Assignment #13: Final Report
Course: BDAT 1011 – Data Analytics Project
Professor: Brian Broda

Title: Analysing Factors Affecting the Price of Cars in Canada

Overview

Amidst the aftermath of the pandemic, there has been a substantial surge in second-hand car prices across Canada, indicative of a notable shift in consumer behavior. Individuals, in response to prolonged waiting periods for new releases, are increasingly opting for purchasing slightly older models. This trend is a significant contributor to the soaring prices of used cars in the Canadian market. Seizing this opportunity, resellers and dealers strategically manipulate the actual market prices of used cars.

To confront this challenge head-on, our project takes on the mantle of creating awareness among potential buyers seeking used cars. The overarching objective is to empower individuals in navigating the complexities of second-hand car pricing, providing them with comprehensive insights. By doing so, we aim to equip both buyers and other stakeholders in the automotive industry with the necessary information to make informed decisions.

Our project is designed to conduct a thorough analysis, delving into historical second-hand car price data and relevant variables. Through this exploration, we seek to unravel the intricacies of market dynamics, enabling us to make informed predictions about future trends. This strategic approach positions our project at the forefront of addressing a pertinent issue in the Canadian automotive landscape.

Project Team:

1. Tilak Pandya
   Domain: Artificial Intelligence, Software engineer.

2. Kushal Ghimire
   Domain: Front- End Developer, Web Designer.

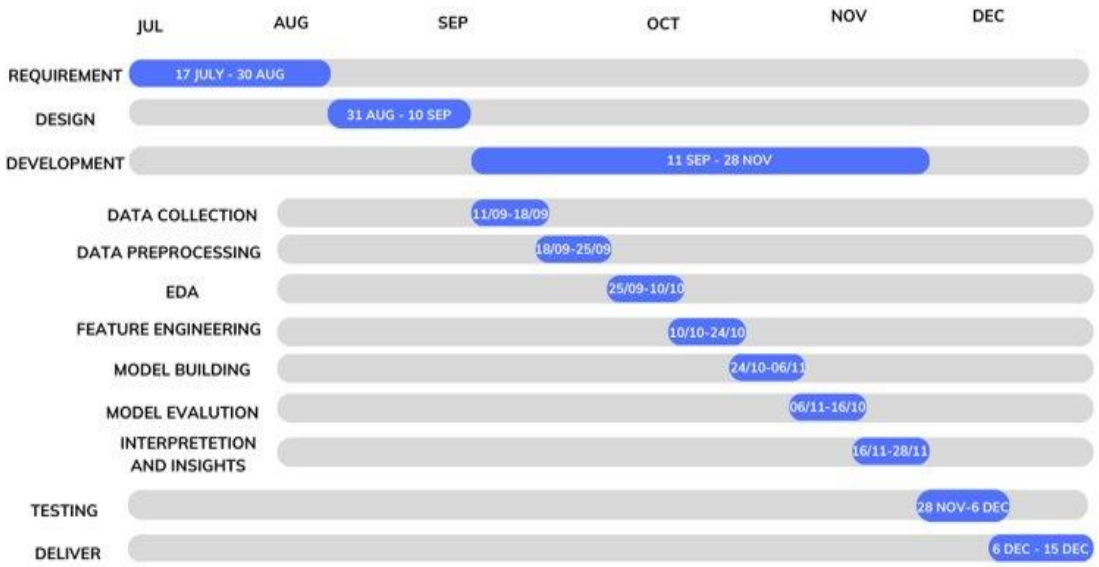
3. Satya Gaurav Palakollu
   Domain: Cyber Security, Electronics and Communications.

Project Planning:

Executed the project by applying machine learning techniques, specifically Random Forest and Gradient Boosting, to predict car prices. Followed a systematic data analytics workflow involving data preprocessing, model training, and feature analysis while adhering to Python programming conventions and utilizing Plotly and matplotlib Express for data visualization.

Explore and analyze the relationships between car prices and various attributes within the dataset to uncover patterns and trends. Various sources were explored to get reference for analysis such as Kaggle, Github and Medium. Identify and explain the main factors significantly influencing car prices, aiming to understand the key drivers in the market. To fulfill these objectives a concrete plan was created:

- Identify and collect a comprehensive dataset from reliable sources that includes relevant information on used car prices in Canada.
- Standardize and preprocess the data to make it suitable for analysis.
- Conduct an initial exploratory analysis to understand the distribution of data, identify trends, and detect outliers.
- Utilize statistical methods to identify significant correlations between cars price and other attributes.
- Interpret the findings and insights obtained from the analysis.
- Validate the results through sensitivity analysis or by testing the model on new data.



Execution Plan for Development:

1. Data Collection:
Gather a comprehensive dataset containing car sales information from various sources.

2. Data Preprocessing:
Clean the data by handling missing values, duplicates, and converting categorical variables into numerical representations.

3. Exploratory Data Analysis (EDA):
Visualize and analyze relationships between car features and prices to gain insights.

4. Feature Engineering:
Create new variables or transform existing ones to enhance the predictive model's performance.

5. Model Building:
Use regression techniques like linear regression, decision trees, random forests, or gradient boosting to build the predictive model.

