COMPARING DIFFERENT ML TECHNIQUES USING THREE DIFFERENT DATASETS

SHREYANSH GUPTA

Abstract

In this paper, we have explored the applications of machine learning techniques in different fields through the analysis of three datasets. The datasets include the Palmer Archipelago penguin dataset, King County House Sales dataset and credit card default user dataset. The project outlines the KDD process, including data selection, cleaning, feature projection, and data mining algorithm selection.Decision tree and random forest algorithms are used for classification and regression analysis of the datasets. For each of it, the project discusses the steps involved in modeling, evaluating, and selecting the best-performing machine learning model. The project concludes with suggestions for future work, including improving model performance through hyperparameter tuning, cross-validation, and feature engineering. Overall, this project highlights the diverse applications of machine learning and the importance of careful data preparation and model evaluation in producing reliable and accurate results.

1 INTRODUCTION

A branch of artificial intelligence known as ”machine learning” is concerned with creating machines that can gather information from data and make judgments or predictions using that knowledge. It has emerged as one of the most sophisticated and popular solutions for a variety of issues, ranging from image and speech recognition to scam detection and self-driving cars. As a result, there is an increasing need for professionals who can effectively create and execute machine learning models. This has inspired many people to learn how to create machine learning models, including how to choose and evaluate training data.
For this project, we have selected three diverse datasets to explore and compare the effectiveness of two popular machine learning techniques, decision trees (DT) and random forest (RF)

1.1 Palmer Archipelago (Antarctica) penguin data :

This dataset [1] was obtained from Kaggle.The population statistics of three different species of penguins in the Antarctic region are included in the Palmer Archipelago penguin dataset, which is accessible on Kaggle. It has 344 rows and 17 columns, where each row corresponds to a particular penguin and each column to a different variable. Along with demographic information like the age and sex of the penguins, the dataset also includes various physical measurements like bill length, bill depth, and flipper length. This makes it an excellent dataset for implementing machine learning methods for classification tasks because it also specifies which species of penguin each individual belongs to.

Overall, to learn more about the biology and behavior of various penguin species, one can explore and analyze the Palmer Archipelago penguin dataset’s rich and diverse set of features.Overall, to learn more about the biology and behavior of various penguin species, we will explore and analyze the Palmer Archipelago penguin dataset’s rich and diverse set of features.

1.2 King county house sale prediction :

The King County House Sales dataset [2]is a compilation of information on homes sold between May 2014 and May 2015 in King County, Washington, in the United States. Each row in the dataset corresponds to a different home sale, and each column to a different property feature or attribute, such as the total number of bedrooms, bathrooms, and lot square footage. The dataset is made up of 21,614 rows and 21 columns. The dataset is helpful for implementing machine learning algorithms for regression analysis, such as predicting the sale price of a house based on its features, because it also contains information on each house’s sale price.

Overall, The King County House Sales dataset, as a whole, offers a rich and varied collection of features that can be studied and analyzed to learn more about the real estate market and to create efficient machine learning models for estimating house sale prices.

1.3 Credit card default prediction :

Credit card default is a frequent issue that both individuals and financial institutions deal with. This dataset [3] was obtained from Kaggle, a well-known website for data science contests and machine learning projects, in order to better comprehend and predict credit card defaults. Customers' credit card usage and payment habits in Taiwan from April 2005 to September 2005 are covered by the "Default of Credit Card Clients Dataset". A total of 30,000 samples make up the dataset, which has 23 features including demographic information, credit card usage patterns, and payment status. Researchers and data analysts frequently use this dataset to create models that can precisely predict credit card defaults and lessen financial losses.

In light of this, the dataset presents an invaluable resource for researching and delving into the causes of credit card defaults, as well as for creating and evaluating machine learning models that can enhance credit risk management.

2 Related work :

The following datasets have multiple articles published about them.

For Palmer Archipelago (Antarctica) penguin data This paper [4] "Data Exploration and Visualization using Palmer Penguins Dataset" is a tutorial on how to perform exploratory data analysis (EDA) on the Palmer Archipelago penguin dataset. The author uses Python and various data visualization libraries such as seaborn, matplotlib, and plotly to analyze the dataset and generate visualizations that help to understand the data better.

The article [4]begins by providing some background information on the Palmer Archipelago penguin dataset and its features. It then proceeds to describe the steps involved in EDA, including data cleaning, data wrangling, and data visualization. The author provides detailed explanations of the various data visualization techniques used in the tutorial, such as scatter plots, box plots, and histograms, and how they can be used to uncover patterns and relationships in the data. However, there are a few drawbacks that readers should be aware of. First off, the article only discusses fundamental EDA and visualization methods; it skips over more sophisticated data analysis procedures or machine learning strategies that could be used to glean additional insights from the dataset. Furthermore, the article lacks statistical analysis, which may restrict the range of inferences that can be made from the data. The article [4] does not offer interactive notebooks or code that readers can experiment with and modify, despite the fact that it does use interactive visualization libraries like plotly. Finally, the article does not offer a comparison with other studies that have looked into the biology and ecology of penguins using the Palmer Archipelago penguin dataset or other penguin datasets. Despite the fact that the article offers an informative start to the Palmer Archipelago penguin dataset, the audience should be aware of these drawbacks and think about conducting further research and analysis to glean more information from the data.

For king county house sale, Several machine learning algorithms have been applied to the King County housing dataset to forecast house resale prices. In one study [5], the algorithm with the highest accuracy was deemed the best among logistic regression, decision tree, Naive Bayes, random forest, and AdaBoost. In this study, Adaboost and decision tree C5.0 came out on top. This work's disadvantage was that cross-validation techniques were not used prior to model construction. Another study [6] compared the outcomes based on the root mean square error (RMSE) using linear models, tree-based models, deep learning models, and Catboost. Data prediction tests were conducted using Catboost, which displayed the lowest RMSE. In a different study,The following techniques were used: support vector machines, random forest, gradient boosting method, elastic net, and neural networks.. Despite the fact that the model was cross-validated, only 10% of the data was used to train the models, which could cause variation when

making predictions in the future. Another study looked at clustering using K-means, gradient tree boosting, stochastic dual gradient ascent, and stochastic gradient descent to predict real estate prices. The gradient tree boosting model delivered the most effective outcomes of all. In one study, the necessary features for prediction were chosen using VIF, information value computation, and principal component analysis (PCA). Although random forest performed better, SVM was chosen because random forest was prone to overfitting, even though ANN, SVM, and random forest algorithms were used.

For Credit card default, This goal of the study [7] and research is to categorize and forecast credit card default customers payments using a modern artificial neural network (ANN) technique known as deep neural network. It explains the dataset, which represents credit card defaults on Taiwanese accounts in 2005 and their prior payment histories taken from the well-known machine learning dataset source known as UCI.This paper [7] also clarifies every idea and procedure that must be followed in order to create, train, test, and validate a deep neural network model for a classification task.. Moreover, they tried to elaborate the relevant and important concepts associated with deep neural network model that must be kept in mind during model building.One limitation of the study [7] is that it is based on a single dataset, the Taiwan credit card defaults in 2005, which may not be representative of credit card defaults in other regions or time periods. Additionally, the study does not compare the performance of the deep neural network approach with other traditional machine learning algorithms, which could provide insights into the relative effectiveness of different approaches. Furthermore, the study does not provide an in-depth analysis of the features and their importance in predicting credit card defaults, which could limit the interpretability of the results.

