Purwadhika
Startup and Coding School

Personal Project

SAUDI USED CARS PRICE PREDICTION

Github: https://bit.ly/SaudiUsedCarsPrice

Content

PROJECT INTRODUCTION

Background

Engaging in the buying and selling of used cars in Saudi Arabia is quite a fascinating activity. According to information from the website www.pajakku.com, there are some great advantages to be gained when dealing with used cars in Saudi Arabia. One of these perks is the exemption from motor vehicle taxes in the country. This **encourages people in Saudi Arabia to sell and purchase used cars** within the nation.

When it comes to this economic activity, **people generally need a helpful guide to determine the right price for a used car**. This helps sellers set a fair price based on the car's specifications and ensures they make a profit. At the same time, potential buyers can also find out the prevailing market prices and make a purchase at the most suitable rate.

In light of this opportunity, a company called Syarah wants to create a friendly online marketplace platform catering to aspiring sellers and buyers of used cars. To meet the market's needs in **determining the prices of used cars in Saudi Arabia**, Syarah is seeking an IT solution.

Stakeholder

Syarah Company
an online marketplace company for used cars in Saudi Arabia

Problem Statement

1. How can we predict the prices of used cars in the Saudi Arabian market?
2. What variables influence the pricing of used cars in the Saudi Arabian market?

Objectives

1. Develop a machine learning model for predicting the prices of used cars in the Saudi Arabian market.
2. Identify the variables that significantly affect the pricing of used cars in the Saudi Arabian market.

Metric Evaluation

- RMSE : calculates the square root of the mean squared errors.
- MAE : calculates the mean of absolute errors
- **MAPE : calculates the mean percentage error generated by the regression model**

Domain Knowledge

Source of Data : Syara.com

Variable	*Description*
Type	Type of used car
Region	The region in which the used car was offered for sale.
Make	The company name.
Gear_Type	Gear type size of used car.
Origin	Origin of used car.
Options	Options of used car
Year	Manufacturing year.
Engine_Size	The engine size of used car.
Mileage	Mileage of used car
Negotiable	True if the price is 0, that means it is negotiable.
Price	Used car price.

DATA PREPARATION AND UNDERSTANDING

Data Preparation

01

- Import Library and Data Set
- General Information Exploration
- Handling of findings

02

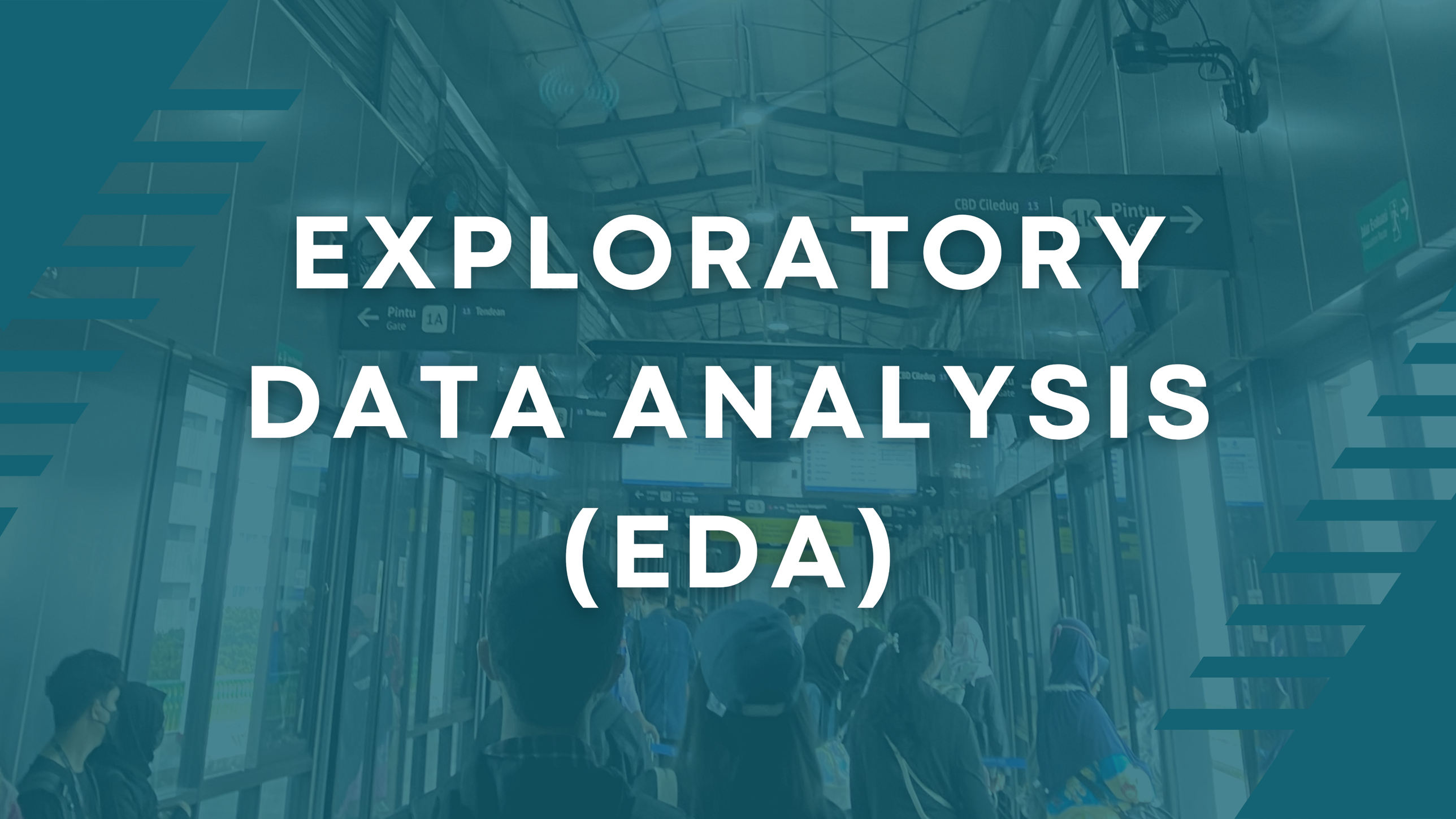
Data Cleaning:
Unnecessary Column, Data-Type, N/A Value, Standardize, Outliers, Duplicate

