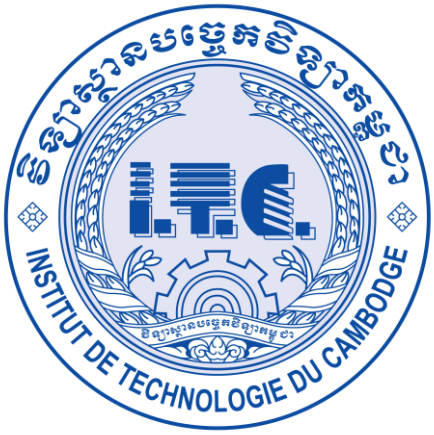

Moto price predictionGuideline

Member:Ngim panha and Lim sunheng
Institute of Technology of Cambodia, Phnom Penh, Cambodia
Department of Applied Mathematics and Statistics, Data Science
ID: e20200836,e20200807

Instuctor: Professor **CHAN Sophal**

Contents

Abstract

The goal of this paper is to predict the moto bike price. The graphical representation of predicting price is a process that aims for developing a computerized system to maintain all the price graph in any field and can predict price after a certain period. This application can take the database for the motorbike system from the organization and makes a graph through this information from the database. It will check the price and then import a graph that helps to observe the graphical representation. And then it can predict a certain period of price through the prediction algorithm. It can also be applied in some other effective predictions.

1 Introduction

The purpose of this machine learning project is to develop a predictive model that can estimate price base on location, power, series. The project aims to assist in price prediction, which can be valuable for customer to check which type they want. The ability to predict price ranges can provide insights into the market and help customer and company make informed decisions about their choices. It can also aid the target customer to get what they want.

By leveraging machine learning techniques, we can analyze the relationships between the independent variables (Locations, model, color, fuel type, transmission, fuel injection) and the target variable (price). This analysis will enable us to develop a model that can effectively predict moto price for customer.

The project will involve data collection, pre-processing, exploratory data analysis, feature engineering, model training, and evaluation. Various machine learning algorithms, such as logistic regression, decision trees, random forests, gradient boosting, support vector machines, k- nearest neighbors, naive Bayes, and neural net- works, will be explored and compared to deter- mine the most accurate and reliable model.

The ultimate goal of this project is to provide a robust and accurate motorcycle price estimation model that can be applied in real-world scenarios. The results of this project can have significant implications for motorcycle buyers, sellers, and the overall motorcycle market.

This guideline aims to provide a comprehensive framework for the project, outlining the essential steps, techniques, and best practices to ensure a systematic and effective approach towards achieving our objectives and delivering reliable results.

In the following sections, we will delve into the methodology, data collection, evaluation metrics, timeline, and expected outcomes of this machine learning project, providing a clear roadmap for its successful execution.

2 Problem Statement

This project aims to develop a machine-learning model that can accurately predict the price range of motorcycles based on various features. The prediction of motorcycle prices is important for both buyers and sellers to determine fair market value and make informed decisions.

The model will be trained using a dataset consisting of motorcycle listings along with their corresponding features, such as make, model, year, mileage, condition, location, and additional specifications. The objective is to analyze the relationships between these features and the price range and build a predictive model that can estimate the price range for new motorcycle listings.

By solving this problem, we aim to provide a valuable tool for motorcycle buyers and sellers to understand the price expectations for different motorcycle listings. Additionally, sellers can benefit from this model by gaining insights into the market rates and setting competitive prices for their motorcycles. The accurate prediction of motorcycle prices will enable buyers to make informed

decisions and negotiate fair deals, while sellers can optimize their pricing strategies based on market trends and demand.

2.1 Objectives

The main objectives of this project are:

- Develop a machine learning model that can predict the price range of motorcycles based on their features.
- Evaluate the model's accuracy in predicting the price range of motorcycles.
- Provide a tool for motorcycle buyers to estimate the price expectations for different motorcycle listings.
- Assist sellers in setting competitive prices for their motorcycles based on market rates.