3 METHODOLOGY

To extract knowledge from the data in this project, we used the KDD (Knowledge discovery in databases) [8] method.Following steps are involved :

A. Data selection
B. Data preparation and cleaning
C. Feature projection
D. Data mining algorithm selection

It is worth noting that the KDD process is not a rigid, linear process, and the steps involved may vary depending on the specific goals and nature of the analysis. Additionally, the quality of the final results is heavily dependent on the quality of the data used and the choices made at each step of the process. Therefore, it is crucial to carefully evaluate and validate each step to ensure the reliability and accuracy of the results.

3.1 Palmer Archipelago (Antarctica) penguin data -

3.1.1 Data Selection -

The penguin dataset [1] includes various physical measurements of penguins, such as their bill length, bill depth, flipper length, body mass, species, sex, and the island they were observed on. The dataset includes observations of penguins taken from 2 islands in the Palmer Archipelago, Antarctica, between 2007 and 2009. The target variable is not explicitly defined in this dataset, but it could be inferred as any of the physical measurements such as the body mass or flipper length.

3.1.2 Description of the dataset

The penguin dataset consists of 344 rows and 8 columns.

Data Type	Description
Species	The species of penguin (Adelie, Chinstrap, or Gentoo)
Island	The island where the penguin was observed (Biscoe or Dream)
Bill length mm	The length of the penguin's bill in millimeters
Bill depth mm	The depth of the penguin's bill in millimeters

Flipper _length mm	The length of the penguin's flipper in millimeters
Body mass g	The body mass of the penguin in grams
Sex	The sex of the penguin (male or female)
Year	The year the penguin was observed

Table 1: Description of the penguin dataset.

3.1.3 Data cleaning and Pre-processing -

The penguin dataset used has undergone some data cleaning and preprocessing steps. There were some missing values found when dataset was explored.As there were some null values in the dataset, they were removed using the .dropna() method. Then, the independent variable (X) was created by storing all columns except the last column, and the dependent variable (Y) was created by storing the last column which is 'sex'.

As there were some categorical string values in some columns, I converted it to numerical values in X. The columns containing categorical string values were identified and then each column was label encoded, Finally we use the information about the dataset including the number of non-null values, data types, etc.

3.1.4 Feature selection and Data visualisation -

Fig. 1 creates a Heatmap of the dataset using some resourceful variables. The heatmap shows the correlation between each pair of variables in the penguin dataset. The correlation values range from -1 to 1, with 1 indicating a perfect positive correlation and -1 indicating a perfect negative correlation.

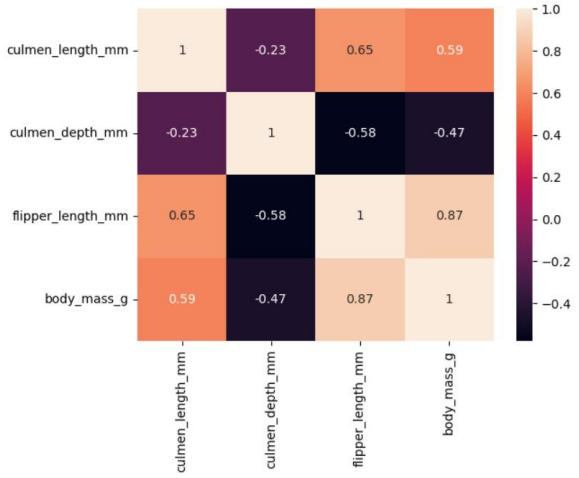


Figure 1: Heat map

Looking at the heatmap (fig. 1), we can see that there are some variables that are strongly correlated with each other. For example, culmen length and body mass have a strong positive correlation (0.59), which means that penguins with longer culmens tend to have a higher body mass. Flipper length and body mass also have a strong positive correlation (0.87), which means that penguins with longer flippers tend to have a higher body mass.

On the other hand, culmen depth and body mass have a weak negative correlation (-0.47), which means that there is some relationship between them, but it is not as strong as the other correlations mentioned above.Similarly, there is a weak negative correlation (-0.23) between culmen length and culmen depth, which means that penguins with longer bills tend to have shallower bills. Overall, the heat map provides a useful visualization of the correlations between variables in the dataset and can help in identifying which variables might be useful for predicting the target variable.

Next, A pair plot (Fig. 2) was plotted to further visualize the relationship between different variables (columns) of the data set. It allows us to see the scatter plot of all the variables against each other, as well

as the histogram of each variable on the diagonal. Additionally, by setting the ”hue” parameter to ”species”, we can see how the variables are distributed across the three different penguin species.

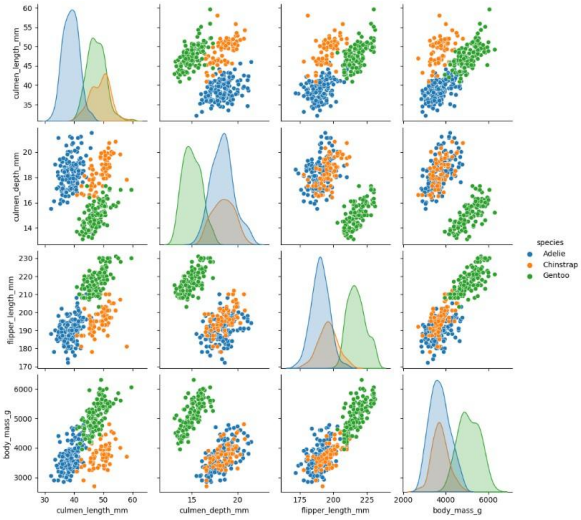


Figure 2: Pair Plot

From the pair plot, (Fig. 2) we can observe some interesting patterns and correlations between the different variables, We can see that different species of penguins have distinct characteristics when it comes to the size of their body parts such as bill length, bill depth, flipper length, and body mass. The Adelie and Chinstrap species appear to have more similarities in size, while the Gentoo species stands out with larger bill and body measurements. The pair plot also shows the distributions of each variable and any potential outliers that may need to be addressed in data cleaning or modeling.

we can also observe some interesting patterns and correlations between the different variables. For instance, there is a strong positive correlation between flipper length and body mass, which is expected since larger penguins tend to have larger flippers. We can also observe that Adelie and Gentoo penguins tend to have shorter and deeper bills compared to Chinstrap penguin.

Overall, the pair plot helped providing insights into which variables may have the most significant impact on predicting the species of a penguin based on its physical attributes.

3.1.5 MODEL BUILDING -

For the penguin dataset, we cleaned the data by dropping all null values and converting string values to float using label encoding. We then split the dataset into training and testing subsets with a 70:30 ratio. We used various machine learning algorithms such as Decision Tree and Random Forest to build our models. Since this is a classification problem, we evaluated the models based on accuracy, precision, recall, and F1 score. The model with the best performance was selected as the final model for predicting the sex of penguins based on their physical attributes.

DECISION TREE:

The decision tree [9]is a machine learning algorithm used for both regression and classification problems. It operates by recursively dividing the feature space into smaller areas in accordance with a set of predetermined decision rules. In the implementation of decision trees using the scikit-learn library, a node in the decision tree represents a decision rule, and a split in the tree represents a decision boundary. Scikit-learn’s decision tree algorithm can handle both numerical and categorical data and can handle missing values as well.