03

- Data Preparation Review
- Export cleansed data

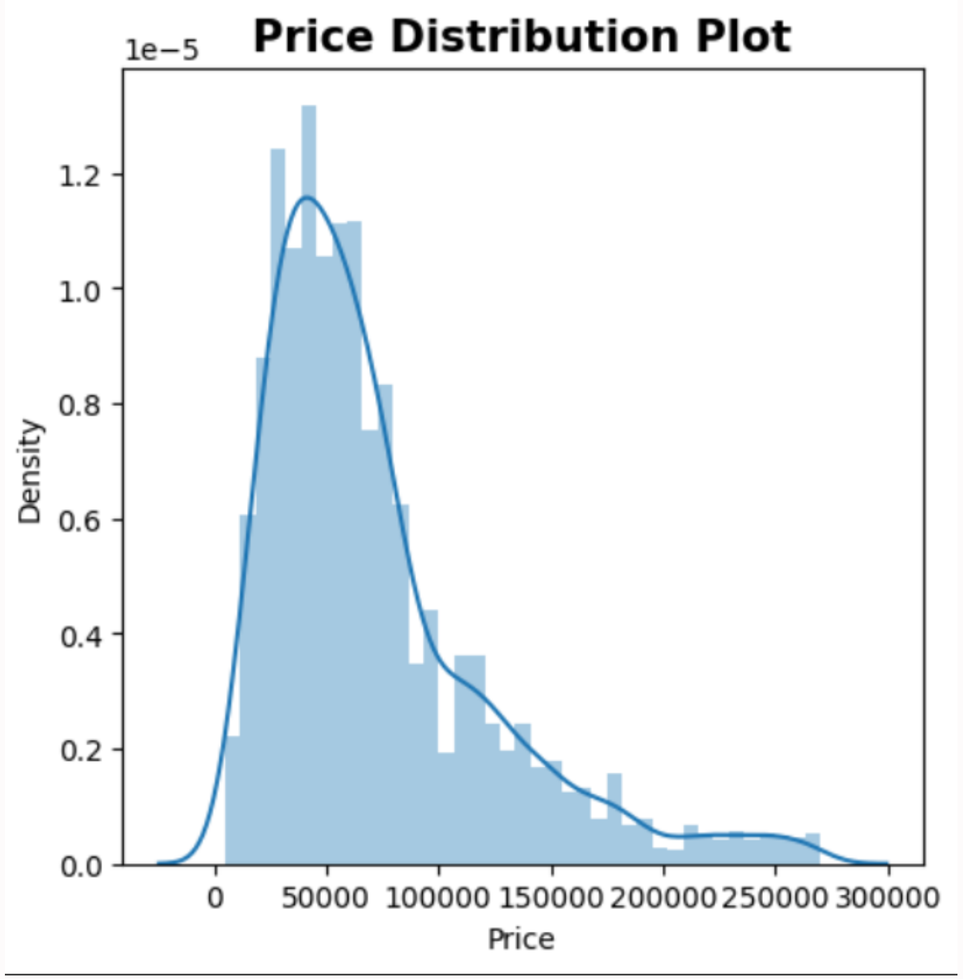
DATA PRE-PROCESSING

