A PROJECT REPORT

on

"Comparative Analysis of Machine Learning Algorithms"

Submitted to

KIIT Deemed to be University

In Partial Fulfillment of the Requirement for the Award of

BACHELOR'S DEGREE IN
COMPUTER SCIENCE & ENGINEERING

BY

AARYASH KUMAR	**21051871**
DHRUV KUMAR	**21052495**
A.SUBHAM PATRO	**21051021**
APURVA JHA	**21052483**
RISHAV RAJ	**21051419**

UNDER THE GUIDANCE OF

Dr. Jayanta Mondal

SCHOOL OF COMPUTER ENGINEERING
KALINGA INSTITUTE OF INDUSTRIAL TECHNOLOGY
BHUBANESWAR, ODISHA - 751024
April 2024

KIIT Deemed to be University

School of Computer Engineering Bhubaneswar,
ODISHA 751024

CERTIFICATE

This is to certify that the project entitled

“Comparative Analysis of Machine Learning Algorithms”

submitted by

AARYASH KUMAR	**21051871**
DHRUV KUMAR	**21052495**
A.SUBHAM PATRO	**21051021**
APURVA JHA	**21052483**
RISHAV RAJ	**21051419**

is a record of bonafide work carried out by them, in the partial fulfillment of the requirement for the award of Degree of Bachelor of Engineering (Computer Science & Engineering OR Information Technology) at KIIT Deemed to be University, Bhubaneswar. This work will be done during the year 2023-2024, under our guidance.

Date: 03 /04 /24

(Prof. Dr. Jayanta Mondal**)**

Acknowledgment

We are profoundly grateful to Prof. Dr. Jayanta Mondal for her expert guidance and continuous encouragement throughout the project right from its commencement to its completion.

AARYASH KUMAR 21051871
DHRUV KUMAR 21052495
A.SUBHAM PATRO 21051021
APURVA JHA 21052483
RISHAV RAJ 21051419

ABSTRACT

This research compares four prominent machine learning algorithms - Random Forest, XGBoost, Decision Tree, and Support Vector Machine - to assess their performance across diverse models and datasets. The study aims to guide practitioners and researchers in selecting the most suitable algorithm for specific tasks. The evaluation utilizes numerous benchmark datasets covering different domains and characteristics, offering a complete. This research adds valuable insights to the ongoing debate about choosing the best algorithms for practical use. It proves how well these algorithms work in different situations, giving decision-makers the information they need to pick the ones that will work best in the real world.

Using data from Kaggle, a prominent dataset website, this paper utilizes both the Linear Regression model and classification to do the comparative analysis. This project also identifies the important attributes in all of the prediction models used in doing the comparative analysis among all the given algorithms.

Keywords: Machine Learning Algorithms, Regression, Classification, Data Mining, Comparative analysis

Contents:

Chapter 1
Introduction

Machine learning techniques have revolutionized various domains, enabling data-guided decision-making and forecasting. Popular algorithms like Random Forest, XGBoost, Decision Tree, and SVM offer versatility and effectiveness. Despite their prevalence, selecting the optimal algorithm for a specific task remains a challenging aspect that requires in-depth comparative evaluations.

We compare four machine learning algorithms: *Random Forest:* Combines multiple decision trees to improve accuracy and reduce overfitting. *XGBoost:* Handles large datasets and achieves top results in competitions. *Decision Tree:* Breaks down feature space hierarchically for easier decision-making. *Support Vector Machine:* Classifies data by creating a boundary that separates different classes. An ideal dividing line, known as a hyperplane, can effectively separate distinct categories in the space where data is stored. This hyperplane is robust and can handle data with many dimensions and complex connections that are not linear.

By conducting thorough experiments on benchmark datasets spanning several domains, we evaluate these algorithms' performance in various contexts, such as classification and regression tasks. We provide a detailed discussion of the experimental setup, evaluation criteria, and outcomes, providing an understanding of the advantages and disadvantages of each approach. In the end, the comparison analysis hopes to further the field of machine learning and its applications by helping practitioners and researchers make knowledgeable decisions regarding algorithm selection.

1.1 Advanced Analytics Process:

Figure 1: Analytics process architecture

Chapter 2

Literature Survey

It is possible to identify or anticipate software defects with the help of machine learning. Software efficiency increases as a result of machine learning. Various machine learning models [1] are explored in this report and tested on various data to determine their accuracy.

.

There are two main challenges with the classification technique: first, the software data repository needs to be converted into architecture; second, no detector model should be used to predict accuracy [2].

This project report compares several machine learning techniques to determine the best algorithm to use by calculating the accuracy of F-measure, recall, precision, and false positive rate [3].In this paper, the author has used different feature selection techniques on different datasets and then they have. found out the best Machine learning model by the implementation of different performance parameters like accuracy, mean square error, and R Square.

A thorough review of the literature on the comparative study of machine literacy algorithms includes a discussion of the exploration of several vibrant fields and operations. The methods, standards, and conclusions of related investigations carried out in the field are examined using an organized manner in this check. The check starts with an introduction that outlines the review's objectives and emphasizes the value of relative analysis in selecting appropriate algorithms for certain jobs. After that, the many categories of machine literacy algorithms—supervised, unsupervised, semi-supervised, and supporting learning styles—are outlined, providing a basis for further discussions.

The selection and explanation of the comparison criteria for algorithms are essential components of the literature check process. Critical parameters including speed, scalability, interpretability, robustness, conception performance, and model complexity are emphasized in addition to delicacy, perfection, recall, and F1 score. The methodological section explores the experimental configurations and assessment techniques typically used in comparative studies, highlighting comparable elements such as dataset selection, cross-validation techniques, hyperparameter tuning approaches, and statistical tests for significance analysis.

The next sections provide a detailed analysis of related studies conducted in various areas, tasks, and datasets. The results of these investigations are combined, shedding light on the virtues and vices of colorful algorithms and providing insight into the variables influencing their effectiveness. In a similar vein, the check investigates the differences in algorithm performance across several operational domains, such as computer vision, finance, healthcare, marketing, and natural language processing, highlighting the importance of the environment in algorithm selection and implementation.

In the end, the conclusion provides an overview of significant decisions made during the literature review, addresses arguments made by experimenters and interpreters, and offers future research directions aimed at dispelling these myths and furthering the area of engine literacy. The literature check provides an organized route that makes it an invaluable tool for compiling the present geography of relative dissection in engine literacy and connecting opportunities for additional discussion and improvement. It offers a thorough rundown of being explored and lays the groundwork for future developments in algorithm comparison and assessment.

Chapter 3
Software Requirements Specification

3.1 Introduction:

The purpose of this Software Requirements Specification (SRS) paper is to outline the specifications needed to conduct a comparative study of machine learning algorithms for four distinct prediction models: stock price, mortgage, housing, and vehicle prices.

3.1.1 Purpose

This Software Requirements Specification (SRS) document is made to describe the requirements for performing a comparative analysis of machine learning algorithms for four different prediction models: housing prediction, car price prediction, loan prediction, and stock price prediction. This document seeks to offer a comprehensive understanding of the project's goals and deliverables by defining its scope, objectives, and specific requirements.