6. Model Evaluation:
Assess the model's performance using metrics such as MAE, RMSE, and R-squared.

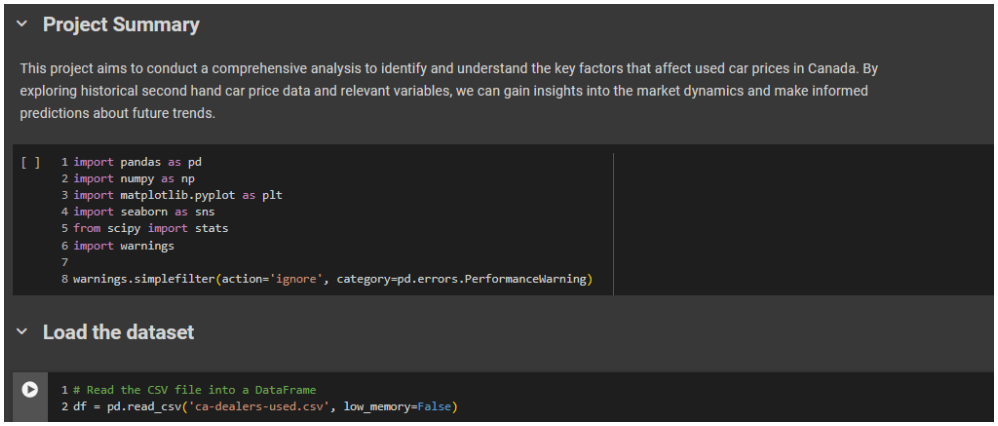
7. Model Selection:
Compare different models to select the one with the best accuracy in predicting car prices.

8. Interpretation and Insights:
Interpret the model's coefficients to identify the most influential factors affecting car prices using python libraries for visualization.

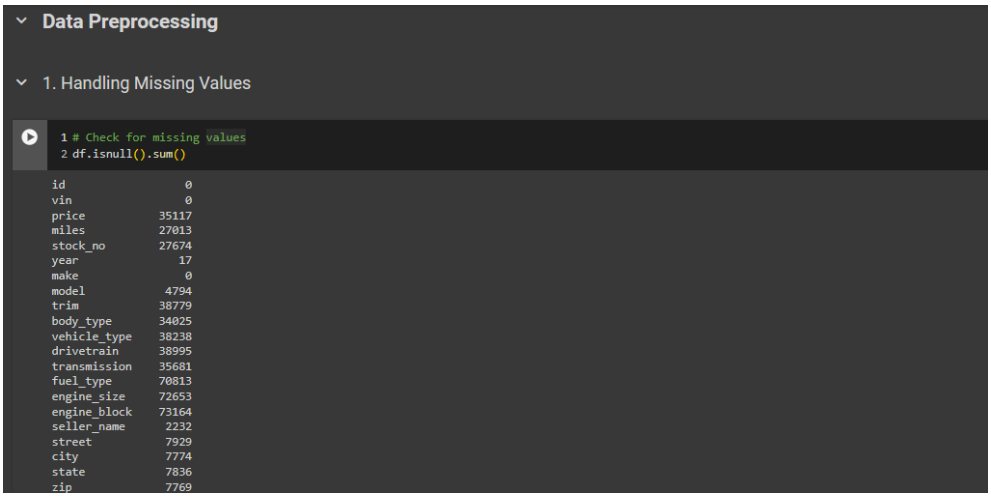
9. Summarization:
Summarize the findings and insights, providing valuable information for the automotive industry stakeholders.

By following these steps, systematic planning and execution of the project could be implemented, ensuring a thorough analysis of the correlations and dependencies within the car pricing dataset.

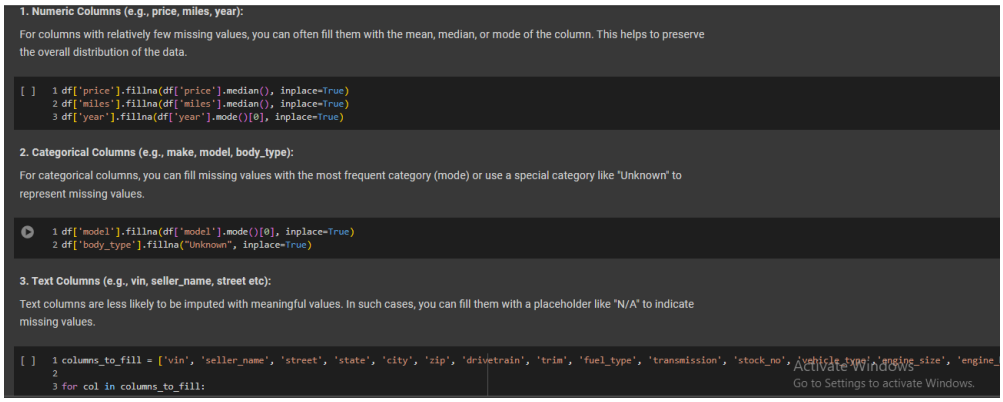
Python Scripts



All the important libraries imported and dataset were read using pandas library in python.



Here, Data preprocessing was the first step, there were multiple missing values in dataset shown above.



Numeric, Categorical and Text columns were handled individually with median, mode or ‘N/A’ text due to different types of data.

Again check if there is still any missing value or not.

```
[ ]  1 df.isnull().sum()
```

```
id              0
vin             0
price           0
miles           0
stock_no        0
year            0
make            0
model           0
trim            0
body_type       0
vehicle_type    0
drivetrain      0
transmission    0
fuel_type       0
engine_size     0
engine_block    0
seller_name     0
street          0
city            0
state           0
zip             0
dtype: int64
```

Again, we check if still any missing value available in dataset or not.

2. Outliers Handling

Handling outliers in numeric columns is an important step in data preprocessing. Outliers can significantly impact the performance of machine learning models.

```
[ ]  1 # Define a threshold for the Z-score
     2 threshold = 3
```

```
[ ]  1 #Store numeric columns
     2 numeric_columns = df.select_dtypes(include=[np.number]).columns.tolist()
     3 numeric_columns
```

```
['price', 'miles', 'year']
```

```
1 # Loop through each numeric column and handle outliers
2 for column in numeric_columns:
3     z_scores = np.abs(stats.zscore(df[column]))
4     df = df[(z_scores < threshold)] # Remove data points with z-score < 3 (adjust the threshold as needed)
```

We first define a threshold for the Z-score (commonly set to 3).