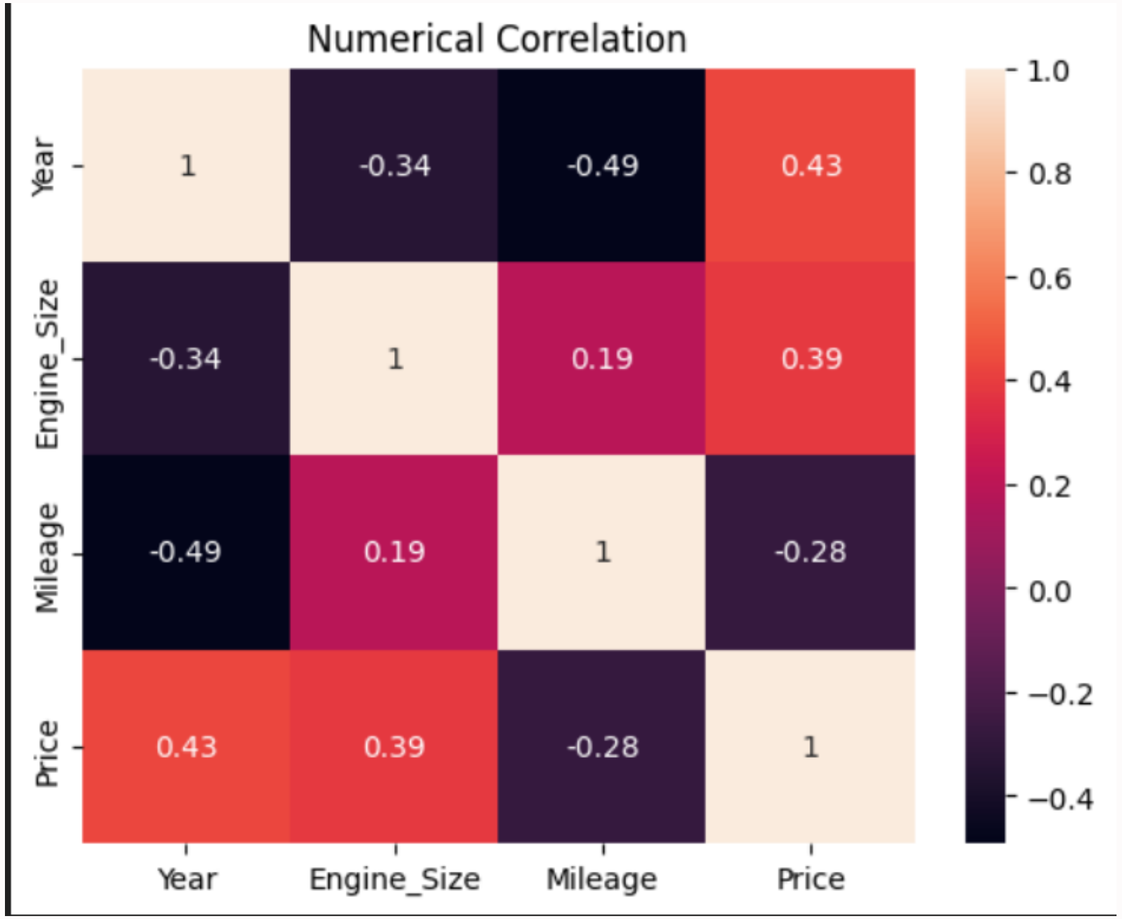
Data Pre-Processing



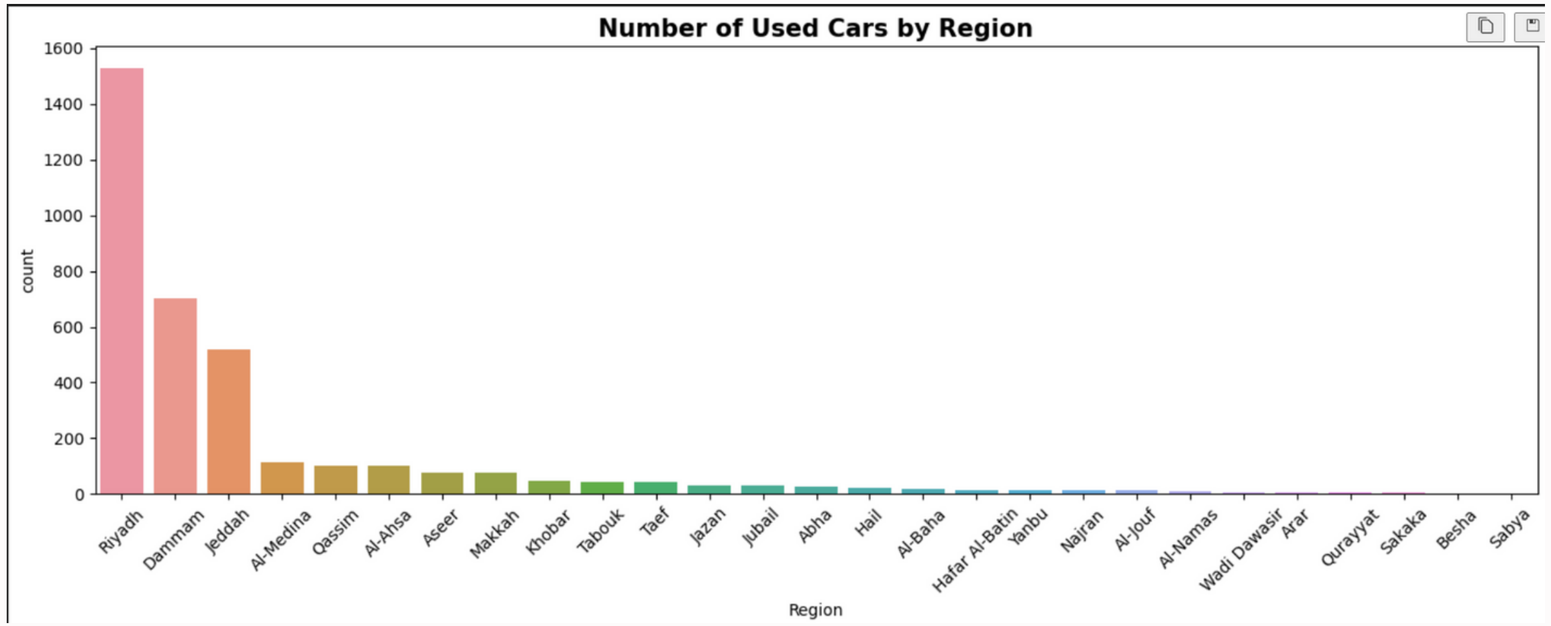
EXPLORATORY DATA ANALYSIS (EDA)