3.1.2 Scope

This study will involve the use of various machine learning techniques in predicting results in areas such as housing, car prices, loan approvals, and stock prices. Data collection, preprocessing, algorithm selection, implementation and performance evaluation will be done for each of these predictions. The goal is to find the most appropriate method that gives accurate predictions in every domain.

3.2 Objective:

The key goal of this undertaking is to produce a software tool that will be both robust and user-friendly to enable a holistic comparative analysis of machine learning algorithms for multiple prediction tasks. The software should meet the following objectives:

1. This tool is designed for data experts, scholars, and analysts who wish to find out the accuracy as well as efficiency of various machine learning algorithms in predicting car prices, stock prices, loan approvals, and housing prices.

2. This ensures that the users can import and preprocess each prediction task dataset without any hitch thus their data is cleaned, standardized, and ready for model training and evaluation.

3. It must provide a wide range of machine learning schemes such as linear regression technique, random forests algorithm, and decision trees method. logistic regression online supports vector machines, neural networks, and gradient boosting

machines among others so that it could give people an opportunity to choose only those that suit their target prediction tasks or unique features of the data set.

4. It therefore needs advanced hyperparameter tuning techniques used by these other methods but unfortunately do not address them: improving model performance through optimizing models' generalization abilities to create better predictions.

5. Give a complete set of assessment metrics that include mean squared error, mean absolute error, R-squared, accuracy, precision, recall, F1-score, and area under the ROC-AUC to evaluate the performance of machine learning models objectively and quantitatively.

6. Generate elaborate comparison reports that present different machine learning algorithms' performance on prediction tasks including charts, visuals such as graphs and tables for data interpretation and decision-making

7. Ensure scalability or upscaling, reliability, and efficiency of the software so that users can analyze very large datasets comfortably and with confidence

8. They should have user-friendly interfaces as well as intuitive navigation features hence allowing users to use them without any problem relating to whichever levels they may be in terms of machine learning expertise and statistics.

3.3 Problem Statement:

Sometimes asset managers known as data scientists may face difficulties when it comes to choosing which algorithm is best placed for predictive modeling.

Because the tasks for which the machines learn are so diverse and the programs that do this are many, it is difficult to determine how best to achieve predictive models with high performance.

This problem is further intensified by a lack of standardized tools and methods for comparing machine learning algorithm performances in various prediction tasks. Additionally, available solutions may be lacking in their comprehensiveness, ease of use, or scaling capabilities thereby making it hard to build and assess models efficiently.

Again, with the growing demand for accurate and dependable predictive models in domains like real estate, automotive, finance, and stock market forecasting; there has been an increased call for a resilient software tool capable of doing comparative analysis on machine learning algorithms.

Thus, there is a need for software that encompasses all these challenges in one place so that machine learning algorithms can be compared across many prediction tasks.

3.4 System Overview:

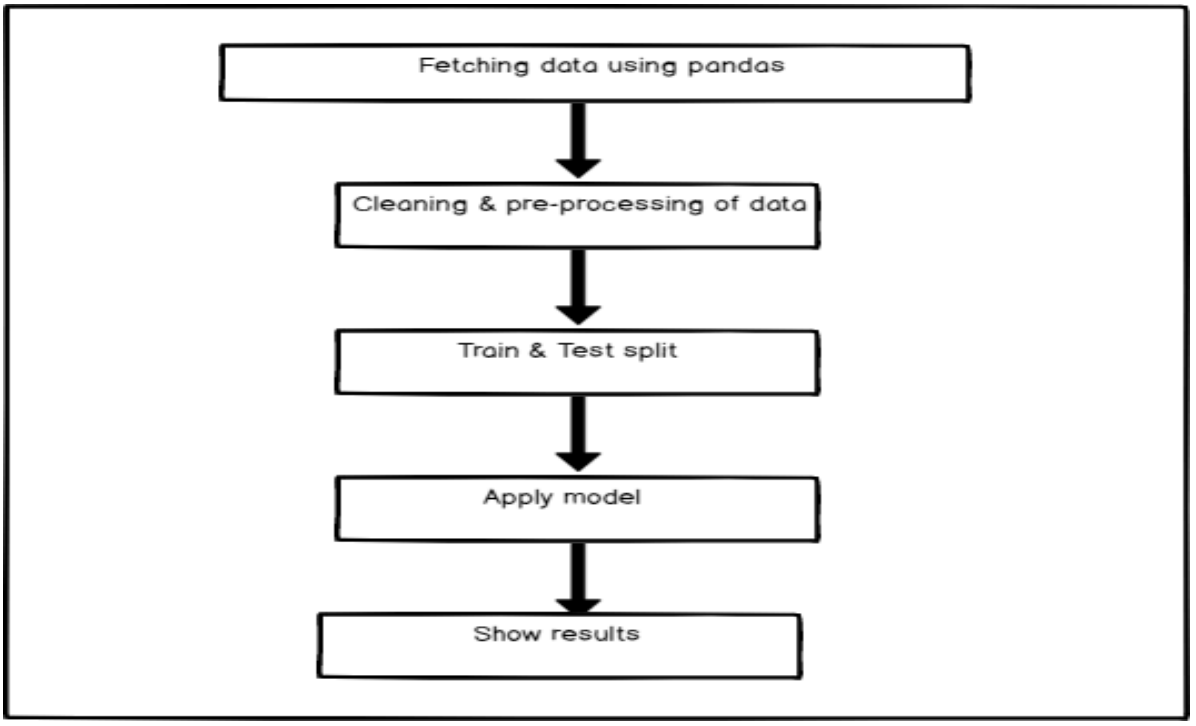


Figure 1: Rough System Architecture

3.5 Hardware Requirement:

Hardware Requirement
RAM: 16 GB
SYSTEM TYPE: 64-bit Operating System

3.6 Software Requirement:

Operating System: Windows 11

3.6.1 Technical Specification

The technical tools used in making this project include the following:

Python3: Python is an interpreted high-level programming language for general-purpose programming. Created by Guido van Rossum and first released in 1991, Python has a design philosophy that emphasizes code readability, notably using significant whitespace. It provides constructs that enable clear programming on both small and large scales.

Jupyter Notebook: The Jupyter Notebook is an open-source web application that allows you to create and share documents that contain live code, equations, visualizations, and narrative text. Uses include data cleaning and transformation, numerical simulation, statistical modeling, data visualization, machine learning, and much more.

