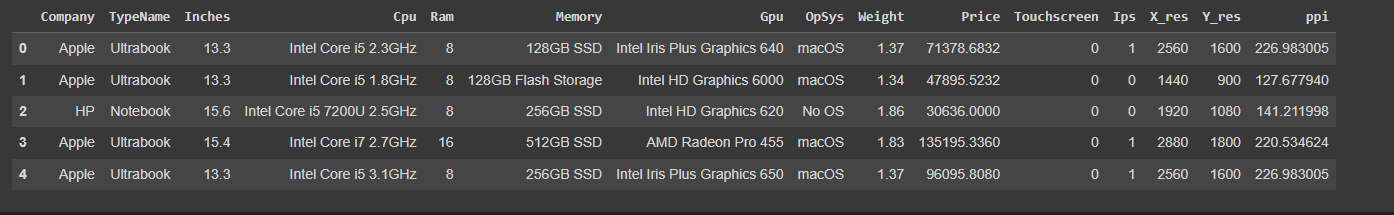
**10.1Data Collection:**

**This stage involves collecting data related to laptops, including brand, model, specifications, and prices, from various sources such as online retailers and manufacturers.**

****

**figure1:Dataset Before preprocessed**

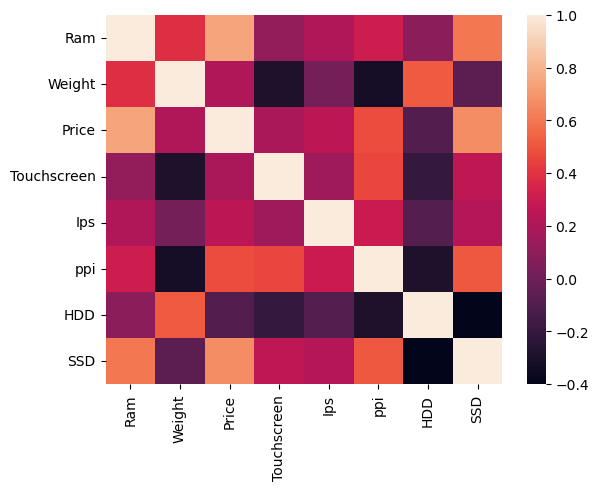
**10.2 Data Preprocessing: In this stage, the collected data is cleaned, normalized, and transformed into a format suitable for analysis. This involves handling missing or inconsistent data, removing duplicates, and performing feature scaling.**

****

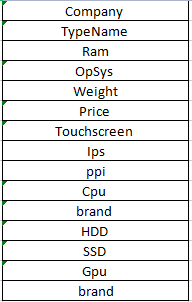
**Figure 2:Dataset after preprocessed**

**10.3 Feature Selection: This stage involves selecting the most relevant features that can help predict laptop prices. Feature selection techniques such as correlation analysis and principal component analysis (PCA) may be used to identify the most important features.**

**Feature selection Based on Heatmap:**

****

**figure:Heatmap**

****

**Figure 3:Feature selection on this dataset**

**10.4 Model Development: In this stage, a machine learning model is developed using the selected features and the preprocessed data. Various machine learning algorithms such as linear regression, decision trees, and random forests may be used to develop the model.**

**Using Linear regression:**

**Linear regression is a machine learning algorithm that is commonly used for predicting continuous numerical values, such as the price of a laptop. In the context of laptop price prediction, linear regression works by modeling the relationship between the input features of a laptop (such as brand, model, specifications, etc.) and its price.**

**The basic idea behind linear regression is to find a linear equation that best fits the relationship between the input features and the output (price) variable. The equation takes the form of:**

**The goal of linear regression is to find the values of the coefficients b1 to bn that minimize the difference between the predicted prices and the actual prices of the laptops in the training data. This is done using an optimization algorithm such as gradient descent.**

**Once the model is trained, it can be used to predict the price of a new laptop by simply plugging in its input features into the equation above. The predicted price will be the value of y calculated by the model.**

**It's worth noting that linear regression assumes that the relationship between the input features and the output variable is linear. If there are non-linear relationships or interactions between the input features, other machine learning algorithms such as decision trees or neural networks may be more suitable for predicting laptop prices.**

 Using SVM:

Support Vector Machines (SVM) is a popular machine learning algorithm used for classification and regression tasks, including laptop price prediction. In the context of laptop price prediction, SVM works by finding the best hyperplane that separates the laptops with different prices based on their input features.