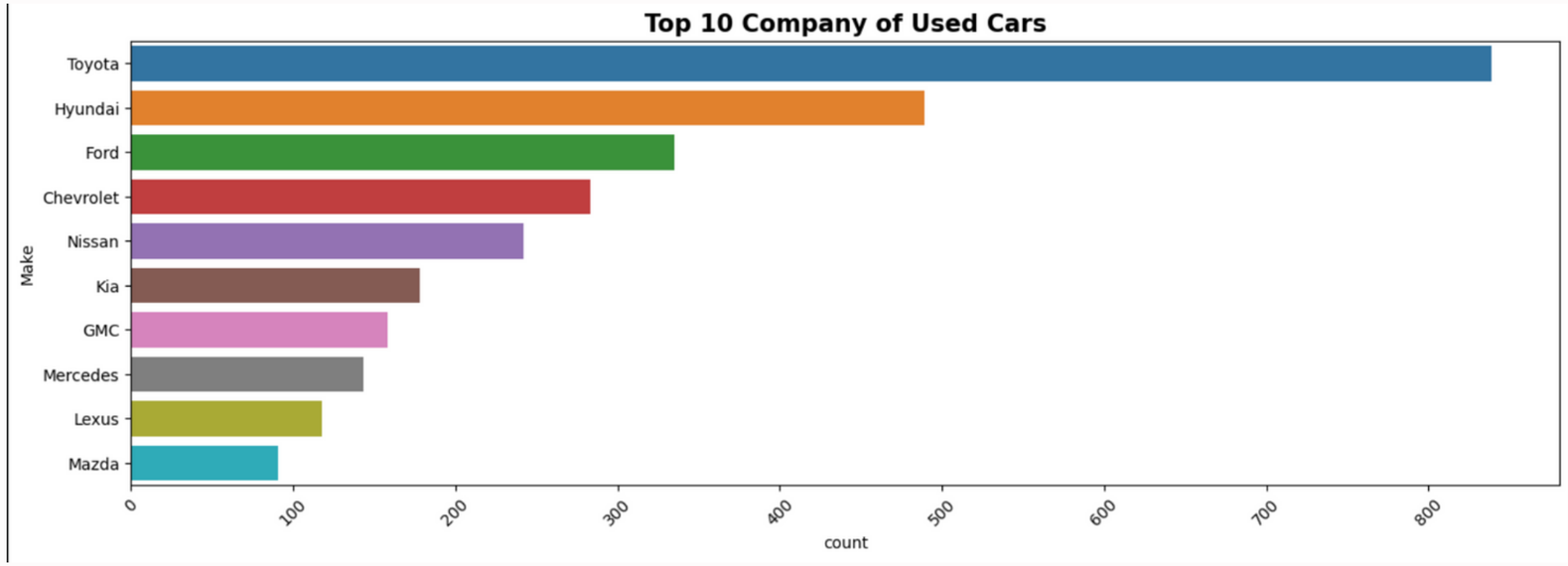
Exploratory Data Analysis (EDA)