With respect to the penguin dataset, the decision tree algorithm can be used to determine a penguin’s sex based on physical traits like body mass, species, island, and year as well as traits like bill length, bill depth, and flipper length. Based on the values of these features and how they relate to the target variable, sex, the algorithm would construct a decision tree.

The outcome of the decision tree would be made up of a number of nodes, each of which would represent a choice based on a feature. The tree would be trained using a subset of the data, and its performance would then be assessed using a holdout set. Metrics like accuracy, precision, recall, and F1 score would be used to assess the decision tree's performance.

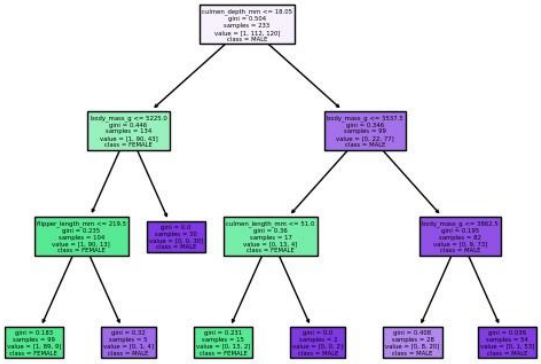


Figure 3: Decision Tree

Fig.3 shows the visual representation of the decision tree. It displays the decision-making process of the model in a hierarchical tree structure, where each internal node represents a decision based on one of the input features, and each leaf node represents a predicted output.

In this particular case, the decision tree is built to predict the sex of penguins based on their physical attributes such as culmen length, culmen depth, flipper length, and body mass. The tree starts with the root node at the top and then branches out into child nodes based on the decision rules.

The decision tree's attributes, such as the split criterion, feature importance, and number of samples, are displayed for each node in a graphical representation of the decision tree created by the plot tree function used in the code. Each node's color represents the predicted class, and the color's intensity shows the percentage of samples that fall into that class. Overall, the decision-making process of the model is effectively visualized in the final decision tree Fig.3, which facilitates comprehension and interpretation of the model behavior.

Random Forest (Sklearn Library):

Random Forest [10] was utilized in the penguin dataset to predict the species of penguins based on their physical characteristics. Initially, all the 2 predictors, culmen length and culmen depth, were taken as features at each split. The model was then run against the test data to assess the quality of the prediction. The Random Forest algorithm, implemented using the 'RandomForestClassifier' function from the 'sklearn' library, was used to create a model with hundred trees. The model's quality was evaluated using a confusion matrix and a classification report. However, the confusion matrix revealed that the model struggled to distinguish between the Adelie and Chinstrap species, as they share similar physical characteristics. In summary, the implementation of Random Forest using 'sklearn' in the penguin dataset code provided above successfully predicted the species of penguins with high accuracy, and the library made it easier to create and evaluate the model.

Visualisation -

Fig.4 shows a scatterplot of the penguin dataset with the 'culmen length mm' on the x-axis and 'culmen depth mm' on the y-axis. The points are colored based on the species of the penguin (Adelie, Chinstrap, and Gentoo), with Adelie in red, Chinstrap in blue, and Gentoo in black. The plot gives an

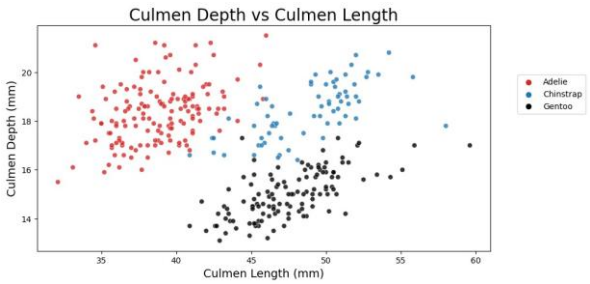


Figure 4: Scatter plot

idea about how these two features are distributed among the three species of penguins. It seems that culmen length and culmen depth are two important features that can be used to differentiate between the penguin species, as they appear to cluster differently for each species. The legend in the plot gives a clear indication of which species each color represents.

The scatter plot also reveals that certain values of culmen length and depth are indicative of a particular penguin species. This information can be used to train a machine learning algorithm, In our case, random forest to predict the species of a penguin based on its culmen dimensions. We must first divide our data into a training set and a test set so as to build the random forest algorithm. To generate return predictions for a specific target variable, in this case species, we will need a number of features, in this case culmen length and depth.

We import train test split from sklearn to carry out this split. Since the entire dataset contains 344 rows, I have selected a test size of 0.33, which means that 1/3 of the penguin dataset will be used to generate final predictions and 2/3 of the dataset will be used to train the random forest algorithm. The data is shuffled before the split is applied, with random state controlling how much. X and y are the features and target values, respectively.

Next, we will define the required variables and import the Random Forest Classifier from Sklearn. Additionally, there will be 100 trees in this forest because I used the standard value of 100 for n_estimators. I also set max depth to None, which allows each tree to expand as much as it needs to in order to divide the data.

Evaluating performance -

We need to assess the performance of our random forest algorithm after training it to identify the species of penguins based on the length and depth of their culmen. Tools for evaluating a classification model's effectiveness include a classification report and a confusion matrix. For each class in our data, the classification report gives us several metrics, including precision, recall, and f1-score. The f1-score is the harmonic mean of precision and recall. Precision measures the proportion of true positives among all the positive predictions generated by our model, recall evaluates the proportion of true positives among all the actual positives in our data. We can assess how well our model is working for each species of penguin by examining these metrics for each class.

On the other hand, the confusion matrix gives us a table that lists how many predictions our model made correctly and incorrectly. It shows the number of true positives, true negatives, false positives, and false negatives for each class in our data. The confusion matrix can be used to determine statistics like accuracy, precision, and recall to get a more accurate assessment of how well our model is doing overall.

The classification report and confusion matrix are important tools for evaluating the performance of a machine learning model. In the case of the random forest algorithm we trained on the penguin dataset, the Fig. 5 confusion matrix tells us how many penguins were correctly and incorrectly classified based on their culmen length and depth. From the Fig. 5 confusion matrix, we can see that the model made 7 incorrect predictions out of the 114 penguins in the test data. Also, the classification report provides a more detailed evaluation of the model's performance. It includes metrics such as precision, recall, and F1-score for each class, as well as an overall accuracy score. In our case, the classification report shows that

the random forest algorithm achieved an accuracy of 94%. This means that 94% of the penguins in the test data were correctly classified based on their culmen length and depth.

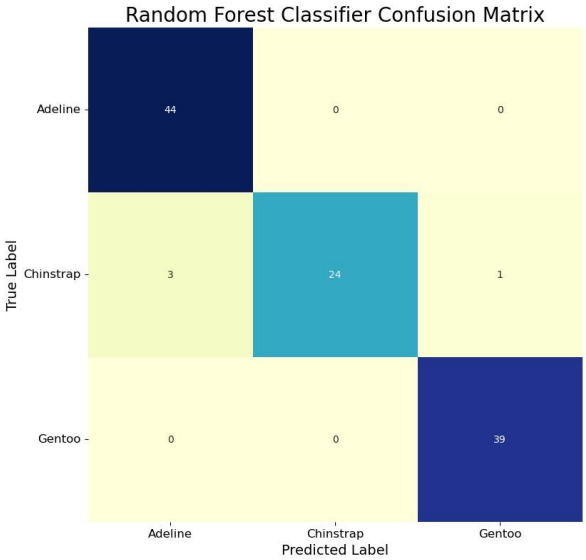


Figure 5: Confusion Matrix

Random Forest (Lightgbm Library): In this case, we train a random forest classifier using LightGBM library on the penguin dataset. The model is trained with the given initial parameters, and it is evaluated using the test data. The classification report provides useful information about the precision, recall, and F1-score of the model on each class, as well as the overall accuracy and macro-average metrics.