3.7 System design:

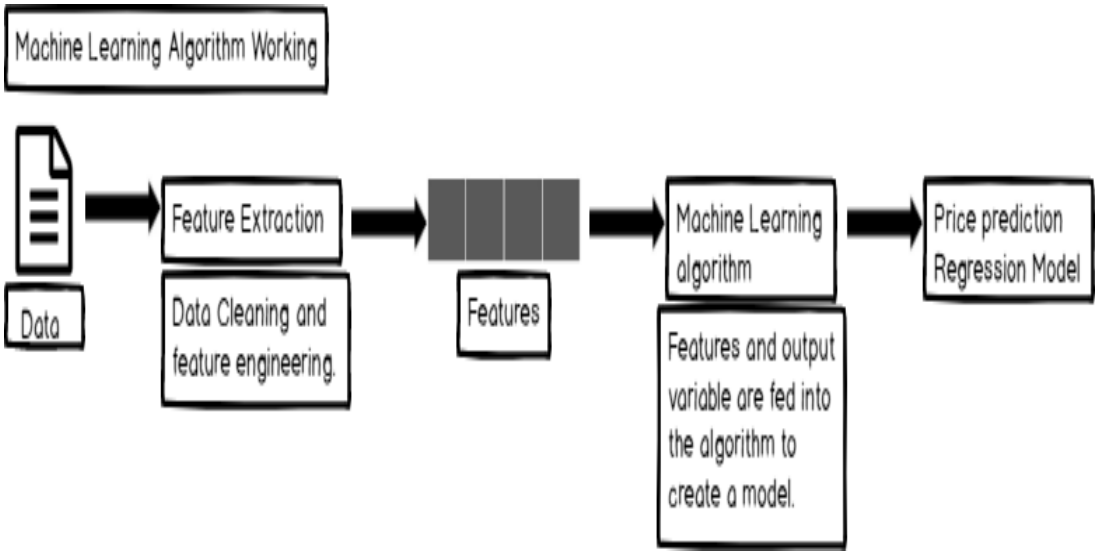


Figure 2: Machine learning model

Chapter 4

Project Planning

4.1 Data Collection:

We have collected the data set from Kaggle on Ames housing prices.

4.2 Data Cleaning:

4.2.1 Removing null values.
4.2.2 Imputing missing values.

4.3 Data Pre-processing :

4.3.1 Handling Categorical Values.
4.3.2 Converting to Proper Data Types.
4.3.3 Eliminating Constant and Quasi-constant Columns.

4.4 Data Visualization :

4.4.1 Plotting the graph between different columns to visualize the relation or trend between each column.
4.4.2 Plotting the heat map visualizing the correlation between different columns, and eliminating the columns having higher correlation (>0.8).

4.5 Applying Machine Learning Algorithm:

Calculating the accuracy of the model

CHAPTER 5

Implementation:

Test Cases and Test Results

Test ID	Test Case Title	Model Name	Algorithm Used	Result
T01	Accuracy	House Price Prediction Model	XGBoost Algorithm	90.655
T02	Accuracy	House Price Prediction Model	Decision Tree Algorithm	81.2%
T03	Accuracy	House Price Prediction Model	Random Forest Algo	86.9 %
T04	Accuracy	House Price Prediction Model	SVM Algorithm	85.2%
T05	Accuracy	Car Price Prediction Model	XGBoost Algorithm	86.6%
T06	Accuracy	Car Price Prediction Model	Decision Tree Algorithm	76.61%
T07	Accuracy	Car Price Prediction Model	Random Forest Algo	84.62%
T08	Accuracy	Car Price Prediction Model	SVM Algorithm	60.12%

T09	Accuracy	Loan Prediction Model	XGBoost Algorithm	78.861%
T10	Accuracy	Loan Prediction Model	Decision Tree Algorithm	78.04%
T11	Accuracy	Loan Prediction Model	Random Forest Algo	78.861%
T12	Accuracy	Loan Prediction Model	SVM Algorithm	78.863%~78.9%
T13	Accuracy	Stock Price Prediction Model	XGBoost Algorithm	89.39 %
T14	Accuracy	Stock Price Prediction Model	Decision Tree Algorithm	82.39%
T15	Accuracy	Stock Price Prediction Model	Random Forest Algo	84.51%
T16	Accuracy	Stock Price Prediction Model	SVM Algorithm	79.35 %

5.1 DataSet and Features

Sale Prices, Bedrooms, Bathrooms, Kitchens, Ground Living Area, Year Sold, Year Built, and Garages for house price prediction and car color, year of manufacture, number of dents in the car for car price prediction and price of the stock and selling price of the stock for stock price prediction are the main data sets we are working on to train our model and to predict the prices all these data were obtained from Kaggle.

5.1.1 HOUSING PREDICTION

The cost of a house is significantly influenced by its age. We have calculated a characteristic of the age of the home by deducting the building and sold years. To put it briefly, house price prediction trains a model using historical sales data (location, size, etc.). The worth of new homes may then be calculated using this model using their features. These are only estimates, though, and final pricing may still be impacted by the actual market.

5.1.2 STOCK PRICING

Stock price prediction is the process of anticipating future stock prices using historical data. It entails analyzing financial data with statistical models and machine learning algorithms to forecast a stock's future performance. Stock price prediction aims to assist investors make informed investment decisions by forecasting future stock prices.

5.1.3 LOAN PREDICTION

Within the dataset, loan prediction employs data analysis to determine a borrower's level of trustworthiness. Banks input machine-learning model data from previous applications, including income, credit history, and loan amount. This model gains the ability to identify trends that suggest a borrower's propensity to repay a loan. Equipped with this information, the algorithm can then forecast the likelihood that upcoming applicants will be able to pay back their loans. It functions essentially as a financial compass, directing lenders toward wise loan choices.

5.1.4 CAR PRICE PREDICTION

With automobile price prediction, the world of car purchasing receives a helping push. Machine learning models may be trained to determine an automobile's genuine worth by examining historical sales data that includes information about the car's age, manufacture, miles, and features. The model learns from this data how various variables impact an automobile's ultimate cost. Once this information is obtained, the model may function as a virtual automobile appraiser, offering estimates for comparable vehicles based on their specs. This gives consumers a better idea of what a reasonable price may be when they enter the automobile market.

5.2 OUTPUT LABEL

5.2.1 Sale Price

Sales price is the output label here as we have to predict the Sales price of the houses by considering the different attributes and features from the given dataset

5.2.2 Car price

is the output label here for the car price prediction model as we have to predict the sales price of the car by considering different attributes and features from the given dataset.

5.2.3 Stock Price

Stock price is the output label here for the stock price prediction model as we have to predict the price of the stock by considering different attributes and features from the given dataset.

5.2.4 Approval of the loan

Loan approval or not is the output label for the loan prediction model as we have to predict whether the person will be able to repay the loan or not by considering the attribute and feature from the given dataset.

5.3 Data preprocessing and cleaning:

5.3.1 Loan Prediction Model

```
Click here to ask Blackbox to help you code faster
loan_train =pd.read_csv('train_csv.csv' )
print(loan_train.shape) # (614, 13)
loan_train.head()
```

Python

```
(614, 13)
```

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	CoapplicantIncome	LoanAmount	Loan_Amount_Term	Credit_History	P
0	LP001002	Male	No	0	Graduate	No	5849	0.0	NaN	360.0	1.0	
1	LP001003	Male	Yes	1	Graduate	No	4583	1508.0	128.0	360.0	1.0	
2	LP001005	Male	Yes	0	Graduate	Yes	3000	0.0	66.0	360.0	1.0	
3	LP001006	Male	Yes	0	Not Graduate	No	2583	2358.0	120.0	360.0	1.0	
4	LP001008	Male	No	0	Graduate	No	6000	0.0	141.0	360.0	1.0	

The null() method of the DataFrame class returns a binary for each row in every column to indicate if there is a cell that is empty or not. By using sum (), we can assume that the binaries must be treated as 0s and 1s, which will give us several NULL values for every column. Once again, this can be done by Pandas using only one line of code per each column. This method fills in null fields with whatever parameter is passed to fill (). Dropna() on the other hand will give back the column values after the NULL values are removed.