The SVM algorithm tries to maximize the margin between the hyperplane and the closest points from each class, which are called support vectors. The support vectors are the training data points that are closest to the hyperplane and are used to define it. The hyperplane is used to predict the price of new laptops based on their input features.

In SVM, the input features of the laptop are transformed into a high-dimensional feature space, where the hyperplane is found. The algorithm uses a kernel function to map the input features to the higher-dimensional space. The kernel function can be linear, polynomial, or radial basis function (RBF), among others.

The SVM algorithm then optimizes the position of the hyperplane by minimizing the classification error or maximizing the margin between the support vectors. The optimization is typically done using a technique called gradient descent or a quadratic programming solver.

Once the model is trained, it can be used to predict the price of new laptops by transforming their input features into the high-dimensional feature space and evaluating their position relative to the hyperplane. The predicted price will depend on which side of the hyperplane the laptop falls on.

SVM is a powerful algorithm for laptop price prediction because it can handle non-linear relationships between the input features and the price. It also has good generalization properties, meaning it can make accurate predictions on new, unseen data. However, SVM can be sensitive to the choice of kernel function and hyperparameters, which may require careful tuning to achieve optimal performance.

Using RandomForest:

Random Forest is a popular machine learning algorithm used for regression and classification tasks, including laptop price prediction. In the context of laptop price prediction, Random Forest works by constructing an ensemble of decision trees and combining their predictions to make a final prediction.

The Random Forest algorithm works as follows:

1. A random subset of the input features and a random subset of the training data are used to build each decision tree in the ensemble. This is done to ensure that each tree in the ensemble is different and has a unique perspective on the data.
2. Each decision tree is trained on the selected subset of features and data using a process called recursive partitioning. The algorithm recursively splits the data into smaller subsets based on the values of the input features until each subset contains only laptops with similar prices.
3. During training, the algorithm evaluates multiple possible split points for each feature and selects the one that maximizes the reduction in the variance of the price predictions. This ensures that the splits are chosen to separate the laptops with different prices as effectively as possible.
4. Once all the decision trees in the ensemble have been trained, the algorithm combines their predictions to make a final prediction for a new laptop. This is done by taking the average of the individual tree predictions for regression tasks or using a voting mechanism for classification tasks.

Random Forest is a powerful algorithm for laptop price prediction because it can handle non-linear relationships between the input features and the price, and it can handle interactions between multiple input features. It also has good generalization properties, meaning it can make accurate predictions on new, unseen data.

However, Random Forest can be sensitive to the choice of hyperparameters, such as the number of trees in the ensemble and the maximum depth of each tree. These hyperparameters may require careful tuning to achieve optimal performance.

10.4 Model Evaluation and Fine-tuning:

Once the model is developed, it is evaluated using metrics such as mean squared error (MSE) and R-squared. The model is then fine-tuned by adjusting its parameters and hyperparameters to improve its performance.