Based on the classification report, the model has achieved an accuracy of 88% on the test data, which is quite low when compared to the model that was built using the Sklearn library. The precision, recall, and F1-score of the model for each class also indicates that the model is able to make less accurate predictions for each species of penguin. Overall, this random forest classifier is also a good choice for predicting the species of penguins based on the length and depth of their culmen.

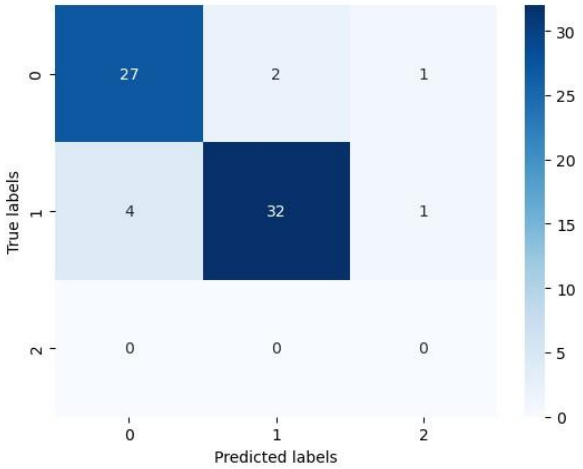


Figure 6: Confusion Matrix

The Fig. 6 confusion matrix shows the number of true positives, true negatives, false positives, and false negatives for each class in the test data. The rows correspond to the true labels, and the columns correspond to the predicted labels. In this case, there are 3 classes: Adelie, Chinstrap, and Gentoo.

The Fig. 6 confusion matrix shows that there are 27 true positives (correctly predicted Adelie penguins), 32 true positives (correctly predicted Chinstrap penguins), and 0 true positives (correctly

predicted Gentoo penguins). There are 2 false positives (Chinstrap penguins predicted as Adelie), 1 false positive (Gentoo penguin predicted as Adelie), and 1 false negative (Adelie penguin predicted as Chinstrap). There are no true negatives in this case since there is no fourth class in the data.

3.1.6 Interpreting mined result :

ALGORITHM	ACCURACY
DECISION TREE	0.87
RANDOM FOREST (Sklearn)	0.96
RANDOM FOREST (Lightgbm)	0.88

Table 2: Comparison of mined result.

The table 2 above shows the accuracy of three different algorithms on the penguin dataset. The decision tree algorithm achieved an accuracy of 0.87, while the random forest algorithm from the Sklearn library achieved the highest accuracy of 0.96. The random forest algorithm from the Lightgbm library achieved an accuracy of 0.88, which is lower than the Sklearn version but still higher than the decision tree algorithm. Overall, the random forest algorithm from the Sklearn library is the most accurate on this dataset.

3.2 King county house price sale -

3.2.1 Data Selection :

A well-known dataset from Kaggle, King County House Sales [2] provides comprehensive data on house sale prices and associated attributes for the months of May 2014 and May 2015 in King County, USA. This dataset, which has 21613 rows and 21 columns overall, contains a wealth of data that can be used to understand the King County real estate market. In this report, we will examine the dataset by examining a number of features, including the house's location, the state of the land, and the year it was built.

Column Name	Description
Id	A notation for a house
Date	Date house was sold
Price	Price is prediction target
Bedrooms	Number of Bedrooms/House
Bathrooms	Number of bathrooms/House
Sqft_Living	Square footage of the home
Sqft_Lot	Square footage of the lot
Floors	Total floors (levels) in house
Waterfront	House which has a view to a waterfront
View	Has been viewed
Condition	How good the condition is (Overall)
Grade	Overall grade given to the housing unit, based on King County grading system
Sqft_Above	Square footage of house apart from basement
Sqft_Basement	Square footage of the basement
Yr_Built	Built Year
Yr_Renovated	Year when house was renovated
Zipcode	Zip
Lat	Latitude coordinate
Long	Longitude coordinate
Sqft_Living15	Living room area in 2015 (implies– some renovations) This might or might not have affected the lotsize area
Sqft_Lot15	Lot size area in 2015 (implies– some renovations)

Table 3: Description of the King County House Sales dataset.

Figure 7: Description of the dataset

In addition, using various libraries, we will build a decision tree and random forest to forecast the sale price of a home in King County. We will visualize the data in addition to building a regression model to better understand the various factors that influence the sale price of a home in King County. We will conduct exploratory data analysis and produce visualizations that aid in our understanding of the data using a variety of Python libraries, including Pandas, Matplotlib, and Seaborn. Finally, we will assess the effectiveness of our models using a variety of metrics, including precision and accuracy. The overall goal of this report is to offer an in-depth examination of the King County House Sales dataset along with insights into the variables affecting local home sale prices.

3.2.2 Data cleaning and Pre-processing -

The initial stage of data cleaning involves verifying the presence of null values. Looking at the dataset info it is clear that the dataset have no null values present, so we checked for unique values for each feature. We looped through the dataset for checking the unique values

Also, I do not require the column id and date at this point so I dropped them from the dataset.

3.2.3 Feature selection and Data visualisation -

So, all the column data remains the same except the two column is dropped. Let's now forward with data visualization using a pairplot.