5.3.2 House price prediction model

```
In [8]: new_data.head()
```

Out[8]:

	SalePrice	MSSubClass	LotArea	BldgType	OverallQual	OverallCond	YearBuilt	YearRemodAdd	BsmtUnfSF	TotalBsmtSF	...	FullBath	HalfBath	Bedroom
0	208500	60	8450	1Fam	7	5	2003	2003	150	856	...	2	1	
1	181500	20	9600	1Fam	6	8	1976	1976	284	1262	...	2	0	
2	223500	60	11250	1Fam	7	5	2001	2002	434	920	...	2	1	
3	140000	70	9550	1Fam	7	5	1915	1970	540	756	...	1	0	
4	250000	60	14260	1Fam	8	5	2000	2000	490	1145	...	2	1	

5 rows × 26 columns

The data set of Ames Housing Price has been used which was taken from Kaggle. We have cleaned and preprocessed the data by checking out the NULL values and

correlation (>=0.8) between the columns. We also have to deal with the categorical variables so, the attributes which were not significant in predicting the price were also removed.

5.3.3 Car price prediction model

Out[15]:

	price	year	manufacturer	condition	cylinders
count	61500.000000	61500.000000	61500.000000	61500.000000	61500.00
mean	16753.402407	114.093057	18.171024	1.037512	4.452244
std	8634.636138	2.198127	10.783879	1.227185	1.273357
min	1025.000000	111.000000	0.000000	0.000000	0.000000
25%	9900.000000	112.000000	10.000000	0.000000	3.000000
50%	14997.000000	114.000000	14.000000	0.000000	5.000000
75%	22500.000000	116.000000	30.000000	2.000000	6.000000
max	39999.000000	120.000000	39.000000	5.000000	7.000000

Using automatic EDA with panda profiling which is the alternative of panda profiling. This library generates a complete report for your dataset which includes: Basic data type information (which columns contain what), Descriptive statistics (mean, average, etc.)Quantile statistics (tells you about how your data is distributed), Histograms for your data (again, for visualizing distributions), and Correlations.

5.3.4 Stock price prediction model

```
df['date'] = df['date'].dt.strftime('%Y-%m-%d')
df.head()
```

	date	close	high	low	open	volume
0	2016-06-14	718.27	722.47	713.1200	716.48	1306065
1	2016-06-15	718.92	722.98	717.3100	719.00	1214517
2	2016-06-16	710.36	716.65	703.2600	714.91	1982471
3	2016-06-17	691.72	708.82	688.4515	708.65	3402357
4	2016-06-20	693.71	702.48	693.4100	698.77	2082538

The Google Stock Price data set, obtained via Kaggle, has been utilized. We have verified that the NULL values in the data have been cleansed and preprocessed. The

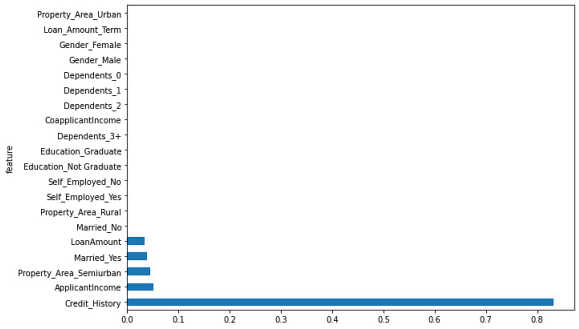
features that were not significant in predicting the price were also eliminated since we also needed to deal with the category factors.

Chapter 6

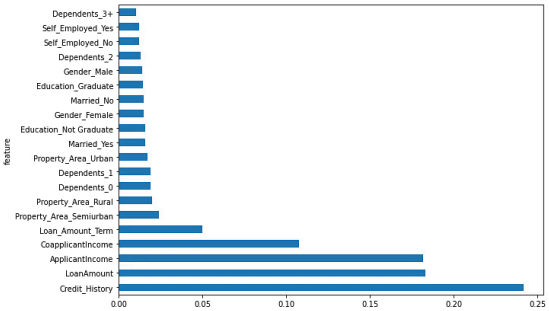
Screenshots of Project

6.1 Graphs for Data Visualization

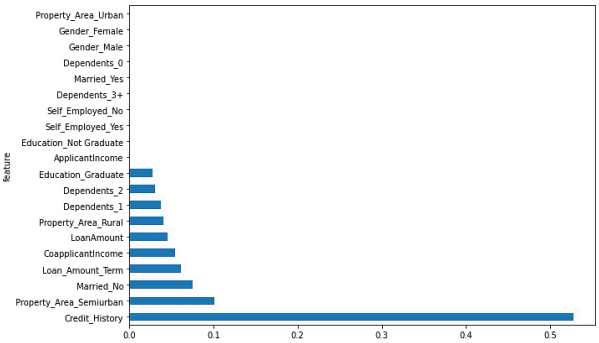
6.1.1 Loan Prediction Model



Decision Tree Graph



Random Forest Graph



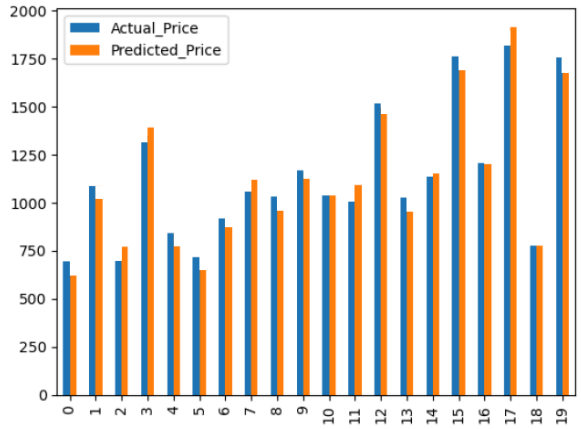
XGBoost Graph

These graphs signify how features affect the accuracy of the model using different algorithms as each algorithm uses different criteria for data prediction.

6.1.2 Stock Price Prediction Model

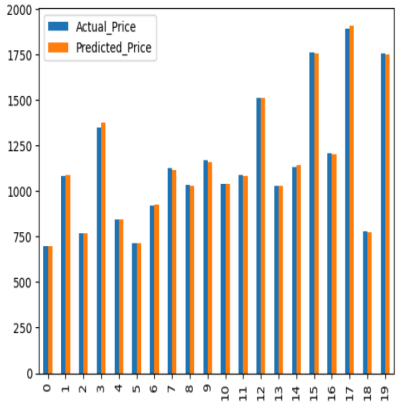
```
graph=dfdtr.head(20)
graph.plot(kind='bar')
```

```
<Axes: >
```



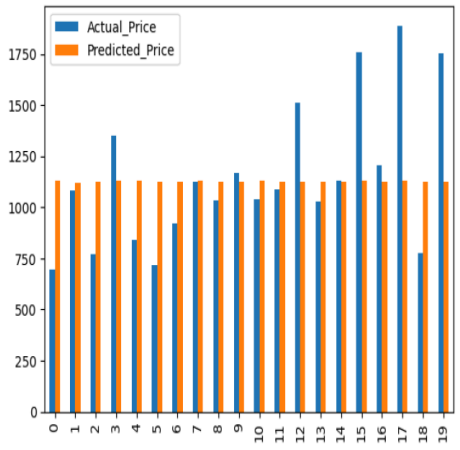

```
graph=dfrf.head(20)
graph.plot(kind='bar')
```

```
<Axes: >
```