Using Linear regression:

****

Using SVM:

****

Using RandomForest:

****

10.5 Deployment:

Final Output:

**14 Project Plan:**

Here is a tentative project plan for developing a Laptop Price Prediction Model:

1. Data Collection and Preprocessing (4 weeks)

* Collecting laptop data from various sources
* Cleaning and preprocessing data to remove duplicates, missing values, and outliers
* Exploratory data analysis to understand the data distribution

1. Feature Engineering (2 weeks)

* Selecting the relevant features that are important in predicting laptop prices
* Creating new features by combining or transforming the existing features

1. Model Selection and Training (6 weeks)

* Evaluating multiple regression models such as linear regression, SVM, Random Forest, and XGBoost
* Training and validating the selected models on the training data
* Hyperparameter tuning to optimize model performance

1. Model Evaluation and Testing (2 weeks)

* Evaluating the model's performance on the test data
* Analyzing model errors and improving the model where necessary

1. Web Application Development (4 weeks)

* Building a user-friendly web application to provide laptop price predictions
* Integrating the model into the web application
* Testing and debugging the web application

1. Deployment and Maintenance (2 weeks)

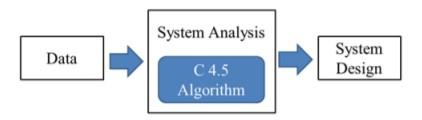
* Deploying the web application on a server or cloud platform
* Monitoring and maintaining the web application to ensure its proper functioning
* Updating the model and web application as necessary to improve performance

Note that the estimated time duration for each activity may vary depending on the complexity of the task and the experience of the development team. Additionally, some activities may be performed concurrently or may need to be repeated multiple times to ensure satisfactory results.

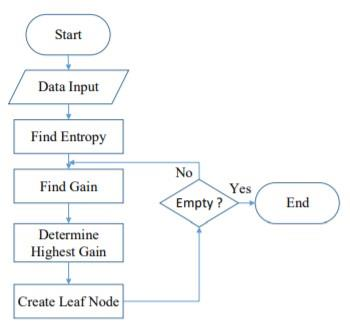
**15.System Design:**

**16 Implementation Details:**

* Algorithm and Flowchart of Implementation



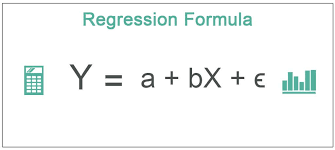
**Figure: Flow of design and analysis**



**Figure: C 4.5 algorithms flowchart**

**Equation Used in Building Model:**

**For Linear Regression:**



**link:link:**[**https://cdn.wallstreetmojo.com/wp-content/uploads/2019/04/Regression-Formula.jpg**](https://cdn.wallstreetmojo.com/wp-content/uploads/2019/04/Regression-Formula.jpg)

**For SVM:**

**The SVM algorithm works by finding the hyperplane that maximally separates the two classes of data points, represented by the equation:**

**f(x) = w^T \* x + b**

**where f(x) is the decision function that determines which side of the hyperplane a data point falls on, w is the weight vector that defines the orientation of the hyperplane, x is the input data vector, and b is the bias term that shifts the hyperplane away from the origin.**

**To predict the price of a laptop based on its specifications, the SVM algorithm uses a regression function that predicts the continuous value of the output variable, represented by the equation:**

**y = w^T \* x + b**

**where y is the predicted price of the laptop, w is the weight vector, x is the input data vector, and b is the bias term.**

**The SVM algorithm uses a kernel function to transform the input data into a higher dimensional space, represented by the equation:**

**K(x, y) = <Φ(x), Φ(y)>**

**where K(x, y) is the kernel function, Φ is the feature mapping function that maps the input data into the higher dimensional space, and <,> denotes the inner product between two vectors.**

**The SVM algorithm can be fine-tuned by adjusting the kernel function and other hyperparameters, such as the regularization parameter, to improve the accuracy of the model.**

**Random Forest Classfier:**

**18.User manual:**

**Here are the installation steps for the laptop price prediction system:**

1. **Ensure that your system meets the minimum hardware and software requirements listed in the Software and Hardware Requirements section.**
2. **Download the source code for the project from the repository or obtain it from the developer.**
3. **Install the required software, including Python 3.7 or higher, Jupyter Notebook or any other IDE for Python, and the machine learning libraries such as scikit-learn, pandas, and numpy.**
4. **Open the command prompt or terminal and navigate to the project directory.**
5. **Run the following command to start the application:**