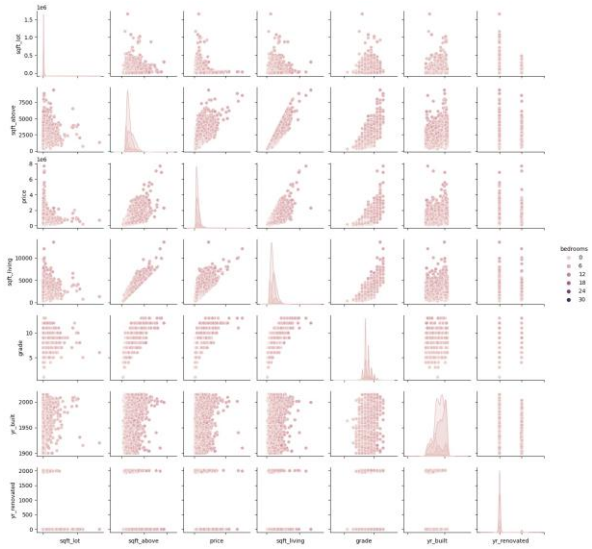


Figure 8: Pair plot

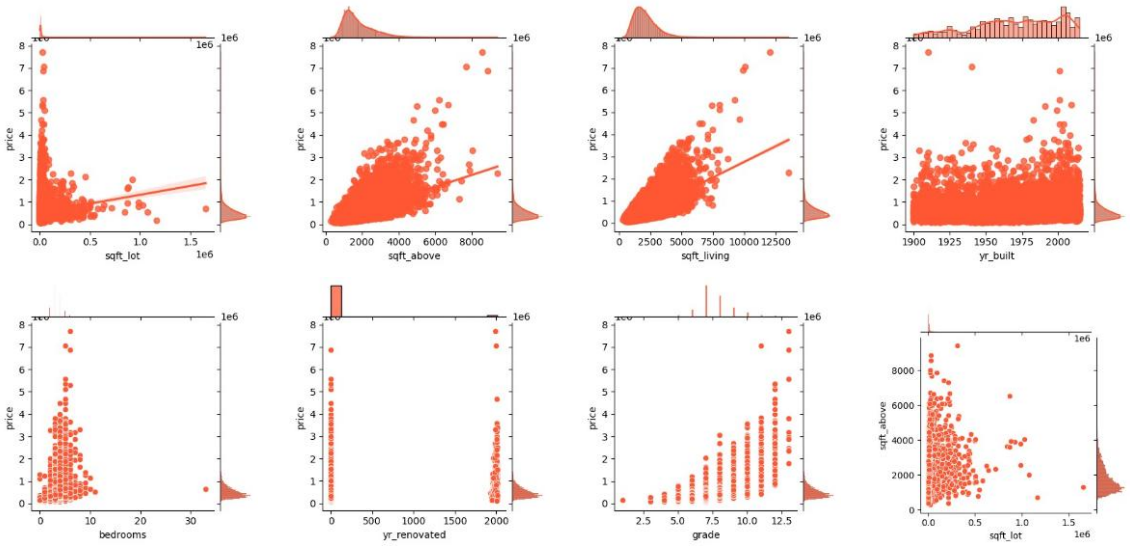


Figure 9: Joint Plot

The Fig. 8 pairplot help us to visually identify patterns, correlations, and potential outliers in the data. In our case, the scatterplots between sqft living and price show a positive linear relationship, suggesting that as the square footage of the living area of a house increases, its price also tends to increase. Additionally, we can see that the majority of the houses in the dataset have 1-4 bedrooms, with fewer houses having 5-6 bedrooms. The plot between yr built and price shows a relatively weak positive relationship, which suggests that the year a house was built may not be a strong predictor of its price.

Overall, this plot can provide useful insights into the relationships between variables in the dataset.To investigate this relationship further, A joint plot is created to visualize the data more clearly.

We will now plot a heatmap to further investigate the co-relation between variables.

Figure 10: Heatmap

From the Fig. 10 heatmap, we can see that 'sqft living' has a strong positive correlation with 'price' and 'grade', and a moderate positive correlation with 'bathrooms'. 'Bedrooms' and 'floors' have a weak positive correlation with 'price'. 'yr built' and 'condition' have a weak negative correlation with 'price'.

3.2.4 Model Building -

We made the assumption that the id and date column does not provide much information in analyzing house price throughly examining the dataset and correlation matrix, so it was eliminated from the model building process. Dataset was split into a 70:30 train and test subset.

Decision Tree:

The decision tree [9] regression model is trained and tested on the given dataset, which contains information about housing prices in King County, Washington. The model is evaluated using the score metric and the explained variance score metric. The decision tree regressor is created with a random state of 0, which ensures that the results can be reproduced. The model is then trained on the training dataset and used to predict the target variable on the test dataset. The model score is found to be 75, which means that the model explains 75% of the variance in the target variable.

Random Forest:

The code above is fitting a Random Forest Regression model [11] using the RandomForestRegressor class from the scikit-learn library. The model is initialized with 28 decision trees and a random state of 0. It is then fitted to the training data and then The model's performance is evaluated on the test data and the predictions are generated.

The output of the model score is 88, where the model score is calculated as the R-squared score which represents the proportion of variance in the dependent variable that is explained by the independent variables in the model.

Light gradient boosting: I have performed a regression analysis on the King County housing dataset using LightGBM, a gradient boosting framework.A LightGBM dataset is created using the training data, and the parameters for the model are set. The boosting type is set to 'rf', which stands for random forest, and the objective is set to regression. The metric for evaluation is set to RMSE (root mean squared error). Other parameters such as the number of leaves, learning rate, and feature and bagging fractions are also specified.The LightGBM model is then trained using the training set and the specified parameters. The model is used to predict on the test set, and the R-squared score is calculated.

The R-squared score is found out to be 0.7849, which gives an indication of how well the model is fitting the data.

Regression model using Light gradient boosting:

In this model, we fit a multiple linear regression model using the OLS method from the statsmodels library.we begin by loading the ”kc house data.csv” dataset and dropping the ”id” and ”date” columns. We then split the data into training and testing sets using a test size of 0.3 and a random state of 0. To cross check, We perform several tests on the model to evaluate its performance, including the NCv test, Durbin-Watson test, and VIF test.

F-static	1.020
P-Value	0.187
Durbin-Watson test statistic	1.997
Model R-squared score	0.897

Table 3: Test results.

After performing these tests, we move on to building a LightGBM regression model. We create a LightGBM dataset using the training data and set up the parameters for the model. We then train the model and make predictions on the test set. Finally, we calculate the R-squared score for the LightGBM model and plot the predicted vs actual prices using matplotlib.

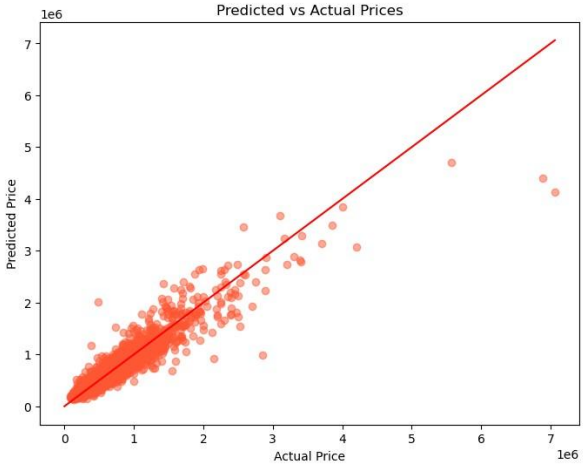


Figure 11: Predicted vs Actual prices

The Fig. 11 shows a scatter plot of the predicted prices on the y-axis against the actual prices on the x-axis. Each point on the plot represents a single data point in the test set. The red line in the plot represents the line of perfect predictions, where predicted prices are equal to the actual prices.

In this case, the Fig. 11 scatter plot shows that there is some deviation from the line of perfect predictions, but overall the predicted prices are fairly close to the actual prices. The points on the plot are clustered fairly tightly around the red line, indicating that the model is making reasonably accurate predictions. However, there are some outliers, particularly at the higher end of the price range, which suggests that the model may be overestimating prices for some of the more expensive homes.

3.2.5 Interpreting mined result :

The Table 5 compares the model scores produced by various algorithms that were used to create regression models for a specific dataset. Decision Tree, Random Forest, Regression model using Lightgbm, and RF Gradient Boosting are four algorithms that are contrasted. The model scores are expressed as a percentage value, which shows how well the algorithm performed in predicting the intended variable. The Lightgbm algorithm performed the best, as shown by the results in the table, obtaining a model score

of 89%, which was higher than the scores attained by the other algorithms. With a second-best score of 88%, the Random Forest algorithm proved to be a useful tool for creating regression models.

The RF Gradient Boosting algorithm scored 78%, while the Decision Tree algorithm scored 75%, indicating that these algorithms might not be the best ones to use when creating regression models for this dataset.