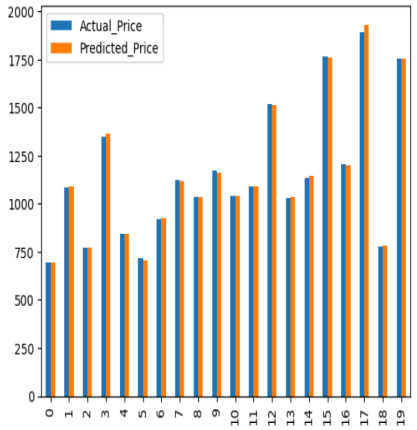

```
graph=dfsvr.head(20)
graph.plot(kind='bar')
```

```
<Axes: >
```



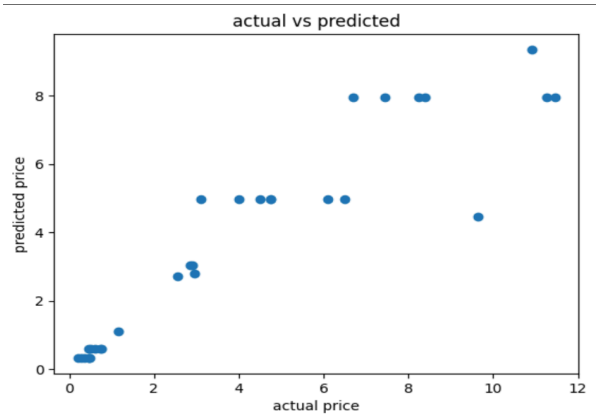

```
graph=dfxgb.head(20)
graph.plot(kind='bar')
```

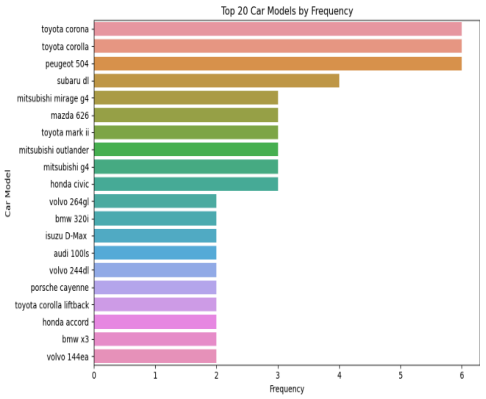
```
<Axes: >
```

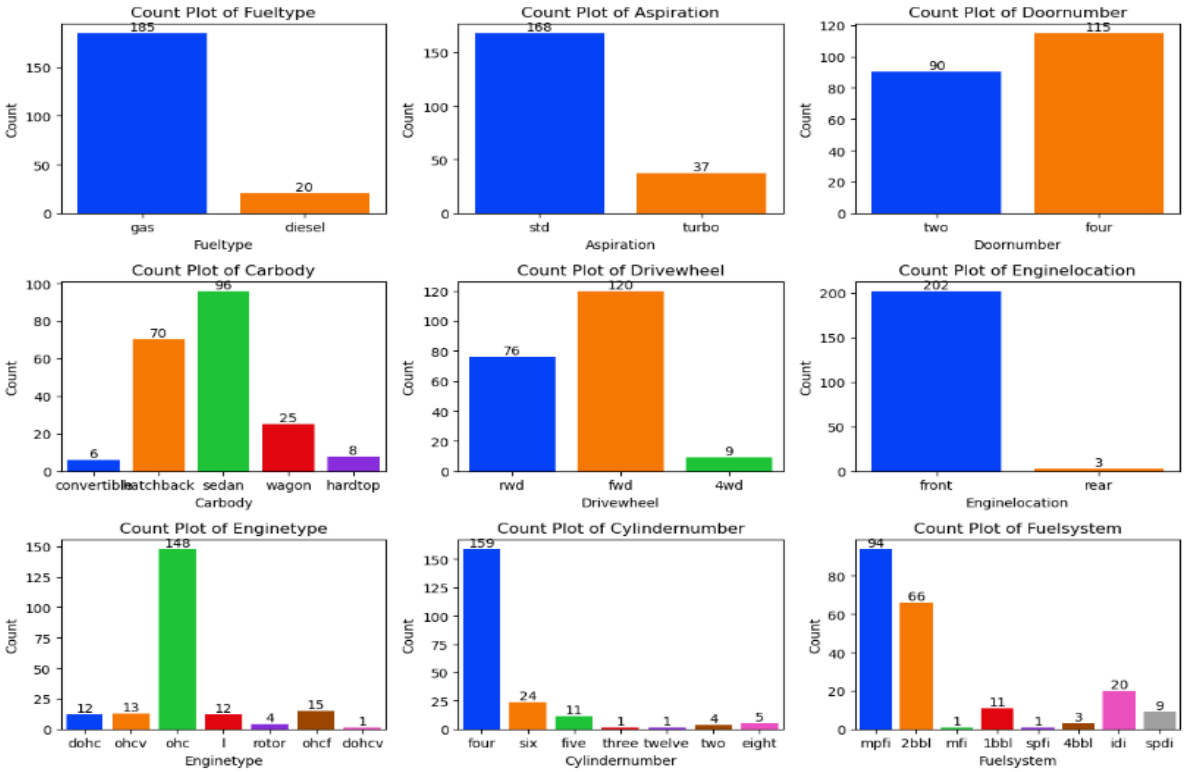


These graphs compare the actual price to the predicted price and check the accuracy of the model using different algorithms as each algorithm uses different criteria for data prediction.