2.2 Constraints

The project may face the following constraints:

- Limited availability of labeled data for training the model, which may impact the model's accuracy and generalization ability.
- Variability in motorcycle prices based on different makes, models, conditions, and geographic locations, making it challenging to capture all factors accurately.
- Inherent biases in the dataset, such as underrepresentation of certain motorcycle brands or regions, which may introduce biases in the model's predictions.
- Computational resources and time constraints for training and evaluating the model, as training a complex machine learning model with a large dataset can be computationally intensive.

2.3 Expected Deliverables

The expected deliverables of this project are:

- A trained machine learning model capable of predicting the price range of motorcycles.
- Documentation describing the data collection, preprocessing, model development, and evaluation processes.
- Visualizations and analysis of the relationships between features and motorcycle prices.
- Insights and recommendations based on the model's predictions and analysis to assist motorcycle buyers and sellers in making informed decisions.

2.4 Success Criteria

The success of this project will be measured by the following criteria:

- High accuracy in predicting the price range of motorcycles, validated through rigorous evaluation methods.
- Clear and actionable insights derived from the model's analysis of motorcycle features and their impact on prices.
- Positive feedback and acceptance from motorcycle buyers and sellers who utilize the model's predictions and insights.

3 Data Collection and Preprocessing

Data collection and preprocessing are fundamental steps in preparing the dataset for the machine learning model. In this project, we will outline the process of collecting and preprocessing the data to ensure its quality and suitability for the task of motorcycle price prediction.

3.1 Data Collection

The dataset for this project will be collected from various sources, such as khmer24, motorcycle dealer websites, classified ads platforms, or publicly available datasets related to motorcycle listings.

The data should include information on motorcycles, including features such as make, model, year, mileage, condition, location, and additional specifications, along with their corresponding price range.

Care should be taken to ensure the data is representative and diverse, covering different motorcycle makes, models, conditions, and geographic locations. It is important to collect a sufficient amount of data to ensure robust model training and evaluation. The dataset should encompass a wide range of motorcycles to capture the variations in prices based on different factors.

3.2 Data Cleaning

Once the data is collected for motorcycle price prediction, it needs to be preprocessed to handle inconsistencies, missing values, outliers, or formatting issues. The following steps should be performed during data preprocessing:

- Handling missing values: Identify any missing values in the dataset and handle them appropriately. This can involve techniques such as imputation or removal of incomplete data points for features like make, model, year, mileage, or condition.
- Handling outliers: Identify and address outliers, if present, in the dataset. Depending on the context, outliers can be treated through techniques such as removal or transformation to minimize their impact on the model's training.
- Encoding categorical variables: Convert categorical variables such as make, model, or condition into numerical representations suitable for machine learning algorithms. This can be achieved using techniques like one-hot encoding or label encoding.
- Feature scaling: If necessary, scale continuous numerical features to bring them to a similar scale, ensuring they have equal influence during model training. Common scaling techniques include
- standardization (e.g., z-score scaling) or normalization (e.g., min-max scaling).
- Splitting into training and testing sets: Divide the dataset into training and testing sets to evaluate the model's performance. The recommended split is typically around 80% for training and 20% for testing, ensuring an adequate amount of data for both phases.

3.3 Text Preprocessing

- Remove special characters, punctuation, and unnecessary white spaces.
- Tokenize the text into individual words or n-grams (contiguous sequences of n words) for further processing.
- Remove common stop words (e.g., ”the,” ”and,” ”is”) as they add little value to the prediction task.
- Perform stemming or lemmatization to reduce words to their root form and unify similar variations (e.g., ”run,” ”running” to ”run”).