**Copy code**

**python app.py**

1. **Once the application is running, open a web browser and navigate to the following URL:**

**arduino**

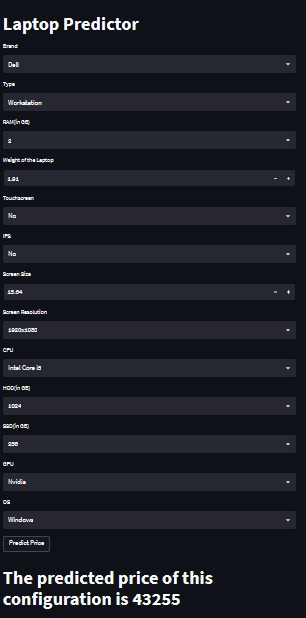
**Copy code**

**http://localhost:5000/**

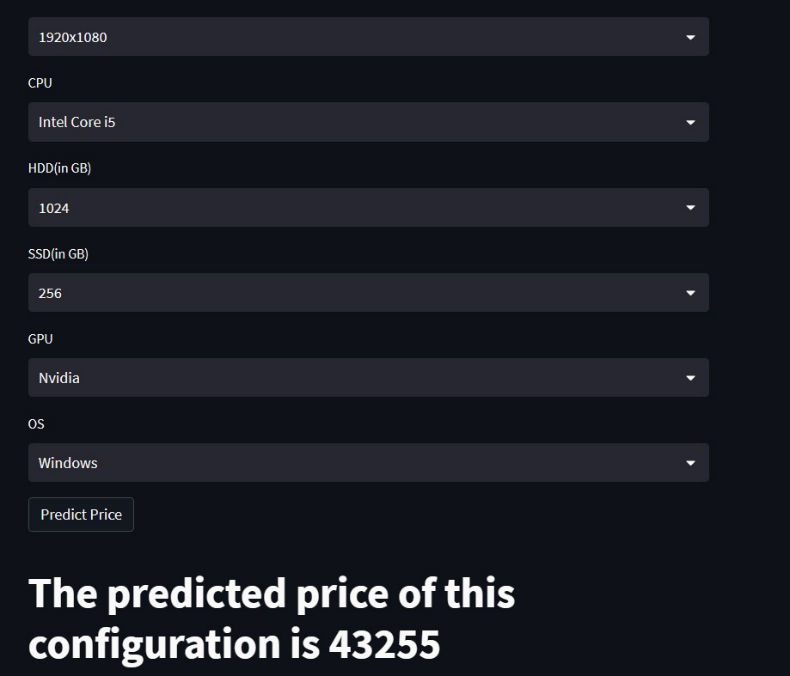
1. **The home page of the web application will be displayed, where you can input the laptop specifications to get a predicted price.**

**Here are some snapshots/screenshots of the laptop price prediction system:**

**Home page**

1. ****

**Laptop specifications form:**

****

**The user manual for the laptop price prediction system is as follows:**

1. **Open a web browser and navigate to the following URL:**

**Copy code**

**http://localhost:5000/**

1. **Enter the laptop specifications, including the brand, model, processor, RAM, storage, and screen size.**
2. **Click on the "Predict" button.**
3. **The predicted price for the laptop will be displayed on the screen.**
4. **If you want to predict the price for another laptop, click on the "Reset" button to clear the form and start again.**
5. **If you encounter any issues with the application, contact the developer for support**

**19.Conclusion and Future work:**

**In conclusion, the laptop price prediction model is a useful tool that allows users to predict the price of a laptop based on its specifications. The model utilizes machine learning algorithms such as linear regression, support vector machines, and random forests to make accurate predictions. The system was designed to be user-friendly, accurate, reliable, and scalable, with measures in place to protect user data and prevent unauthorized access.**