ALGORITHM	Model score
DECISION TREE	75
RANDOM FOREST	88
Regression model (Ligthgbm)	89
RF Gradient boosting	78

Table 4: Comparison of mined result.

In conclusion, the Lightgbm algorithm appears to be the most effective algorithm for creating regression models for the provided dataset, followed by the Random Forest algorithm, based on the comparison of the model scores. It is crucial to remember that additional research may be necessary to fully comprehend the advantages and disadvantages of each algorithm in relation to the unique features of the dataset.

3.3 Credit card default prediction-

3.3.1 Data Selection :

Data on Taiwanese credit card [3] users was gathered for the dataset between April 2005 and September 2005. It has 25 columns and 30,000 rows and 'default.payment.next.month' is the dataset's target variable.

Variable	Description
D	ID of each client
LIMIT$_BAL$	Amount of given credit in NT dollars
SEX	Gender (1=male, 2=female)
EDUCATION	Education level (1=graduate school, 2=university, 3=high school, 4=others, 5=unknown, 6=unknown)
MARRIAGE	Marital status (1=married, 2=single, 3=others)
AGE	Age in years
PAY_0	Repayment status in September, 2005
PAY_2	Repayment status in August, 2005 (scale same as above)
PAY_3	Repayment status in July, 2005 (scale same as above)
PAY_4	Repayment status in June, 2005 (scale same as above)
PAY_5	Repayment status in May, 2005 (scale same as above)
PAY_6	Repayment status in April, 2005 (scale same as above)
BILL_AMT1	Amount of bill statement in September, 2005 (NT dollar)
BILL_AMT2	Amount of bill statement in August, 2005 (NT dollar)
BILL_AMT3	Amount of bill statement in July, 2005 (NT dollar)
BILL_AMT4	Amount of bill statement in June, 2005 (NT dollar)
BILL_AMT5	Amount of bill statement in May, 2005 (NT dollar)
BILL_AMT6	Amount of bill statement in April, 2005 (NT dollar)
PAY_AMT1	Amount of previous payment in September, 2005 (NT dollar)
PAY_AMT2	Amount of previous payment in August, 2005 (NT dollar)
PAY_AMT3	Amount of previous payment in July, 2005 (NT dollar)
PAY_AMT4	Amount of previous payment in June, 2005 (NT dollar)
PAY_AMT5	Amount of previous payment in May, 2005 (NT dollar)
PAY_AMT6	Amount of previous payment in April, 2005 (NT dollar)
default_payment_next_month	Default payment (1=yes, 0=no)

Table 6: Description of the variables in the dataset.

Figure 12: Description of the dataset

3.3.2 Data cleaning and Pre-processing -

After analyzing the dataset, It was observed that there were no null values present. However, we there could be outliers present in the dataset that we must take care of.

By capping the extreme values at each end of the distribution, outliers were detected and dealt with. In order to effectively remove the outliers from the dataset, the resulting dataframe cred only contains values that fall between the first and 99th percentiles for each column. This is demonstrated by using a

heatmap (Fig. 13) to visualize the correlation matrix of ”cred,” in which the extreme values that were found prior to the application of the outlier treatment are no longer detectable.

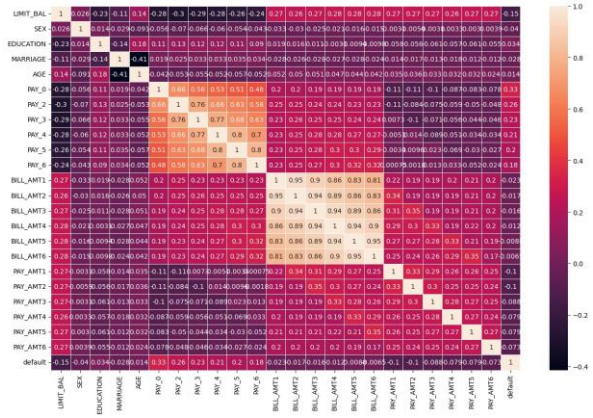
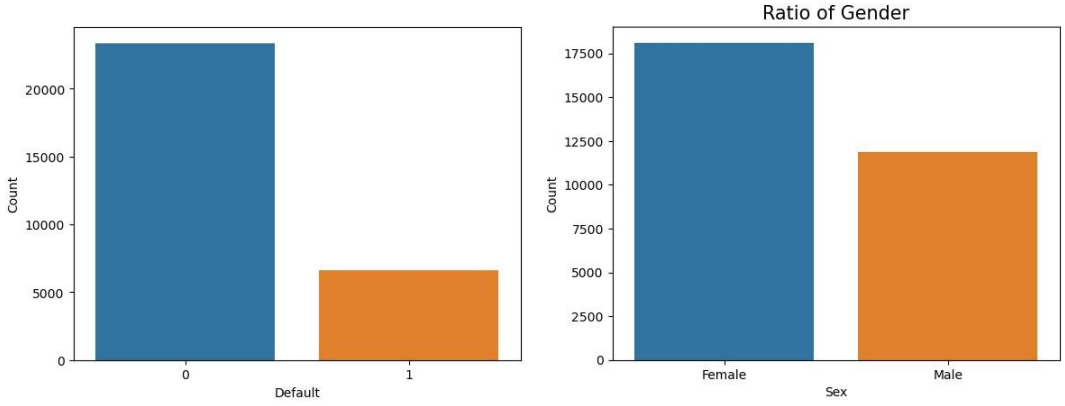


Figure 13: corelation heatmap

3.3.3 Feature selection and Data visualisation -

When the target variable (Fig. 14(a))’s bar graph was plotted, it was discovered that 22% of the sample was in the default class, making the data imbalance insignificant. We discovered by plotting the correlation matrix that the feature variables ’id’,’sex’, and’marriage’ are not particularly important in predicting the target variable, and as a result, they were eliminated from the model-building process.



(a) Count of default payment (b) Gender distribution of default payment

Figure 14: Default payment analysis

Fig. 14(b) demonstrates that women default more frequently than men.

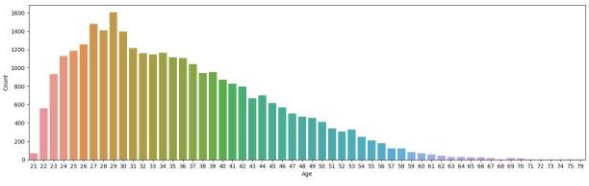


Figure 15: Barplot showing distribution of the age

Furthermore, The age variable’s distribution in the dataset was depicted using a bar plot Fig. 15. We can determine the proportion of people who fall into each age category by counting the frequency of each

distinct age value and plotting it as a bar graph. With the aid of this visualization, we can better comprehend the population's age distribution and spot any trends or irregularities in the data.

The overall distribution of ages in the dataset is displayed in a bar plot (Fig. 15). Age is represented on the x-axis, and the number of individuals in the dataset who match that age is shown on the yaxis. The total number of people in each bar corresponds to that age.The age range with the greatest number of people is between 20 and 30 years old, as can be seen from the plot. The count then declines with age, with the lowest possible count occurring in the group of people over 70. Overall, the plot indicates that there are fewer individuals in the older age groups and that the majority of those in the dataset are younger than average.