We select all numeric columns using df.select_dtypes(include=[np.number]).columns.tolist().

Then, we loop through each numeric column, calculate the Z-scores for the data points in that column, and keep only the data points where the absolute Z-score is less than the defined threshold. This effectively removes outliers from each numeric column.

Defining a Threshold for Z-scores: The first cell defines a threshold for the Z-scores and stores the numeric columns in a list.

Calculating Z-scores and Removing Outliers: The second cell loops through the numeric columns, calculates the Z-scores for each column, and removes the outliers from the data frame.

3. Encoding Categorical Variables

Convert all categorical variables into numerical using one-hot encoding

```
[ ] from sklearn.preprocessing import LabelEncoder
    from sklearn.model_selection import train_test_split
```

```
[ ] categorical_cols = ['make','model','body_type','vehicle_type','drivetrain','transmission','trim','fuel_type','engine_size', 'engine_block', 'state', 'city']
```

```
[ ] encoding_dict = {}
    # Iterate through each categorical column
    for column in categorical_cols:
        unique_values = df[column].unique()

        # Create a mapping of unique values to index numbers
        value_to_index = {value: index for index, value in enumerate(unique_values, start=1)}

        # Add the mapping to the encoding dictionary
        encoding_dict[column] = value_to_index

    # Display the encoding dictionary
    print(encoding_dict)
```

```
{'make': {'Acura': 1, 'Dodge': 2, 'Chrysler': 3, 'RAM': 4, 'Ford': 5, 'Chevrolet': 6, 'Buick': 7, 'Cadillac': 8, 'GMC': 9, 'Lexus': 10, 'Volkswagen': 11, 'BMW': 12, 'Mercedes-Benz': 13, 'Hummer': 14, 'Lincoln': 15, 'Volvo': 16, 'INFINITI': 17, 'Toyota': 18, 'Land Rover': 19, 'Bentley': 20,
```

```
[ ] # Map the encoding_dict to the DataFrame
    data_encoded = df.copy()  # Create a copy of the DataFrame to keep the original data

    # Iterate through each categorical column and replace values with index numbers
    for column, value_to_index in encoding_dict.items():
        data_encoded[column] = data_encoded[column].map(value_to_index)
```

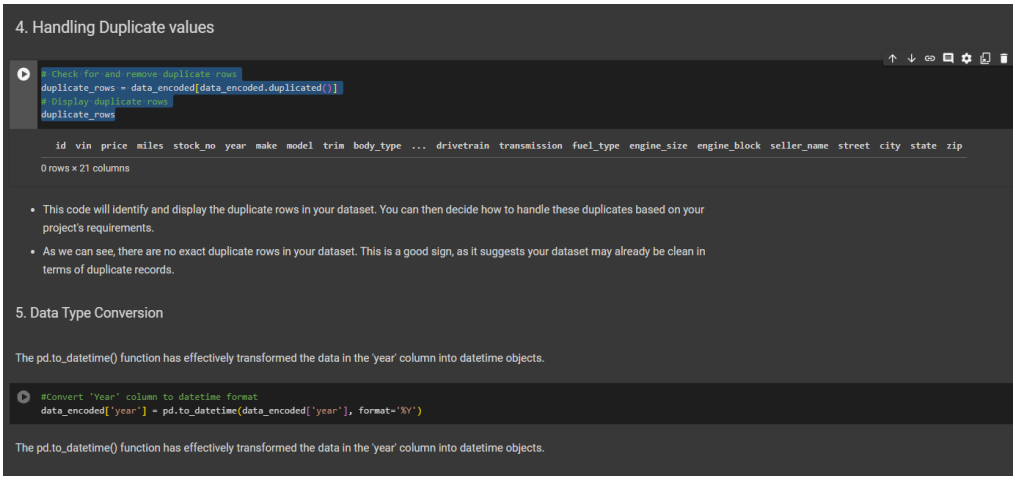
```
data_encoded.head()
```

	id	vin	price	miles	stock_no	year	make	model	trim	body_type	...	drivetrain	transmission	fuel_type	engine_size	engine_block	seller_name	street	city	state	zip
15	79d4a6a9-c265	19UNC1B0XHY800061	21900.0	12864.0	MUZ483	2017.0	1	1	1	1	...	1	1	1	1	1	pfaff leasing calgary	5539 6th Street Se	1	1	T2H 1L6
16	a7b53aaa-ae99	1B3LC56R18N652133	3499.0	174850.0	N/A	2008.0	2	2	2	2	...	2	1	2	2	1	carview motor	1113 Finch Ave W	2	2	M3J 2E5
17	acdf10ba-fa76	1B3LC56R18N660054	5200.0	89124.0	215140	2008.0	2	2	2	2	...	2	1	2	2	1	strickland's brantford chevrolet buick gmc cad...	19-21 Lynden Road	3	2	N3R 8B8
18	3ec82d01-aec0	1B3LC56R38N195397	4789.0	151745.0	30479	2008.0	2	2	2	2	...	2	1	2	2	1	dale wurfel chrysler dodge jeep ltd.	28478 Centre Road	4	2	N7G 3J2
20	d128f381-0f23	1B3LC56R88N566740	5995.0	176631.0	2008DAVGWHT	2008.0	2	2	2	2	...	2	1	2	2	1	first edmonton auto	8303 118 Avenue Northwest	5	1	T5B 0S4