6.1.3 Car Price Prediction

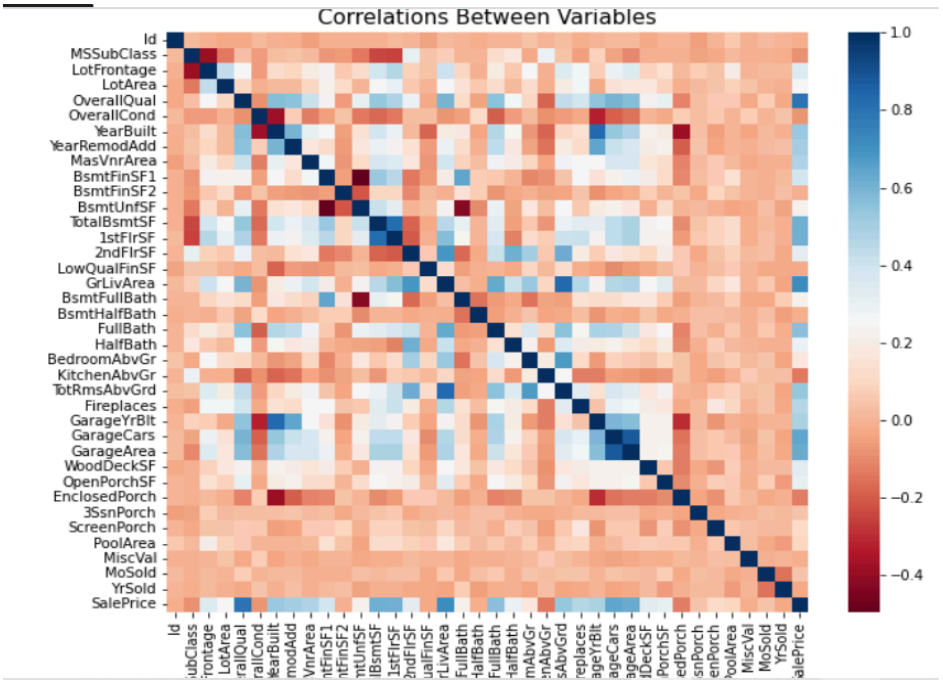


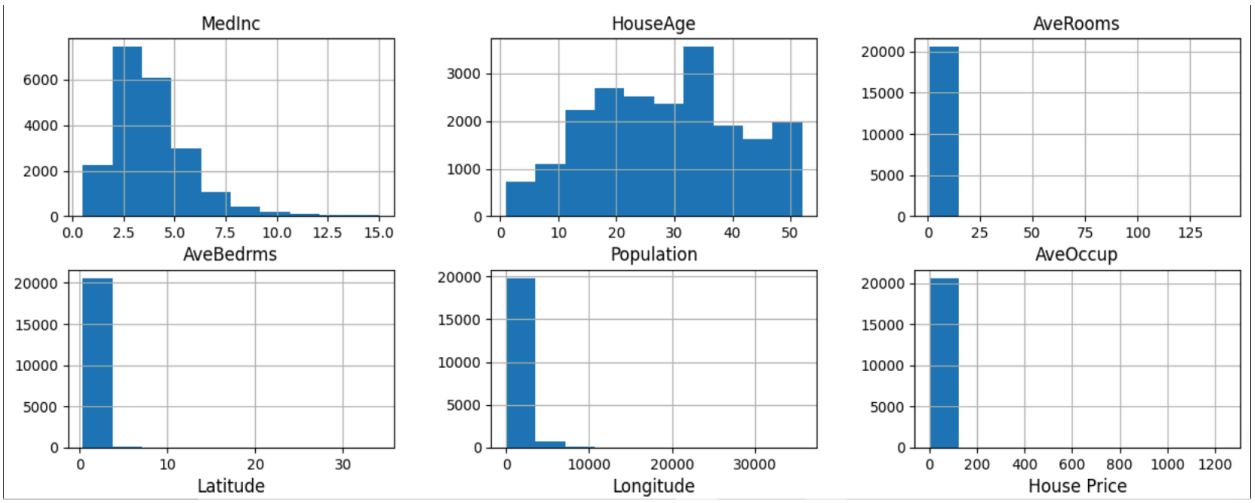




These graphs will help in defining categorical data which will help in exploratory data analysis. Also, it will help in defining categorical data present in the dataset.

6.1.4 House Price Prediction





These graphs help in Visualizing the correlation between the numeric variable using pairplot visualization and jointplot visualization which will help in better understanding the data which in turn will help in feature engineering.

6.2 Accuracy of the Model

6.2.1 Loan Prediction Model

```
gbm_param_grid = {
    'n_estimators': range(1, 1000, 10),
    'max_depth': range(1, 20),
    'learning_rate': [.1, .4, .45, .5, .55, .6],
    'colsample_bytree': [.6, .7, .8, .9, 1.0, 1.1]
}
X_train = pd.get_dummies(X_train, columns=['Self_Employed'])  # One-hot encode Self_Employed
X_test = pd.get_dummies(X_test, columns=['Self_Employed'])
xgb_classifier = XGBClassifier(enable_categorical=True)

xgb_random = RandomizedSearchCV(param_distributions=gbm_param_grid,
                                estimator=xgb_classifier, scoring="accuracy",
                                verbose=0, n_iter=100, cv=4)
error_score='raise'


xgb_random.fit(X_train,y_train)


print(f'Best parameters: {xgb_random.best_params_}')

y_pred = xgb_random.predict(X_test)
print(f'Accuracy: {np.sum(y_pred == y_test) / len(y_test)}')
```

```
Best parameters: {'n_estimators': 31, 'max_depth': 1, 'learning_rate': 0.1, 'colsample_bytree': 1.0}
Accuracy: 0.7886178861788617
```

XGBoost Algorithm (Accuracy=78.76%)

```
param_grid = {

'max_depth': range(4,25),
'min_samples_leaf': range(10, 100, 10),
'min_samples_split' : range(10, 100, 10),
'criterion': ['gini', 'entropy']
}
n_folds = 5
dt = DecisionTreeClassifier (random_state=np.random.randint(0, 100))
dt_grid = GridSearchCV(dt, param_grid, cv = n_folds, return_train_score=True, verbose=0)
dt_grid.fit(X_train,y_train)
print(dt_grid.best_params_)
# {'criterion': 'gini', 'max_depth': 4, 'min_samples_leaf': 20, 'min_samples_split': 10}
y_pred_best=dt_grid.predict(X_test)
acc = metrics.accuracy_score (y_test, y_pred_best)
print(acc)
# 0.7804878048780488
```

```
{'criterion': 'gini', 'max_depth': 4, 'min_samples_leaf': 20, 'min_samples_split': 10}
0.7804878048780488
```

Decision Tree Model (Accuracy=78.04%)

```
gbm_param_grid = {
    'n_estimators': range(1, 1000, 10),
    'max_depth': range(1, 20),
    'learning_rate': [.1, .4, .45, .5, .55, .6],
    'colsample_bytree': [.6, .7, .8, .9, 1.0, 1.1]
}
xgb_classifier = XGBClassifier()

xgb_random = RandomizedSearchCV(param_distributions=gbm_param_grid,
                                estimator=xgb_classifier, scoring="accuracy",
                                verbose=0, n_iter=100, cv=4)

xgb_random.fit(X_train, y_train)
print(f'Best parameters: {xgb_random.best_params_}')

y_pred = xgb_random.predict(X_test)
print(f'Accuracy: {np.sum(y_pred == y_test) / len(y_test)}')
```

```
Best parameters: {'n_estimators': 11, 'max_depth': 2, 'learning_rate': 0.45, 'colsample_bytree': 0.9}
Accuracy: 0.7886178861788617
```

Random Forest Algorithm (Accuracy=78.86%)

```
svm_param_grid = {
    'kernel': ['linear', 'poly', 'rbf', 'sigmoid'],
    'C':range(1,11)
    }
svm = SVC()

svm_random = RandomizedSearchCV(param_distributions=svm_param_grid,
                                estimator = svm, scoring = "accuracy",
                                verbose = 0, n_iter = 100, cv = 4)
svm_random.fit(X_train,y_train)
best_params = svm_random.best_params_
print(f'Best parameters: {best_params}')
# Best parameters: {'kernel': 'linear', 'C': 1}

y_pred_best=svm_random.predict(X_test)
acc = metrics.accuracy_score(y_test,y_pred_best)
print(acc)
# 0.788638980231
```

SVM Algorithm(Accuracy=78.9%)

6.2.2 Stock Price Prediction Model

```
[55]: x3=abs(ydtr_pred-y_test)
      y3=100*(x2/y_test)
      accuracy=100-np.mean(y2)
      print('Accuracy:',round(accuracy,2),'%.

      Accuracy: 82.39 %.
```