3.3.4 Model Building -

Decision Tree:

I performed hyperparameter tuning using a grid search approach for a decision tree classifier model [9]. Hyperparameters are parameters that are set prior to training the model and can have a significant impact on the performance of the model.The grid search is performed using a 5-fold cross-validation scheme. The best hyperparameters are then used to instantiate a new decision tree classifier. Next, the decision tree classifier is trained on the training set, and predictions are generated for both the training and testing sets. The classification report and accuracy scores are then printed for both the training and testing sets.

The model has two classes, 0 for clients who did not default and 1 for clients who did default. The classification report shows the precision, recall, and f1-score for both classes, as well as the support (number of samples) for each class. The precision for class 0 is 0.81, which means that 81% of the samples predicted as class 0 are actually class 0. The recall for class 0 is 0.96, which means that 96% of the actual class 0 samples are correctly predicted as class 0. The f1-score for class 0 is 0.88, which is the harmonic mean of precision and recall. The precision for class 1 is 0.63, which indicates that 63% of the samples that were predicted to be in class 1 are in fact in class 1. Only 23% of the actual class 1 samples were correctly predicted as being in class 1 according to the recall for class 1, which is 0.23. For class 1, the f1-score is 0.34. Class 1 has a support of 4688.

The model correctly identified the class for 79.8% of the samples in the training set, according to the accuracy score for the training set of 0.798. The model correctly identified the class for 79.7% of the samples in the testing set, according to the accuracy score for the testing set of 0.797.

In conclusion, the decision tree model exhibits excellent precision and recall for class 0, showing that it can accurately predict non-default clients. Its low recall and class 1 f1-score, however, show that it has trouble predicting default clients. As both the training and testing sets' accuracy scores are comparable, the model is not overfit to the training data, according to this information.

Light Gradient boosting: A decision tree model was created using LightGBM, a fast and efficient gradient boosting framework. Hyperparameters were tuned using GridSearchCV to find the optimal values for max depth and max _features. The final model had a max depth of 3 and max features of 4 with a criterion of 'gini'. The model's performance was evaluated using classification report and accuracy score on both the training and testing sets. The model achieved an accuracy score of 0.821 on the training set and 0.817 on the testing set, indicating that the model is performing well and is not overfitting the training data.

The classification report showed that the model had an F1-score of 0.50 for predicting defaults and 0.91 for predicting non-defaults on the testing set. The model's recall was lower for defaults (0.37) than for non-defaults (0.96), indicating that the model is better at predicting non-defaults than defaults. The precision of the model was also higher for non-defaults (0.85) than for defaults (0.35), indicating that the model has a higher false positive rate for defaults.

Overall, the decision tree model created using LightGBM was able to achieve good performance on the UCI Credit Card Default Dataset, with an accuracy score of 0.817 on the testing set. However, the model's performance could be further improved by collecting more data, feature engineering, or trying other machine learning algorithms.

Random Forest: For this model, the data is split into training and testing sets, A Random Forest Classifier [10]is then instantiated. The model is trained on the training set to make predictions for both the training and testing sets.

The precision for the default=0 (non-default) class for the training dataset is 0.86, and the precision for the default=1 (default) class is 0.85. Recall for the default=0 class is 0.98, indicating that most non-default cases can be correctly identified by the model. The model is not doing as well at recognizing default cases, as evidenced by the default=1 class's recall of 0.45. For the default=0 class and the default=1 class, respectively, the F1-score is 0.92 and 0.59. The precision for the default=0 class and the default=1 class, respectively, for the testing dataset is 0.82 and 0.62 respectively. Recall for the default=0 class is 0.95, which shows that most non-default cases can be correctly identified by the model. The model is not doing as well at identifying default cases, as evidenced by the default=1 class's recall of 0.28. The default=0 class's F1-score is 0.88, while the default=1 class's is 0.39.

The model is capable of correctly categorizing 86% of the cases in the training dataset, according to the accuracy score for the training dataset, which is 0.86. The model can correctly identify 81% of the cases in the testing dataset, according to the accuracy score for the testing dataset, which is 0.81.

In general, it seems that the model does a better job recognizing non-default cases than default cases. Given the low recall for the default=1 class, it is likely that the model is missing many instances of default. This might mean that further feature engineering or model tuning is required to enhance the model's performance.

Random Forest (Light gradient boosting):

In this model, he dataset is split into training and testing sets, and the model is trained using the hyperparameters specified in the params dictionary. The num leaves, learning rate, feature fraction, bagging fraction, and bagging freq parameters are all tuned to optimize the model's performance.The model is then used to predict on both the training and testing datasets, and the resulting probabilities are converted to binary predictions using a threshold of 0.5

The classification report shows the precision, recall, and F1-score for each class, as well as the accuracy, for both the training and testing datasets.For the training dataset, the model achieved an accuracy score of 0.833, indicating that it correctly classified 83.3% of the samples. The precision for the negative class (0) was high at 0.85, indicating that when the model predicted a sample was negative, it was correct 85% of the time. For the testing dataset, the model achieved an accuracy score of 0.821, which is slightly lower than the training accuracy. The precision for the negative class was still high at 0.85, but the precision for the positive class was lower at 0.65.Overall, the model appears to perform well on the negative class, but struggles to correctly identify positive samples, particularly in terms of recall. This could be an indication of class imbalance or other issues with the dataset, and may require further investigation.

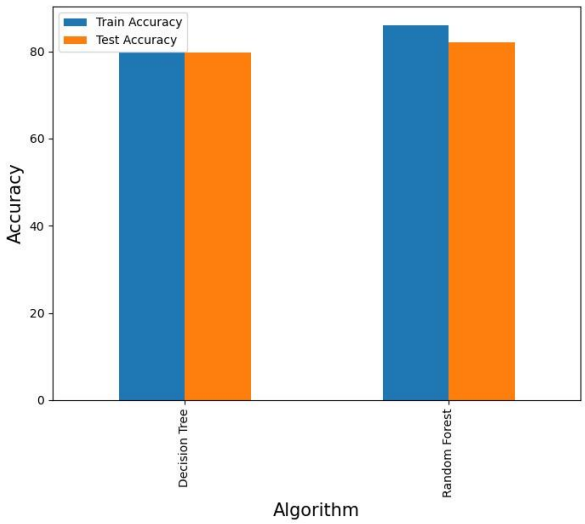


Figure 16: Comaparison of Decision tree and random Forest

3.3.5 Interpreting mined result :

The provided barplot Fig. 16 compares the performance of Decision Tree and Random Forest classification algorithms on the training and testing datasets.The accuracy score is represented by the y-axis, and the two algorithms are represented by the x-axis. The accuracy scores for the training dataset are shown in blue, and the accuracy scores for the testing dataset are shown in orange.It is clear from the plot that both algorithms perform better on the training dataset than the testing dataset. This indicates that the models might be poorly generalizing to new data and overfitting the training set.

The Random Forest algorithm has higher scores for accuracy on both the training and testing datasets, and it also seems to function better than the Decision Tree algorithm.

ALGORITHM	Accuracy
DECISION TREE	0.79
RANDOM FOREST	0.85
Decision tree (Ligthgbm)	0.82
RF Gradient boosting	0.83

Table 5: Comparison of mined result.