5 rows × 21 columns

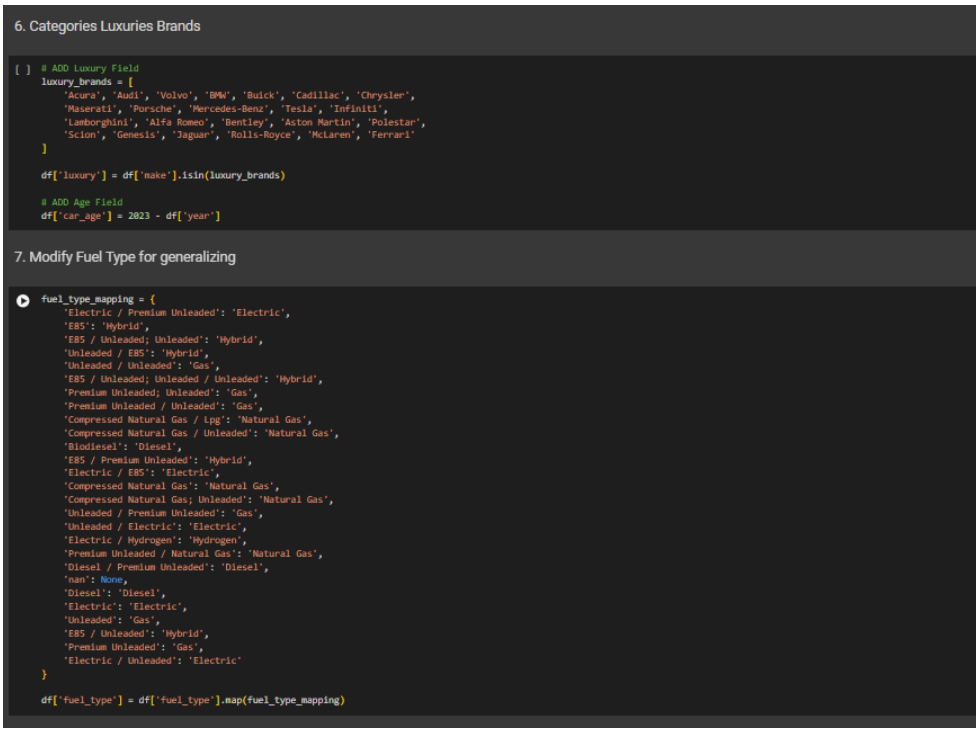
This code will create binary (0 or 1) columns for each category in the specified categorical columns, effectively converting them into a numerical format suitable for machine learning algorithms.

Encoding Categorical Variables: A function named encode_categorical is defined, which takes a DataFrame and a list of columns as arguments. This function encodes the categorical variables in the specified columns using LabelEncoder.



Handling Duplicate Values: The code checks for duplicate rows in the DataFrame data_encoded using data_encoded.duplicated(). It then displays these duplicate rows. If there are any duplicate rows, handle them based on project's requirements.

Data Type Conversion: The code converts the 'year' column in the DataFrame data_encoded to datetime format. This is useful when time series analysis or extract features from the date like the year, month, day, etc.



Categories Luxuries Brands: The code creates a new column 'luxury' in the DataFrame df that indicates whether the 'make' of the car is in the list of luxury brands. It also creates a new column 'car_age' by subtracting the 'year' column from 2023. This gives the age of each car.

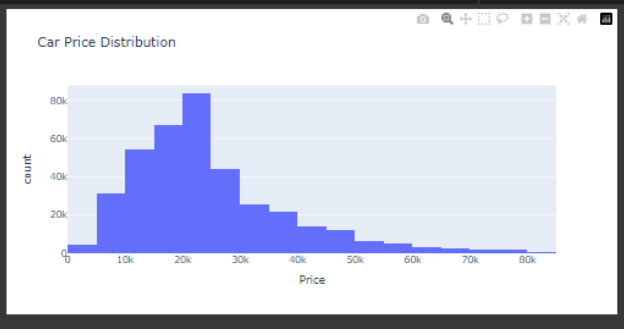
Modify Fuel Type for generalizing: Categories fuel_type in "Diesel"," Gas", "Hybrid", "Natural Gas", "Hydrogen" and "Electric".

Exploratory Data Analysis (EDA)

1. Histogram for Price Distribution

This visualization helps you understand the distribution of car prices in your dataset.

```
import plotly.express as px

# Assuming df is your DataFrame and 'price' is the column representing car prices
fig = px.histogram(df, x='price', nbins=40, title='Car Price Distribution',
                   labels={'price': 'Price', 'count': 'Frequency'},
                   template='plotly', width=800, height=400)

# Show the plot
fig.show()
```



Visualization for Price Distribution: The code is creating a histogram to visualize the distribution of car prices.

2. Bar Chart for Average Car Prices of the model of the Car by Year

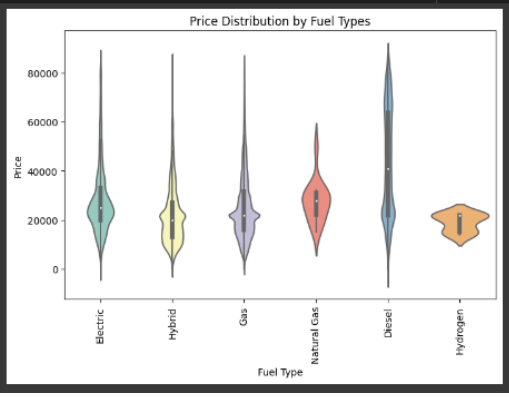
```
import plotly.express as px
import pandas as pd

# Filter the DataFrame for Ford cars and non-zero years
ford_df = df[(df['make'] == 'Ford') & (df['year'] > 0)]

# Group by year and calculate the average price for Ford cars
avg_price_by_year = ford_df.groupby('year')['price'].mean().reset_index()

# Plotting the interactive bar chart
fig = px.bar(avg_price_by_year, x='year', y='price', title='Average Car Prices for Ford Model by Year',
             labels={'year': 'Year', 'price': 'Average Price'},
             template='plotly', width=800, height=600)

# Show the plot
fig.show()
```

Visualization for Average Car Prices for Ford Model by Year: The code is creating a bar chart to visualize the average car prices for a Ford model by year.