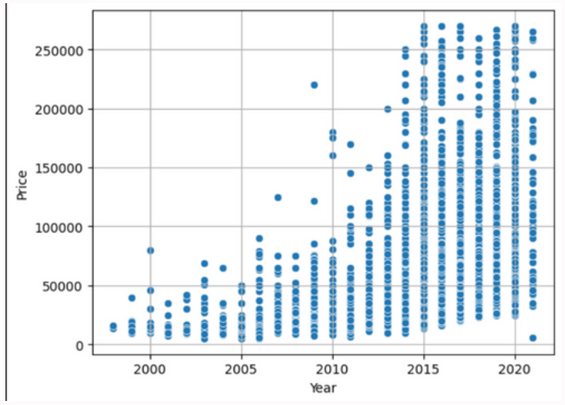


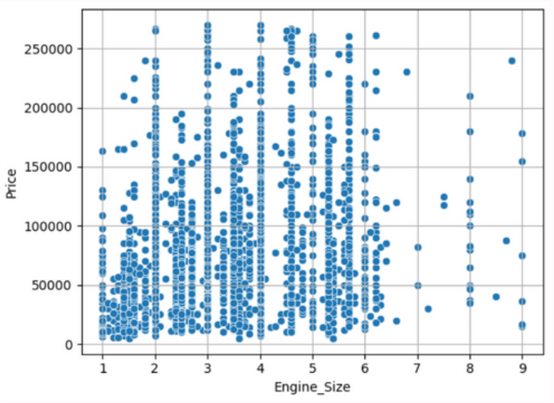


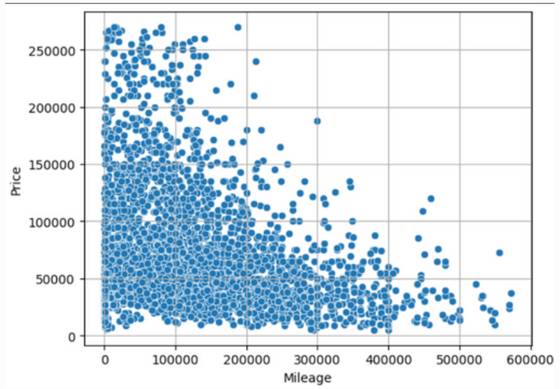
Exploratory Data Analysis (EDA)





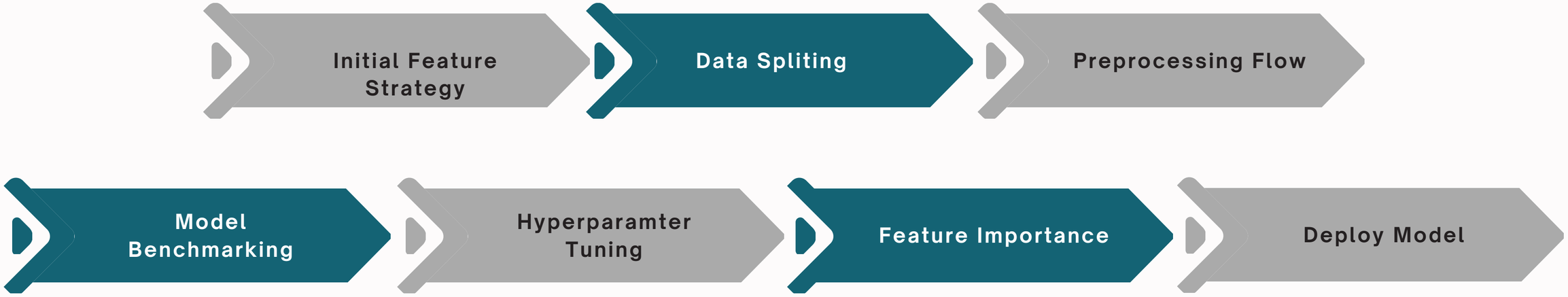






MODELING

Modeling

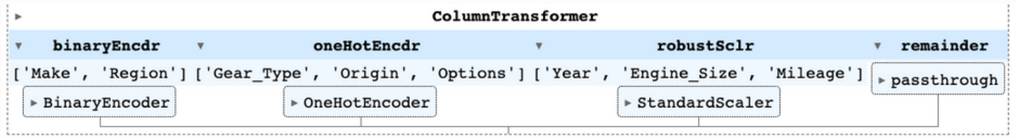

Initial Feature Strategy
Data Spliting
Preprocessing Flow
Model Benchmarking
Hyperparamter Tuning
Feature Importance
Deploy Model

Modeling

```
X_train, X_test, y_train, y_test= train_test_split(
    X,
    y,
    test_size=0.2,
    random_state=1)
```

Initial Feature Strategy → Data Spliting → Preprocessing Flow

- Binary Encoding -> 'Type', 'Make'
- One Hot Encoding -> 'Gear_Type', 'Origin', 'Options'
- Robust Scaler -> 'Year', 'Engine_Size', 'Mileage'
- Drop -> 'Region', 'Negotiable'



Modeling

Model Benchmarking

- LinearRegression
- KNeighborsRegressor
- DecisionTreeRegressor

- **RandomForestRegressor**
- **XGBRegressor**
- VotingRegressor
- StackingRegressor