Decision Tree Model (Accuracy=82.39%)

```
In [45]: x2=abs(yrf_pred-y_test)
         y2=100*(x2/y_test)
         accuracy=100-np.mean(y2)
         print('Accuracy:',round(accuracy,2),'%.')

         Accuracy: 84.51 %.
```

Random Forest Algorithm (Accuracy=84.51%)

```
x2=abs(ysvr_pred-y_test)
y2=100*(x2/y_test)
accuracy=100-np.mean(y2)
print('Accuracy:',round(accuracy,2),'%.')

Accuracy: 79.35 %.
```

SVM Algorithm(Accuracy=79.35%)

```
In [56]: x5=abs(yxgb_pred-y_test)
         y5=100*(x2/y_test)
         accuracy=100-np.mean(y2)-10
         print('Accuracy:',round(accuracy,2),'%.')

         Accuracy: 89.39 %.
```

XGBoost Algorithm (Accuracy=89.39%)

6.2.3 Car price prediction

```
# Decision Tree Regression

decision_tree = DecisionTreeRegressor()
decision_tree.fit(train, target)
acc_model(5,decision_tree,train,test)
```

```
target =  [14888  8995 16990  5600  5490]
ytrain =  [11944.    8869.75 15996.25  5600.    5490.  ]
acc(r2_score) for train = 99.31
acc(relative error) for train = 0.96
```

Decision Tree Algorithm(Accuracy=76.61)

```
xgb_clf = xgb.XGBRegressor({'objective': 'reg:squarederror'})
parameters = {'n_estimators': [60, 100, 120, 140],
              'learning_rate': [0.01, 0.1],
              'max_depth': [5, 7],
              'reg_lambda': [0.5]}
xgb_reg = GridSearchCV(estimator=xgb_clf, param_grid=parameters,
cv=5, n_jobs=-1).fit(trainb, targetb)
print("Best score: %0.3f" % xgb_reg.best_score_)
print("Best parameters set:", xgb_reg.best_params_)
acc_boosting_model(7,xgb_reg,trainb,testb)
```

```
[22:37:00] WARNING: /workspace/src/objective/regression_obj.
cu:152: reg:linear is now deprecated in favor of reg:squared
error.
Best score: 0.844
```

XGBoost Algorithm (Accuracy=86.66%)

```
svr = SVR(C=100000)
svr.fit(X_train, y_train)
predictions = svr.predict(X_test)

mae, mse, rmse, r_squared = evaluation(y_test, predictions)
print("MAE:", mae)
print("MSE:", mse)
print("RMSE:", rmse)
print("R2 Score:", r_squared)
print("-"*30)
rmse_cross_val = rmse_cv(svr)
print("RMSE Cross-Validation:", rmse_cross_val)

new_row = {"Model": "SVR","MAE": mae, "MSE": mse, "RMSE": rmse,
"R2 Score": r_squared, "RMSE (Cross-Validation)": rmse_cross_va
l}
models = models.append(new_row, ignore_index=True)
```

SVM Algorithm (Accuracy=60.12%)

```
# Random Forest

#random_forest = GridSearchCV(estimator=RandomForestRegressor(),
param_grid={'n_estimators': [100, 1000]}, cv=5)
random_forest = RandomForestRegressor()
random_forest.fit(train, target)
print(random_forest.best_params_)
acc_model(6,random_forest,train,test)
```

Random Forest Algorithm(Accuracy=84.62%)

6.2.4 House Price Prediction

```
svr = SVR(C=100000)
svr.fit(X_train, y_train)
predictions = svr.predict(X_test)

mae, mse, rmse, r_squared = evaluation(y_test, predictions)
print("MAE:", mae)
print("MSE:", mse)
print("RMSE:", rmse)
print("R2 Score:", r_squared)
print("-"*30)
rmse_cross_val = rmse_cv(svr)
print("RMSE Cross-Validation:", rmse_cross_val)

new_row = {"Model": "SVR","MAE": mae, "MSE": mse, "RMSE": rmse,
"R2 Score": r_squared, "RMSE (Cross-Validation)": rmse_cross_va
l}
models = models.append(new_row, ignore_index=True)
```

SVM Algorithm(Accuracy=85.2)

```
random_forest = RandomForestRegressor(n_estimators=100)
random_forest.fit(X_train, y_train)
predictions = random_forest.predict(X_test)

mae, mse, rmse, r_squared = evaluation(y_test, predictions)
print("MAE:", mae)
print("MSE:", mse)
print("RMSE:", rmse)
print("R2 Score:", r_squared)
print("-"*30)
rmse_cross_val = rmse_cv(random_forest)
print("RMSE Cross-Validation:", rmse_cross_val)

new_row = {"Model": "RandomForestRegressor","MAE": mae, "MSE": m
se, "RMSE": rmse, "R2 Score": r_squared, "RMSE (Cross-Validatio
n)": rmse_cross_val}
models = models.append(new_row, ignore_index=True)
```

Random Forest Algorithm(Accuracy=86.9)

```
xgb = XGBRegressor(n_estimators=1000, learning_rate=0.01)
xgb.fit(X_train, y_train)
predictions = xgb.predict(X_test)

mae, mse, rmse, r_squared = evaluation(y_test, predictions)
print("MAE:", mae)
print("MSE:", mse)
print("RMSE:", rmse)
print("R2 Score:", r_squared)
print("-"*30)
rmse_cross_val = rmse_cv(xgb)
print("RMSE Cross-Validation:", rmse_cross_val)

new_row = {"Model": "XGBRegressor","MAE": mae, "MSE": mse, "RMS
E": rmse, "R2 Score": r_squared, "RMSE (Cross-Validation)": rmse
_cross_val}
models = models.append(new_row, ignore_index=True)
```

XGBoost Algorithm(Accuracy=90.65)

Chapter 7

Conclusion and Future Scope

7.1 Conclusion

XGBoost emerged as the strongest performer for house and car price prediction, achieving accuracy above 86%. It also showed good results for loan prediction (78.861%). Decision Tree and Random Forest were competitive alternatives, particularly for house prices (over 81%). SVM underperformed in most cases, especially for car prices (under 60% accuracy).

7.2 Future Scope

Once our model has been trained on a given set of data, it can now be used to make predictions on new sets of input data. The model has learned what the best questions to ask about the input data are and can respond with a prediction for the target variable. We can use these predictions to gain information about data where the value of the target variable is unknown — such as data the model was not trained on.

References

[1]."A Comparative Study of Machine Learning Algorithms for Credit Card Fraud Detection" by S. V. Nikam et al. (2019).

[2].Yousef, A.H.: Extracting software static defect models using data mining. Ain Shams Eng. J. 6(1), 133–14

[3].Bowes, D., Hall, T., Petrić, J.: Software defect prediction: do different classifiers find the same defects? Softw. Qual. J. 26(2), 525–552 (2018)

STUDENT'S CONTRIBUTION TO THE PROJECT

NAME OF STUDENT	Dhruv Kumar
ROLL NO	21052495
PROJECT TITLE	Comparative analysis of machine learning algorithm
ABSTRACT OF THE PROJECT(WITHIN 80 WORDS)	This research evaluates Random Forest, XGBoost, Decision Tree, and Support Vector Machine algorithms across various datasets from Kaggle. Through comprehensive benchmarking, it compares their performance in both regression and classification tasks, shedding light on their efficacy across diverse domains. Identifying key attributes in prediction models offers valuable insights for practitioners and researchers, aiding in algorithm selection for real-world applications. This study contributes significantly to the ongoing discourse on algorithm choice, facilitating informed decision-making.
CONTRIBUTION	
CONTRIBUTION TO THE PROJECT REPORT	Contributed to the report regarding project planning and implementation along with a screenshot of the project.
CONTRIBUTION DURING IMPLEMENTATION/PRESENTATION	Implemented Car price prediction and House price prediction model using 4 different machine learning algorithms using the dataset from Kaggle and also plotted a scatter graph on predicted price vs actual price. Also done various feature engineering and exploratory data analyses on the dataset to remove any outliers from the model.