3.Violin Plot for Price Distribution by Fuel Type

This violin plot illustrates the distribution of car prices for different fuel types.

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

top_fuel_types = df['fuel_type'].value_counts().head(10).index
filtered_df = df[df['fuel_type'].isin(top_fuel_types)]

plt.figure(figsize=(8, 5))
sns.violinplot(x='fuel_type', y='price', data=filtered_df, palette='Set3', split=False)
plt.title('Price Distribution by Fuel Types')
plt.xlabel('Fuel Type')
plt.ylabel('Price')
plt.xticks(rotation=90)
plt.show()
```

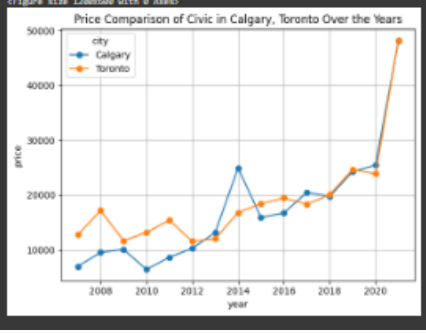


Visualization for Price Distribution by Fuel Type: The code is creating a violin plot to visualize the distribution of car prices by fuel type.

4.Time Series Plot for Car Prices according to the cities Over the Years:

```
import matplotlib.pyplot as plt
import pandas as pd

# Filter the DataFrame for the specific car model (e.g., Elantra) and cities (e.g., Calgary and Toronto)
selected_cities = ['Calgary', 'Toronto']
selected_model = 'Civic'

filtered_df = df[(df['city'].isin(selected_cities)) & (df['model'] == selected_model)]

# Clean the 'price' column by removing non-numeric values and converting to a numeric data type
filtered_df['price'] = pd.to_numeric(filtered_df['price'], errors='coerce')

# Drop rows with missing or non-numeric values in the 'price' column
filtered_df.dropna(subset=['price'], inplace=True)

# Aggregate data for the same year and city combination (e.g., by taking the mean)
pivot_df = filtered_df.groupby(['year', 'city'])['price'].mean().unstack()

# Create a line plot to compare prices over the years
plt.figure(figsize=(12, 6))
pivot_df.plot(kind='line', marker='o')
plt.title(f'Price Comparison of {selected_model} in {", ".join(selected_cities)} Over the Years')
plt.xlabel('year')
plt.ylabel('price')
plt.grid(True)
plt.legend(title='city')
plt.tight_layout()
plt.show()
```

```
<ipython-input-24-b8d3f7bf3d2b>:11: SettingWithCopyWarning:


A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy

<ipython-input-24-b8d3f7bf3d2b>:14: SettingWithCopyWarning:


A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy

<Figure size 1200x600 with 0 Axes>
```

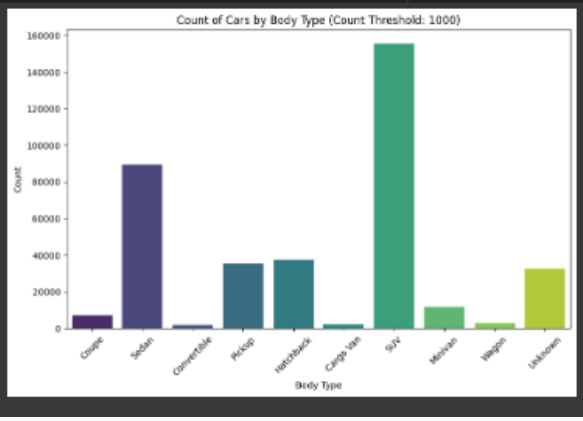


Visualization for Price Comparison of Honda Civic Car in Calgary, Toronto Over the Years: The code is creating a line graph to visualize the price comparison of Honda Civic Car in Calgary and Toronto over the years.

5. Count Plot for Car Body Type:

A count plot can show the distribution of car body types in your dataset.

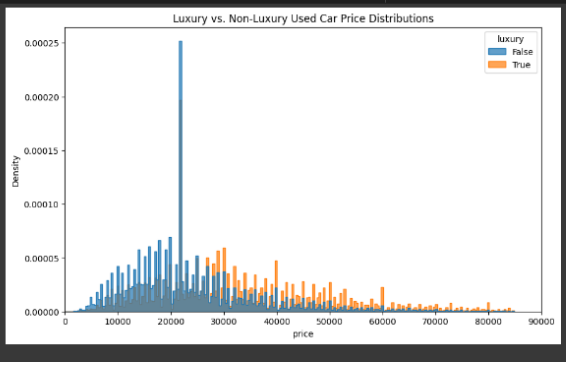
```
import seaborn as sns
import matplotlib.pyplot as plt

# Calculate the counts of each body type
body_type_counts = df['body_type'].value_counts()

# Set a threshold for the minimum count
count_threshold = 1000  # Adjust this threshold as needed

# Filter body types that meet the count threshold
filtered_body_types = body_type_counts[body_type_counts >= count_threshold].index

# Filter the DataFrame based on the selected body types
filtered_df = df[df['body_type'].isin(filtered_body_types)]

plt.figure(figsize=(10, 6))
sns.countplot(data=filtered_df, x='body_type', palette='viridis')
plt.title(f"Count of Cars by Body Type (Count Threshold: {count_threshold})")
plt.xlabel("Body Type")
plt.ylabel("Count")
plt.xticks(rotation=45)  # Rotate x-axis labels for better readability
plt.show()
```



Visualization for Car Body Type Count: The code is creating a bar graph to visualize the count of cars by body type. This suggests that in the dataset SUVs, Sedans, and Trucks are more common than Convertibles, Wagons, and Vans.

6. Distribution of prices between luxury cars:

```
import seaborn as sns
import matplotlib.pyplot as plt

plt.figure(figsize=(10, 6))
sns.histplot(df, x='price', hue='luxury', element='step', common_norm=False, alpha=0.7, stat='density')
plt.xlim(0, 90000)
plt.title("Luxury vs. Non-Luxury Used Car Price Distributions")
plt.show()
```



Visualization for Luxury vs. Non-Luxury Used Car Price Distribution: The code is creating a bar graph to visualize the price distribution of luxury and non-luxury used cars. From the chart, it appears that luxury cars have a higher density at higher prices, while non-luxury cars have a higher density at lower prices.

Feature Engineering

1. Feature Scaling

Scaling numeric features can be important, especially if we need to use models that are sensitive to feature scales. We can use techniques like Min-Max scaling or Standardization to scale your numeric features.

```
from sklearn.preprocessing import StandardScaler

# Initialize the StandardScaler to scale numerical features
scaler = StandardScaler()
new_data = data_encoded.drop(columns=['vin','id','stock_no','seller_name', 'street', 'zip','year'])

print(new_data.columns)

# Select the numeric columns in your dataset for scaling
numeric_columns = data_encoded.select_dtypes(include=[np.number]).columns.tolist()

# Scale the selected numeric columns
data_encoded[numeric_columns] = scaler.fit_transform(data_encoded[numeric_columns])
```

```
Index(['price', 'miles', 'make', 'model', 'trim', 'body_type', 'vehicle_type',
       'drivetrain', 'transmission', 'fuel_type', 'engine_size',
       'engine_block', 'city', 'state'],
      dtype='object')
```

Standardization rescales the numeric features to have a mean of 0 and a standard deviation of 1. This ensures that the features have a similar scale, which is important for some machine learning algorithms that are sensitive to feature scales. It helps to improve the model's performance by making the features more comparable. The code identifies numeric columns in the DataFrame, applies the standardization transformation, and updates the DataFrame with the scaled values.

2.Binning Numeric Features

Discretize numeric features by binning them into categories.

```
import datetime

# Get the current year
current_year = datetime.datetime.now().year
# Calculate 'car_age' by subtracting the 'year' of the car from the current year
data_encoded['car_age'] = current_year - df['year']
```

Feature Engineering: The code is performing feature engineering, which is an important step in preparing the data for machine learning models. It mentions that scaling is important, especially for models that are sensitive to feature scale. Standardization is a scaling technique where the features are centered around zero with a standard deviation of one. It transforms the feature to have a mean of 0 and a standard deviation of 1.

Model Building

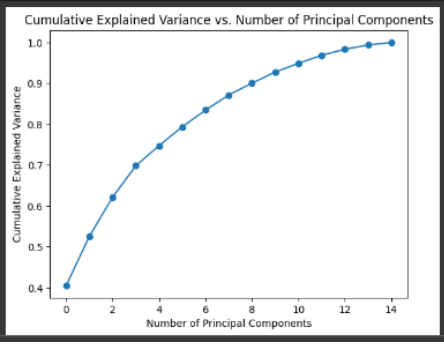
Regression models are chosen for this dataset because the goal is to predict a continuous numerical variable, which is the price of cars. In this case, the price is not limited to specific categories or classes, but can take on a wide range of numerical values. Regression models are specifically designed to model and predict continuous values, making them the most appropriate choice for this type of prediction task.

```
from sklearn.linear_model import LinearRegression
from sklearn.tree import DecisionTreeRegressor
from sklearn.ensemble import RandomForestRegressor, GradientBoostingRegressor
```

```
# Drop columns that are not required for modeling
data_encoded = data_encoded.drop(columns=['vin','id','stock_no','seller_name', 'street', 'zip','year'])
```

PCA - Principal Component Analysis

```
import numpy as np
from sklearn.decomposition import PCA
import matplotlib.pyplot as plt

# Make sure the data is properly preprocessed and standardized before applying PCA

# Create a PCA instance
pca = PCA()

# Fit the data and transform it
data_pca = pca.fit_transform(data_encoded)

# Plot the explained variance ratio
explained_variance_ratio = pca.explained_variance_ratio_
cumulative_explained_variance = np.cumsum(explained_variance_ratio)

plt.plot(cumulative_explained_variance, marker='o')
plt.xlabel('Number of Principal Components')
plt.ylabel('Cumulative Explained Variance')
plt.title('Cumulative Explained Variance vs. Number of Principal Components')
plt.show()
```



Principal Component Analysis (PCA): The code is performing PCA, which is a technique used in machine learning to reduce the dimensionality of datasets while preserving as much information as possible. It's often used when dealing with high-dimensional data. The eigenvalue associated with each principal component represents the variance of the data along that direction.