Cross Validation

	Algorithm	RMSE_mean	RMSE_std	RMSE_score_all	MAE_mean	MAE_std	MAE_score_all	MAPE_mean	MAPE_std	MAPE_score_all
4	xgb	-24071.640	868.103623	[-24387.14, -24990.33, -22420.91, -24199.83, -...	-14774.680	606.735305	[-15211.22, -15117.98, -14921.82, -13576.03, -...	-0.228	0.011662	[-0.24, -0.23, -0.22, -0.21, -0.24]
3	random forest	-26839.848	882.762974	[-26440.11, -27484.26, -25982.74, -28241.31, -...	-16245.766	430.607813	[-16455.1, -16811.4, -16476.29, -15791.52, -15...	-0.244	0.013565	[-0.27, -0.24, -0.24, -0.23, -0.24]
6	stacking	-28685.560	1238.227960	[-28330.49, -30214.59, -26967.39, -29982.81, -...	-17410.612	643.381136	[-17370.78, -18599.75, -17381.24, -16964.82, -...	-0.268	0.007483	[-0.28, -0.27, -0.27, -0.26, -0.26]
5	voting	-29396.594	1303.243962	[-28767.39, -30572.39, -27560.44, -31161.84, -...	-17811.786	547.715643	[-17693.9, -18884.11, -17612.21, -17510.01, -1...	-0.276	0.010198	[-0.29, -0.28, -0.27, -0.28, -0.26]
1	knn	-30415.212	954.049881	[-29892.51, -31968.42, -31032.57, -29842.8, -2...	-19332.606	850.334460	[-19227.68, -20577.52, -19991.14, -18391.42, -...	-0.324	0.008000	[-0.32, -0.33, -0.33, -0.33, -0.31]
0	logreg	-36542.204	1727.768971	[-34435.64, -37875.81, -36851.49, -38836.4, -3...	-22825.398	598.775869	[-22590.0, -23358.05, -23633.94, -22579.42, -2...	-0.342	0.011662	[-0.35, -0.33, -0.33, -0.36, -0.34]
2	tree	-37558.582	1446.503570	[-38092.59, -39099.03, -35012.63, -37016.29, -...	-21851.078	1130.190012	[-22143.18, -23674.48, -20978.47, -20380.31, -...	-0.356	0.018547	[-0.39, -0.36, -0.34, -0.35, -0.34]

Modeling

Model Benchmarking

- LinearRegression
- KNeighborsRegressor
- DecisionTreeRegressor
- RandomForestRegressor
- **XGBRegressor**
- VotingRegressor
- StackingRegressor

Predict to Test Set

	RMSE	MAE	MAPE
XGB	22737.608368	15206.469859	0.271522
RandomForest	24197.892608	15668.360154	0.280039

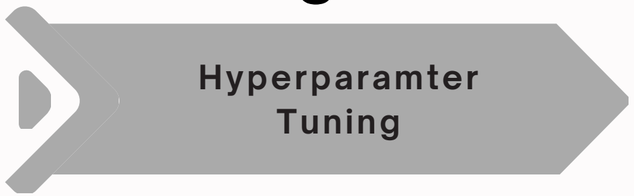
XGBRegressor

Model Benchmarking

- XGBRegressor utilizes the **gradient boosting** framework, which **combines multiple weak prediction models** (usually decision trees) to create a powerful ensemble model.
- It builds the **ensemble sequentially**, with each **new tree correcting the errors made by the previous trees**.
- It **focuses on samples with larger errors**, enabling the model to learn from its mistakes and improve over iterations.
- XGBRegressor **incorporates regularization techniques** such as controlling the tree depth, child weight, and gamma to prevent overfitting and enhance generalization.
- Its strengths lie in high accuracy, flexibility, handling complexity, and scalability.
- Considerations should be given to computational resources, parameter tuning, and interpretability.

Modeling

XGBRegressor

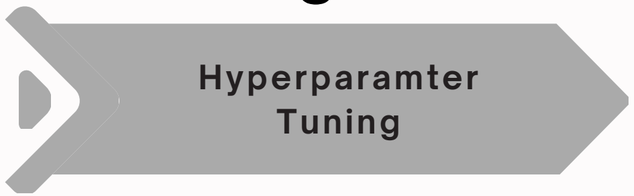


RandomizedSearchCV

- max_depth ->The maximum depth of each tree
- learning_rate -> Controls the step size at each boosting iteration
- n_estimators -> The number of boosting stages or trees to build
- subsample -> The fraction of samples used for training each tree
- gamma -> Minimum loss reduction required to make a further partition on a leaf node
- colsample_bytree -> The fraction of features used for training each tree
- reg_alpha -> L1 regularization term on weights

Modeling

XGBRegressor



RandomizedSearchCV

- max_depth ->The maximum depth of each tree -> **7**
- learning_rate -> Controls the step size at each boosting iteration -> **0.04**
- n_estimators -> The number of boosting stages or trees to build -> **468**
- subsample -> The fraction of samples used for training each tree -> **0.8**
- gamma -> Minimum loss reduction required to make a further partition on a leaf node -> **7**
- colsample_bytree -> The fraction of features used for training each tree -> **0.6**
- reg_alpha -> L1 regularization term on weights -> **0.00278255940220712 57**