The accuracy of all four distinct methods is compared in the table. Decision tree and random forest, two traditional machine learning algorithms, make up the first two algorithms. The third and fourth algorithms, which employ the LightGBM and RF frameworks, are variations of gradient boosting. The findings show that the random forest algorithm, with an accuracy of 0.85, is the most accurate, closely followed by the RF gradient boosting algorithm, with an accuracy of 0.83. The LightGBM gradient boosting algorithm has an accuracy of 0.82, while the decision tree algorithm has the lowest accuracy at 0.79.

In summary, the data in the table indicates that ensemble techniques, like gradient boosting and random forest, can outperform standalone decision tree models. The effectiveness of the model can also be influenced by the framework for gradient boosting that is selected.

4 Conclusion and Future Work :

Overall, we have experimented with a number of machine learning algorithms.We used hyperparameter tuning strategies to enhance the efficiency of Decision Tree, Random Forest, and LightGBM. The maximum tree depth, the minimum number of samples needed at a leaf node, the variety of trees in the forest, and the learning pace of the LightGBM algorithm were all adjusted. To identify the best hyperparameters for each algorithm, we also used methods like grid search and random search. To make sure that our models weren't overfitting on the training data, we also used cross-validation. We were able to improve accuracy and lower the danger of overfitting in our models by adjusting the hyperparameters and utilizing cross-validation.

4.1 Data set I :

According to the study of the penguin dataset, there are correlations between the physical characteristics of penguins, including their body mass, bill length, bill depth, and flipper length. While a few of these correlations are strong, like the one between flipper length and body mass, others are weak, like the one between culmen depth and body mass. The pair plot visualization also demonstrated that various penguin species have unique physical characteristics. Different machine learning algorithms, including decision trees and random forests, were used to model the dataset and achieved 0.87 and 0.96 accuracy respectively, and the model that performed the best was chosen based on evaluation metrics like accuracy, precision, recall, and F1 score. Based on physical characteristics like body mass, species, island, and year as well as characteristics like culmen length, culmen depth, and flipper length, the final model can determine the sex of penguins.

In summary, the analysis of the penguin dataset provides insights into the relationships between different physical measurements of penguins and how they vary across different penguin species. It also

demonstrates the use of machine learning algorithms for modeling and predicting the sex of penguins based on their physical attributes.

Future work : The penguin dataset has, overall, offered insightful information about the habits and trends in the population of these exciting birds. Future research, could be useful in a number of areas. The fluctuations in penguin populations may be better understood with the addition of data, especially data collected over a longer period or in different locations. Also, research into how penguins are affected by climate change may reveal why these birds are adjusting to the changing environment. Investigating penguin social behavior may also reveal important details about their social interactions and communication patterns. A more in-depth understanding of penguin populations might be possible by including data on individual penguins in the dataset, such as age, sex, and genetic information.

4.2 Data set II :

The King County House Sales dataset has been examined using a variety of data visualization methods and machine learning models. We have looked at a number of features including the location, land state, and year the house was built using the dataset, which gave us a thorough overview of the housing market in King County, USA. To visually locate patterns, correlations, and potential outliers in the data, we used pairplot and heatmap data visualization techniques. We discovered a significant positive correlation between a home's square footage of living space and price, as well as a moderate positive correlation between a home's grade and number of bathrooms. We also discovered that a house's price may not be strongly correlated with the year it was built. Then, to predict the sale price of a home in King County, we constructed two regression models, called Decision Tree and Random Forest. With a model score of 88%, the Random Forest algorithm performed better than the other algorithms. With a model score of 89%, the Lightgbm Library algorithm had the best performance.

Overall, our analysis identifies the critical elements that affect the sale price of a home in King County and offers beneficial insights into the connections between variables in the dataset. For upcoming buyers and sellers in the King County real estate market, our machine learning models can be utilized to forecast house prices with accuracy.

Future work : Possible future work related to this analysis could include:

1. **Feature engineering**: The current analysis used the features provided in the dataset as-is. Further exploration and feature engineering could be conducted to see if there are other variables that could be included or created to better predict housing prices in King County.
2. **Other machine learning models**: While decision trees and random forests were used in this analysis and provided excellent results with a model score of 75 and 88 respectively, other machine learning models such as multi-Linear regression and ridge regression could be tested to see if they produce better results.
3. **Time series analysis**: The current analysis did not take into account any time-series aspect of the data. Further exploration of the dataset could include time series analysis to determine if there are any trends or patterns in housing prices over time.
4. **Geospatial analysis**: Location was one of the most important features in predicting housing prices in this analysis. Further exploration could include geospatial analysis to determine if there are any geographic factors that are not currently captured in the dataset that could influence housing prices.

4.3 Data set III :

In the Credit Card Default dataset, the Random forest algorithm outperformed other algorithms in terms of results as it achieved an accuracy of 0.82 while Decision Tree achieved an accuracy of 0.80. However, to further improve the model's performance, we recommend incorporating Dimensional Reduction

Techniques. This is because the dataset contains almost 20 features to predict with only one target variable, making it difficult for the model to accurately capture all relevant information. By reducing the dimensionality of the dataset, we can potentially enhance the model's ability to identify important patterns and relationships between variables. After applying dimensional reduction techniques, we can compare the performance of the new model with the decision tree and random forest algorithm to determine which one yields the best results.

References

[1] P. Pandey, "Palmer archipelago (antarctica) penguin data," https://www.kaggle.com/parulpandey/palmer-archipelago-antarctica-penguin-data, 2020, accessed: 2023-05-05.

[2] HARLFOXEM, "King county house sale dataset," Available: https://www.kaggle.com/datasets/harlfoxem/housesalesprediction, Accessed on : 2016.

[3] I.-C. Yeh, "Uci machine learning repository: Default of credit card clients dataset," https://www.kaggle.com/datasets/uciml/default-of-credit-card-clients-dataset, 2009, accessed on: May 5, 2023.

[4] A. Vidhya, "Data exploration and visualization using palmer penguins dataset," https://www.analyticsvidhya.com/blog/2022/04/ data-exploration-and-visualisation-using-palmer-penguins-dataset/, 2022, accessed: May 5, 2023.

[5] P. Durganjali and M. V. Pujitha, "House price prediction using machine learning and feature engineering," in *2019 International Conference on Smart Structures and Systems (ICSSS)*. IEEE, 2019, pp. 454–458. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/8882842

[6] T. Wang, Y. Wang, and M. Liu, "A price prediction method based on catboost," *IEEE Xplore*, 11 2021. [Online]. Available: https://ieeexplore.ieee.org/document/9637683

[7] W. A. Chishti and S. M. Awan, "Deep neural network a step by step approach to classify credit card default customer," in *2019 International Conference on Innovative Computing (ICIC)*, January 2019. [Online]. Available: https://ieeexplore.ieee.org/document/8966723

[8] "IBM Cloud Learn: Data mining," https://www.ibm.com/cloud/learn/data-mining, accessed on: -.

[9] P. Gupta, "Decision trees in machine learning," Towards Data Science, 2021. [Online]. Available: https://towardsdatascience.com/decision-trees-in-machine-learning-641b9c4e8052

[10] W. Koehrsen, "Random forest in python," *Towards Data Science*, 12 2017. [Online]. Available: https://towardsdatascience.com/random-forest-in-python-24d0893d51c0

[11] Krishni, "Random forest regression," *DataDrivenInvestor*, 11 2018. [Online]. Available: https://medium.com/datadriveninvestor/random-forest-regression-9871bc9a25eb