```
[ ] pca = PCA(n_components=10)
    data_pca = pca.fit_transform(data_encoded)
    # Create a DataFrame to store the loadings
    loadings_df = pd.DataFrame(pca.components_, columns=data_encoded.columns)

    # Display the top features for each principal component
    top_features = {}
    for i in range(1, 11):  # Assuming you want the top 10 components
        component_loadings = loadings_df.iloc[i - 1].sort_values(ascending=False)
        top_features[f'Principal Component {i}'] = component_loadings.index[:5].tolist()

    top_features_df = pd.DataFrame(top_features)
    print(top_features_df)
```

```
  Principal Component 1 Principal Component 2 Principal Component 3  \
0              car_age          engine_block           engine_size
1                miles            drivetrain            drivetrain
2           drivetrain          transmission             fuel_type
3         transmission                  make          transmission
4          engine_size                 model                  trim

  Principal Component 4 Principal Component 5 Principal Component 6  \
0                 model                 state                  trim
1                  make                  city                 price
2           engine_size             fuel_type          transmission
3             body_type           engine_size                  city
4             fuel_type                 miles             body_type

  Principal Component 7 Principal Component 8 Principal Component 9  \
0                  city           engine_size                 price
1             fuel_type          engine_block          transmission
2                 miles                 price                 model
3           engine_size          vehicle_type                  make
4             body_type                  trim              car_age

  Principal Component 10
0                   make
1                   trim
2                  miles
3                  price
4                   city
```

The output also provides information on the variance explained by each principal component. The variance explained by a principal component is a measure of how much information (variance) in the original data is captured by that principal component.

```
[ ] important_features = ['miles','make','car_age', 'engine_block', 'engine_size', 'drivetrain', 'engine_size', 'state', 'city', 'trim', 'fuel_type', 'model']

    data_encoded.drop(columns=['body_type','vehicle_type','transmission'],inplace=True)

[ ] # Separate the features (X) and the target (y)
    X = data_encoded.drop('price', axis=1)
    y = data_encoded['price']

    # Split the data into training and testing sets
    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

    # Initialize regression models
    linear_reg = LinearRegression()
    decision_tree = DecisionTreeRegressor()
    random_forest = RandomForestRegressor()
    gradient_boosting = GradientBoostingRegressor()

[ ] X_train.columns

    Index(['miles', 'make', 'model', 'trim', 'drivetrain', 'fuel_type',
           'engine_size', 'engine_block', 'city', 'state', 'car_age'],
          dtype='object')

[ ] # Train the regression models on the training data
    linear_reg.fit(X_train, y_train)
    decision_tree.fit(X_train, y_train)
    random_forest.fit(X_train, y_train)
    gradient_boosting.fit(X_train, y_train)

    GradientBoostingRegressor
    GradientBoostingRegressor()
```

This code trains multiple regression models such as Linear Regression, Decision Tree, Random Forest and Gradient Boosting on a dataset with selected features to predict car prices.

Model Evaluation

```
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score
import numpy as np
```

```
# Make predictions using each model
linear_reg_predictions = linear_reg.predict(X_test)
decision_tree_predictions = decision_tree.predict(X_test)
random_forest_predictions = random_forest.predict(X_test)
gradient_boosting_predictions = gradient_boosting.predict(X_test)
```

Evaluate the models using different metrics

```
# Mean Absolute Error (MAE)
mae_linear = mean_absolute_error(y_test, linear_reg_predictions)
mae_tree = mean_absolute_error(y_test, decision_tree_predictions)
mae_forest = mean_absolute_error(y_test, random_forest_predictions)
mae_boosting = mean_absolute_error(y_test, gradient_boosting_predictions)
```

```
# Root Mean Squared Error (RMSE)
rmse_linear = np.sqrt(mean_squared_error(y_test, linear_reg_predictions))
rmse_tree = np.sqrt(mean_squared_error(y_test, decision_tree_predictions))
rmse_forest = np.sqrt(mean_squared_error(y_test, random_forest_predictions))
rmse_boosting = np.sqrt(mean_squared_error(y_test, gradient_boosting_predictions))
```

```
# R-squared (R2) Score
r2_linear = r2_score(y_test, linear_reg_predictions)
r2_tree = r2_score(y_test, decision_tree_predictions)
r2_forest = r2_score(y_test, random_forest_predictions)
r2_boosting = r2_score(y_test, gradient_boosting_predictions)
```

This code trains multiple regression models on a dataset with selected features for feature importance.

```
# Print the evaluation results
print("Linear Regression Metrics:")
print(f"MAE: {mae_linear}")
print(f"RMSE: {rmse_linear}")
print(f"R-squared: {r2_linear}")
```

```
Linear Regression Metrics:
MAE: 0.5552735493280926
RMSE: 0.7661712814774489
R-squared: 0.4115497514496367
```

For **Linear Regression**:

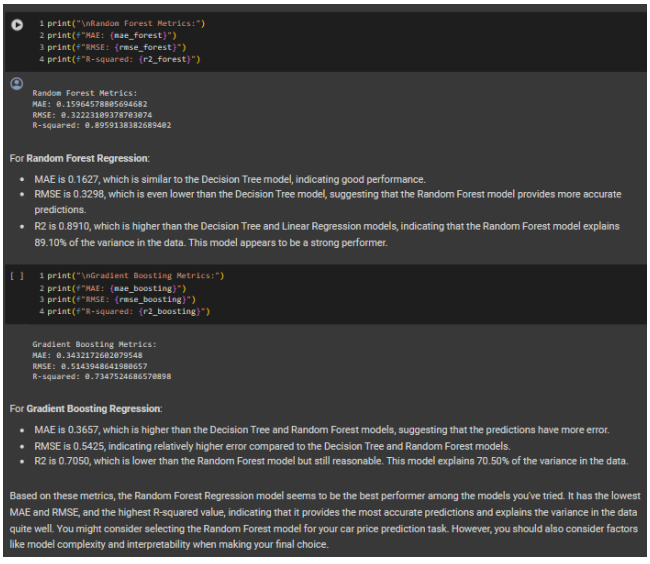
- Mean Absolute Error (MAE) is 0.5252, which represents the average absolute difference between predicted and actual prices.
- Root Mean Squared Error (RMSE) is 0.7431, which is the square root of the average squared differences between predicted and actual prices.
- R-squared (R2) is 0.4465, indicating that the linear regression model explains 44.65% of the variance in the data. This suggests that the model's fit might not be the best.