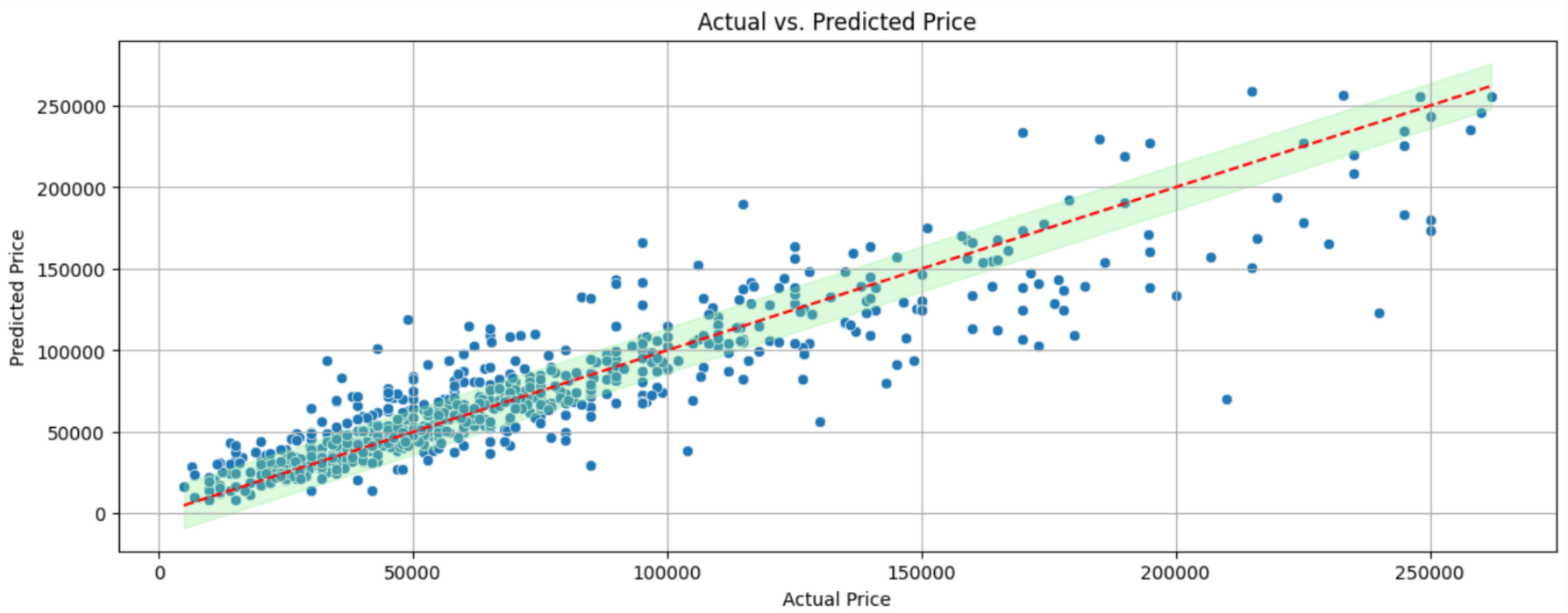
Modeling

XGBRegressor

Hyperparamter Tuning

Predict to Test Set

	Before	After
RMSE	22737.608368	20849.452273
MAE	15206.469859	13816.262108
MAPE	0.271522	0.243962