**However, there is always room for improvement and future work. Some possible areas for improvement include:**

1. **Incorporating more data sources to improve the accuracy of the model.**
2. **Developing a mobile application to increase the accessibility of the system.**
3. **Implementing a recommendation system to suggest laptops based on user preferences and budget.**
4. **Adding more features to the web application, such as user profiles, search functionality, and social sharing.**
5. **Enhancing the user interface and making it more visually appealing.**

**In summary, the laptop price prediction model has a lot of potential for future development and expansion, and we look forward to seeing how it can be improved to better serve its users.**

**20.Conclusions:**

**Predicting something through the application of machine learning using the Decision Tree algorithm makes it**

**easy for students, especially in determining the choice of laptop specifications that are most desirable for**

**students to meet student needs and in accordance with the purchasing power of students. Students no longer**

**need to look for various sources to find laptop specifications that are needed by students in meeting the needs of**

**students, because the laptop specifications from the results of the machine learning application have provided**

**the most desirable specifications with their prices of laptops.**

# CHAPTER 8

# CONCLUSION

**7.1 Conclusion**

After this project work and research our conclusion is that we can identify the best prices by using the trained machine learning model. But for some old things or products it is very difficult to identify their prices but by extending the databases we can reduce the difficulty. Also, when number of inputs is more in the frame then it’s very tough to getting best prices of them. But we learnt a lot from it. Our research still is going on it. Another thing is that we need a lot of positive and true values in dataset for training the machine learning Model.

### **Bibliography**

* **Cars24 :** Sell Your Car Safely with Cars24. Book Home Inspection or Visit a Cars24 Store Near You. Sell Your Car at the Comfort of Your Home.

**Website:** [**https://www.scars24.com/**](https://www.scars24.com/)

* **Car Dekho :** Thinking of buying a car? At CarDekho.com, buy new and used cars, search by filter and preferences, compare cars, read latest news and updates.

**Website:** [**https://www.cardekho.com**](https://www.cardekho.com)

* **Spinny:** Spinny is the most trusted way of buying and selling used cars. Choose from over 5000 fully inspected second-hand car models.

**Website:** [**https://www.spinny.com/**](https://www.spinny.com/)

### **References**

**[1]** TEM Journal. Volume 8, Issue 1, Pages 113-118, ISSN 2217-8309,

DOI:10.18421/TEM81-16, February 2019.

TEM Journal – Volume 8 / Number 1 / 2019. Car Price Prediction using

Machine Learning Techniques Enis Gegic, Becir Isakovic, Dino Keco.

**[2]** Geurts P. (2009) Bias vs Variance Decomposition for Regression and Classification. In: Maimon O., Rokach L. (eds) Data Mining and Knowledge Discovery Handbook. Springer, Boston, MA

**[3]** Robert T. (1996) Regression Shrinkage and Selection Via the Lasso. In: Journal of the Royal Statistical Society: Series B Methodological) Volume 58, Issue 1

**[4]** Hastie, Trevor, and Daryl Region. Shrinking trees. AT & T Bell Laboratories, 1990. [5] Kim, Tae Kyun. "Understanding one-way ANOVA using conceptual figures." Korean journal of anesthesiology 70.1 (2017).

**[5]** Kim, Tae Kyun. "Understanding one-way ANOVA using conceptual figures." Korean journal of anesthesiology 70.1 (2017): 22.

**Figure 1 :** <https://images.app.goo.gl/ss2iRrEELe1P7ohr6>

**Figure 2 :** <https://images.app.goo.gl/SdaZRtYMzPF4zvXdA>

**Figure 4 :** <https://images.app.goo.gl/SdaZRtYMzPF4zvXdA>

**Figure 7 :** <https://images.app.goo.gl/yVnVz3zuxEWwtkau9>

**Figure 8 :** <https://www.wallstreetmojo.com/regression-formula/>

**Figure 9 :** <https://www.tibco.com/reference-center/what-is-a-randoforest>

**Figure 10 :** <https://www.vedantu.com/maths/mean-squared-error>