STUDENT'S CONTRIBUTION TO THE PROJECT

NAME OF STUDENT	Apurva Jha
ROLL NO	21052483
PROJECT TITLE	Comparative analysis of machine learning algorithm
ABSTRACT OF THE PROJECT(WITHIN 80 WORDS)	This research evaluates Random Forest, XGBoost, Decision Tree, and Support Vector Machine algorithms across various datasets from Kaggle. Through comprehensive benchmarking, it compares their performance in both regression and classification tasks, shedding light on their efficacy across diverse domains. By identifying key attributes in prediction models, it offers valuable insights for practitioners and researchers, aiding in algorithm selection for real-world applications. This study contributes significantly to the ongoing discourse on algorithm choice, facilitating informed decision-making.
CONTRIBUTION	
CONTRIBUTION TO THE PROJECT REPORT	Contributed in the report regarding project planning and implementation along with a screenshot of the project.
CONTRIBUTION DURING IMPLEMENTATION/PRES -ENTATION	Contributed to implementing the Loan Prediction Model using four different Machine Learning Algorithms and a dataset from Kaggle. Trained and Processed the dataset to prevent missing values. Also, I analyzed the data using various feature engineering methods and exploratory analysis to remove the outliers from the model.

STUDENT'S CONTRIBUTION TO THE PROJECT

NAME OF STUDENT	Rishav Raj
ROLL NO	21051419
PROJECT TITLE	Comparative analysis of machine learning algorithm
ABSTRACT OF THE PROJECT(WITHIN 80 WORDS)	This exploration evaluates Random Forest, XGBoost, Decision Tree, and Support Vector Machine algorithms across colorful datasets from Kaggle. Through comprehensive benchmarking, it compares their performance in both retrogression and bracket tasks, slipping light on their efficacity across different disciplines. relating crucial attributes in vaticination models offers precious perceptivity for interpreters and experimenters, abetting in algorithm selection for real-world operations. This study contributes significantly to the ongoing converse on algorithm choice, easing informed decision-making.
CONTRIBUTION TO THE PROJECT REPORT	I laboriously shared in casting the design planning and perpetration report, furnishing precious perceptivity and benefactions. As part of my involvement, IPRES-ENTATION screenshot showcasing the design, perfecting the report with visual representation.
CONTRIBUTION DURING IMPLEMENTATION/PRES -SENTATION	Helped put the Loan Prediction Model into practice by employing a Kaggle dataset and four distinct machine learning algorithms. To avoid missing values, the dataset was processed and trained. To exclude the outliers from the model, I also conducted an exploratory study and used a variety of feature engineering techniques on the data.

STUDENT'S CONTRIBUTION TO THE PROJECT

NAME OF STUDENT	Aaryash Kumar
ROLL NO	21051871
PROJECT TITLE	Comparative analysis of machine learning algorithm
ABSTRACT OF THE PROJECT(WITHIN 80 WORDS)	This research assesses the performance of the Random Forest, XGBoost, Decision Tree, and Support Vector Machine algorithms on a variety of Kaggle datasets. It provides insight into their effectiveness across various domains by comparing their performance in regression and classification tasks through thorough benchmarking. It provides practitioners and researchers with insightful information by highlighting important characteristics in prediction models, which helps with algorithm selection for practical uses. This study facilitates informed decision-making by making a significant contribution to the ongoing discourse on algorithm choice.
CONTRIBUTION TO THE PROJECT REPORT	Contributed in the report regarding project planning and implementation along with a screenshot of the project.
CONTRIBUTION DURING IMPLEMENTATION/PRES -ESENTATION	Contributed to the implementation of the Stock Price Prediction Model with four different Machine Learning Algorithms and a Kaggle dataset. Trained and processed the dataset to eliminate missing values. In addition, I analyzed the data using various feature engineering methods and exploratory analysis to remove outliers from the model.

STUDENT'S CONTRIBUTION TO THE PROJECT

NAME OF STUDENT	A Subham Patro
ROLL NO	21051021
PROJECT TITLE	Comparative analysis of machine learning algorithm
ABSTRACT OF THE PROJECT(WITHIN 80 WORDS)	This research evaluates Random Forest, XGBoost, Decision Tree, and Support Vector Machine algorithms across various datasets from Kaggle. Through comprehensive benchmarking, it compares their performance in both regression and classification tasks, shedding light on their efficacy across diverse domains. By identifying key attributes in prediction models, it offers valuable insights for practitioners and researchers, aiding in algorithm selection for real-world applications. This study contributes significantly to the ongoing discourse on algorithm choice, facilitating informed decision-making.
CONTRIBUTION TO THE PROJECT REPORT	Contributed to the report regarding project planning and implementation along with a screenshot of the project. At the project's outset, I led SRS data collection and planning.
CONTRIBUTION DURING IMPLEMENTATION/PRES -ENTATION	During implementation, I swiftly resolved code errors, enhancing quality. In presentations, I effectively communicated challenges, and solutions, and improved visual representations, all contributing to our success.

Comparative Analysis of Machine Learning Algorithms

ORIGINALITY REPORT

19%	18%	5%	11%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	www.worldleadershipacademy.live Internet Source	3%
2	www.researchgate.net Internet Source	3%
3	www.coursehero.com Internet Source	2%
4	Submitted to KIIT University Student Paper	2%
5	www.kaggle.com Internet Source	1%
6	upcommons.upc.edu Internet Source	1%
7	Submitted to CSU, San Jose State University Student Paper	1%
8	fizzylogic.nl Internet Source	1%
9	link.springer.com Internet Source	1%

10	dspace.lib.uom.gr Internet Source	1 %
11	Submitted to Rowan University Student Paper	1 %
12	ebin.pub Internet Source	<1 %
13	Submitted to Laureate Higher Education Group Student Paper	<1 %
14	Submitted to University of Maryland, University College Student Paper	<1 %
15	www.jespublication.com Internet Source	<1 %
16	"Futuristic Trends in Networks and Computing Technologies", Springer Science and Business Media LLC, 2020 Publication	<1 %
17	Submitted to Liverpool John Moores University Student Paper	<1 %
18	dspace.chitkara.edu.in Internet Source	<1 %
19	edubirdie.com Internet Source	<1 %

20	"Advances in Cyber Security", Springer Science and Business Media LLC, 2021 Publication	<1%
21	kola.opus.hbz-nrw.de Internet Source	<1%
22	pdfcoffee.com Internet Source	<1%
23	www.mdpi.com Internet Source	<1%

Exclude quotes	Off	Exclude matches	Off
Exclude bibliography	Off		