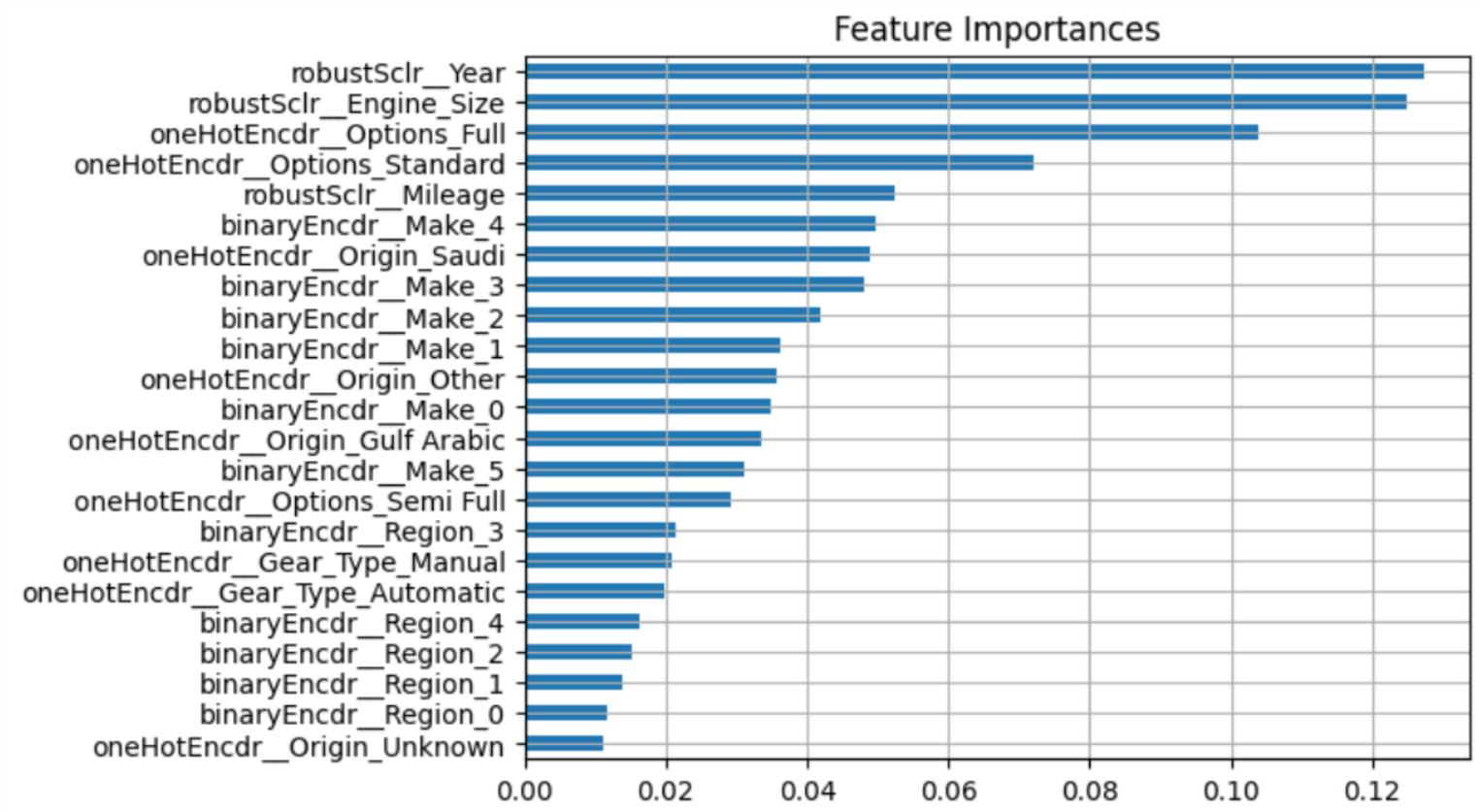
Modeling

```
import pickle
pickle.dump(xgb_tuning, open('Model_CarUsed_Syara.sav', 'wb'))
```
✓ 0.0s

```
loaded_model = pickle.load(open('Model_CarUsed_Syara.sav', 'rb'))
mean_absolute_percentage_error(y_test, loaded_model.predict(X_test))
```
✓ 0.0s

Feature Importance

Deploy Model



CONCLUSION & RECOMMENDATION

Conclusion

1. In achieving the objectives stated earlier, the following conclusions can be drawn:

- Develop a machine learning model for predicting the prices of used cars in the Saudi Arabian market.

A machine learning model has been successfully developed to predict the prices of used cars using the XGB Regressor algorithm, achieving an error level of approximately 0.24 when evaluated using the MAPE metric. This indicates that the predictions may deviate by up to 24% in either direction from the actual prices.

- Identify the variables that significantly affect the pricing of used cars in the Saudi Arabian market.`

Based on the information from the "Feature Importance" graph, the top 5 influential features in price formation, in descending order, are:

1. Year
2. Option (Full)
3. Engine Size
4. Option (Standar)
5. Option (Semi Full)

Meanwhile, the bottom 5 features are:

1. Origin (Unknown)
2. Region (1)
3. Region (0)
4. Gear_Type (Automatic)
5. Region (2)

Conclusion

2. Price predictions tend to align well for used cars priced below SAR 100,000. The data points are still close to the linear line x=y.

3. With all the features used in the dataset, no single feature is found to be significantly essential overall. Even the highest-ranking features show values around 0.13 on a scale of 0-1, indicating relatively balanced importance among the features.

Recommendation

In general, a model with an error rate of ~24% can be implemented as an initial step for a Price prediction product to be launched by Syarah. However, for future improvements and to enhance the performance of the existing model, we recommend the following steps:

1. Increase training data: Collecting a larger volume of diverse training data can provide the model with more examples and patterns to learn from. This can help improve its ability to generalize and make accurate predictions on unseen data.

2. Perform hyperparameter tuning: Utilize advanced computing resources and techniques to perform an extensive hyperparameter tuning process. This involves systematically exploring different combinations of hyperparameters and selecting the ones that yield the best performance. Grid search, random search, or Bayesian optimization can be employed to efficiently search the hyperparameter space.

Recommendation

3. Add additional features: Incorporate additional relevant features that can provide valuable information for predicting used car prices. This may include factors like color, seating capacity, service history, car type, accident history, or any other variables that have a significant impact on pricing. Thorough research and domain knowledge can help identify the most relevant features to include.

4. Explore polynomial features: Consider introducing polynomial features to capture more complex relationships between the features and the target variable. By including polynomial terms, the model can potentially capture non-linear patterns in the data and improve its predictive performance. Experiment with different degrees of polynomial features and assess their impact on the model's accuracy.

By implementing these recommendations, the model's performance can be enhanced, leading to more accurate predictions of used car prices. It is important to experiment, evaluate, and iterate on these steps to continually improve the model's performance over time.

Purwadhika
Startup and Coding School

Personal Project

SAUDI USED CARS PRICE PREDICTION

Github: https://bit.ly/SaudiUsedCarsPrice