3.4 Feature Extraction

- Extract relevant features from the preprocessed data, such as numerical features like year, mileage, and additional specifications, as well as categorical features like motorcycle make, model, and condition.
- Consider extracting additional features like location, engine size, fuel type, transmission type, or any other relevant information available in the dataset.
- Normalize numerical features to ensure they are on a consistent scale. Common normalization techniques include min-max scaling or z-score normalization

3.5 Feature Encoding

- Encode categorical features, such as motorcycle make, model, or condition, using techniques like one-hot encoding or label encoding.
- Convert any ordinal features, such as motorcycle condition (e.g., excellent, good, fair), into numerical representations while preserving their order. This allows the model to understand the inherent ranking

3.6 Train/Test Split

- Split the preprocessed data into training and testing datasets for price prediction.
- Ensure an appropriate split ratio, such as 70% for training data and 30% for testing data,

3.7 Handling Class Imbalance (if applicable)

- If the motorcycle prediction task has imbalanced classes (e.g., rare motorcycle makes or conditions), consider techniques like oversampling the minority class, undersampling the majority class, or using specialized algorithms like SMOTE (Synthetic Minority Over-sampling Technique).

3.8 Save Preprocessed Data

- Save the preprocessed data into a suitable format (e.g., CSV) for further analysis and modeling.

4 Exploratory Data Analysis

EDA is an essential step in understanding the dataset for motorcycle price prediction. It helps identify relationships, detect outliers, and gain insights into the data distribution. By performing EDA, we can gain a deeper understanding and extract valuable insights from the dataset.

4.1 Descriptive Statistics

Descriptive statistics offer an overview of the key characteristics of the dataset, encompassing measures like mean, median, mode, standard deviation, minimum, and maximum. These statistics aid in comprehending the central tendency, dispersion, and overall distribution of the data.

4.2 Data Visualization

Data visualization is a powerful tool for gaining insights from the motorcycle price prediction dataset. It allows us to visually analyze the relationships between variables, detect patterns, and identify any anomalies or outliers. Commonly used visualizations in motorbike prediction include:

- Histograms: to visualize the distribution of numerical variables like price or mileage.
- Box plots: to understand the distribution and identify outliers in numerical variables.
- Scatter plots: to explore the relationship between two numerical variables, such as price and year.
- Bar plots: to compare the frequency or distribution of categorical variables like make or condition.
- Heatmaps: to visualize the correlation between variables, such as the relationship between price and mileage.

4.3 Feature Relationships

Analyzing feature relationships in the motorcycle price prediction dataset is valuable. We can use correlation analysis for numerical variables and explore relationships between categorical variables using machine learning techniques. These analyses provide insights into important connections and dependencies, aiding informed decision-making in motorbike prediction.

4.4 Identifying Outliers

Identifying and understanding outliers is essential in motorbike price prediction. Statistical methods like Z-score or IQR can help detect outliers, ensuring accurate modeling. Handling outliers appropriately improves the reliability of the prediction model.

4.5 Feature Selection

Feature selection is crucial in motorbike prediction for identifying relevant features that contribute significantly. It reduces dimensionality, improves performance, and prevents overfitting. Techniques like correlation analysis, selection methods, and regularization are commonly used. Exploratory data analysis informs data preprocessing, feature engineering, and model selection, enhancing accuracy and efficiency.

By performing exploratory data analysis, we can gain insights into the dataset, understand its characteristics, and make informed decisions regarding data preprocessing, feature engineering, and model selection.

5 Feature Engineering

Feature engineering is a crucial step in the machine learning pipeline that involves transforming raw data into meaningful features that can improve the performance of predictive models. In this project, we will perform feature engineering to extract useful information and create new features from the existing dataset.

5.1 Handling Missing Data

Missing data can significantly affect model performance in motorbike prediction. It is crucial to handle missing values appropriately. Common techniques include imputation, where missing values are filled in using methods like mean, median, or mode. Alternatively, rows or columns with insignificant missing values can be removed. By addressing missing data, we ensure the integrity and accuracy of the motorbike prediction model.

5.2 Encoding Categorical Variables

Categorical variables in motorbike prediction need to be encoded into numerical representations for most machine learning algorithms. Common encoding techniques include one-hot encoding, label encoding, and ordinal encoding. The choice of encoding method depends on the nature of the categorical variable and the specific needs of the model. Proper encoding ensures compatibility with machine learning algorithms and facilitates accurate predictions in the motorbike prediction model.

5.3 Feature Scaling

Feature scaling is vital in motorbike prediction to ensure numerical features are on a consistent scale. It prevents features with larger scales from dominating the model. Common techniques include standardization (zero mean and unit variance) and normalization (0 to 1 or -1 to 1 range). Scaling facilitates fair comparisons and enhances model performance in motorbike prediction.

5.4 Feature Extraction

Feature extraction is a crucial step in motorbike prediction, involving the creation of new features from existing ones to capture additional relevant information. This can include mathematical transformations, interaction terms, or domain-specific conversions. Feature extraction requires careful consideration of the data and the specific problem to enhance predictive performance.

5.5 Dimensionality Reduction

Recursive Feature Elimination (RFE) is a useful technique for feature selection in motorbike prediction. It helps reduce the number of features by iteratively selecting the most important ones based on model performance. RFE is particularly beneficial when dealing with a large number of features, aiming to identify the subset most relevant to the target variable.

The RFE process involves:

- Training a machine learning model on the entire feature set.
- Ranking the features based on their importance, such as feature importance in tree-based models.
- Eliminating the least important feature(s) from the dataset.
- Retraining the model on the reduced feature set.
- Repeating the steps until the desired number of features is achieved.
- The number of selected features can be tuned as a hyperparameter based on model performance.

RFE is compatible with various machine learning algorithms like decision trees, random forests, support vector machines, and gradient boosting. It helps prevent overfitting and mitigates the challenges of high-dimensional data.

6 Model Selection and Training

Model selection is a crucial step in the machine learning pipeline, involving the selection of the most suitable algorithm or ensemble to address the prediction problem. In this project, we will assess and compare multiple models to determine the one that exhibits the best performance on our dataset.

6.1 Model Evaluation Metrics

Both KNN and Gradient Boosting models can be evaluated using common model evaluation metrics to assess their performance. Some commonly used metrics include:

- Accuracy: The proportion of correctly predicted instances out of the total instances.
- Precision: The proportion of true positive predictions out of all positive predictions (true positives + false positives). It measures the accuracy of positive predictions.
- Recall (Sensitivity or True Positive Rate): The proportion of true positive predictions out of all actual positive instances (true positives + false negatives). It measures the model's ability to identify positive instances correctly.
- F1-score: The harmonic mean of precision and recall. It provides a balanced measure of precision and recall.
- Confusion Matrix: A table summarizing the model's predictions against the true labels, displaying true positives, true negatives, false positives, and false negatives.

These metrics help evaluate and compare the performance of KNN and Gradient Boosting models, providing insights into their accuracy, precision, recall, and overall predictive capabilities.

6.2 Train-Test Split

Splitting the dataset into training and test sets using the train_test_split function from scikit-learn is a crucial step. This enables us to evaluate the performance of our models on unseen data.

6.3 Model Training

6.3.1 Model 1: K-Nearest Neighbors (KNN)

To implement the KNN model in scikit-learn, follow these steps:

- Instantiate an instance of the KNeighbors-Classifier class.
- Train the KNN model on the training data using the fit method.
- Make predictions on the test data using the predict method.
- Evaluate the performance of the KNN model using appropriate evaluation metrics.

6.3.2 Model 2: Gradient Boosting

- Instantiate an instance of the GradientBoostingClassifier class.
- Train the Gradient Boosting model on the training data using the fit method.
- Make predictions on the test data using the predict method.
- Evaluate the performance of the Gradient Boosting model using appropriate evaluation metrics.

7 Model Evaluation and Validation

Once the models have been trained, it is essential to assess their performance and validate their effectiveness. This evaluation step enables us to measure how well the models generalize to new, unseen data and verify their ability to make accurate predictions.

overall assessment of the model's performance.

7.1 Model Comparison

After evaluating and validating the models, it is crucial to compare their performance to determine the most effective model for the given task. This comparison can be based on the discussed evaluation metrics or specific project requirements. Factors like accuracy, interpretability, computational efficiency, and ease of implementation should be considered when making the final decision.

Thorough evaluation and validation of our models ensure their reliability, accuracy, and ability to make robust predictions. This assessment is essential for determining the success of our machine learning project and instilling confidence in deploying the models in real-world scenarios.