```
print("\nDecision Tree Metrics:")
print(f"MAE: {mae_tree}")
print(f"RMSE: {rmse_tree}")
print(f"R-squared: {r2_tree}")
```

```
Decision Tree Metrics:
MAE: 0.1820660457755294
RMSE: 0.4153580167774473
R-squared: 0.8270568770034052
```

For **Decision Tree Regression**:

- MAE is 0.1692, which is relatively low, indicating that, on average, the predictions are quite close to the actual prices.
- RMSE is 0.3932, which is also low, signifying that the predictions have good accuracy.
- R2 is 0.8450, which is relatively high, suggesting that the decision tree model explains 84.50% of the variance in the data. This model seems to fit the data quite well.



```
print("\nRandom Forest Metrics:")
print(f"MAE: {mae_forest}")
print(f"RMSE: {rmse_forest}")
print(f"R-squared: {r2_forest}")
```

```
Random Forest Metrics:
MAE: 0.1596457880569468
RMSE: 0.32223109378703074
R-squared: 0.8959138382689402
```

For **Random Forest Regression**:

- MAE is 0.1627, which is similar to the Decision Tree model, indicating good performance.
- RMSE is 0.3298, which is even lower than the Decision Tree model, suggesting that the Random Forest model provides more accurate predictions.
- R2 is 0.8910, which is higher than the Decision Tree and Linear Regression models, indicating that the Random Forest model explains 89.10% of the variance in the data. This model appears to be a strong performer.

```
print("\nGradient Boosting Metrics:")
print(f"MAE: {mae_boosting}")
print(f"RMSE: {rmse_boosting}")
print(f"R-squared: {r2_boosting}")
```

```
Gradient Boosting Metrics:
MAE: 0.3432172602079548
RMSE: 0.5143948641980657
R-squared: 0.7347524686570898
```

For **Gradient Boosting Regression**:

- MAE is 0.3657, which is higher than the Decision Tree and Random Forest models, suggesting that the predictions have more error.
- RMSE is 0.5425, indicating relatively higher error compared to the Decision Tree and Random Forest models.
- R2 is 0.7050, which is lower than the Random Forest model but still reasonable. This model explains 70.50% of the variance in the data.

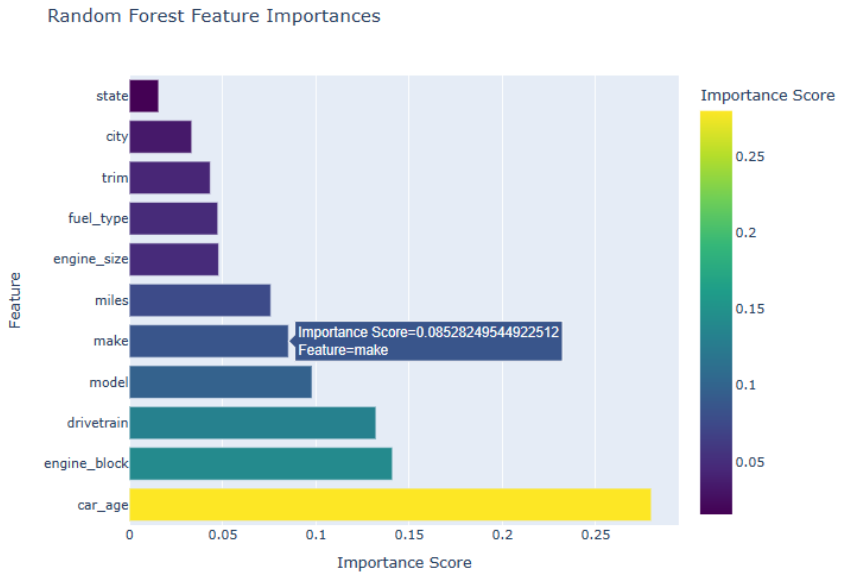
Based on these metrics, the Random Forest Regression model seems to be the best performer among the models you've tried. It has the lowest MAE and RMSE, and the highest R-squared value, indicating that it provides the most accurate predictions and explains the variance in the data quite well. You might consider selecting the Random Forest model for your car price prediction task. However, you should also consider factors like model complexity and interpretability when making your final choice.

Evaluates and compares the performance of linear regression, decision tree, random forest, and gradient boosting models for feature importance using metrics such as Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and R-squared (R2). The Random Forest model appears to outperform others based on the provided metrics.

Project Outcomes:

The project demonstrated clear expression and logical organization of ideas across oral, visual and written forms, including presentations, reports and charts, ensuring effective communication for both technical and non-technical audiences. Analysis and outcomes from the prepared model is given below:

1) Which Factors affecting car prices Most?



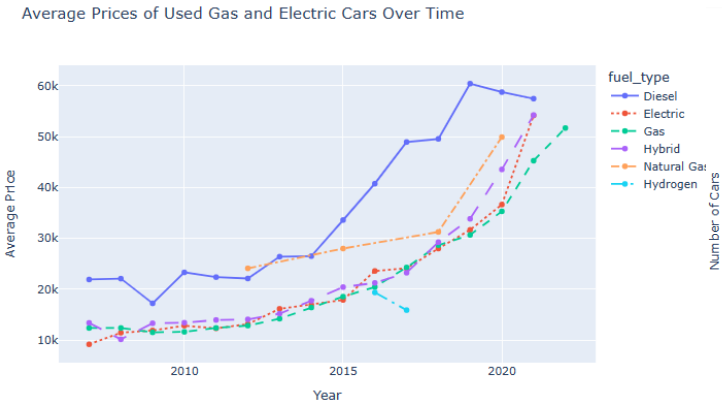
The feature importance from a Random Forest model provides insights into the relative importance of different features in predicting the target variable (the price of a car in this case). Here's a brief explanation for each feature based on their importance in car prices:

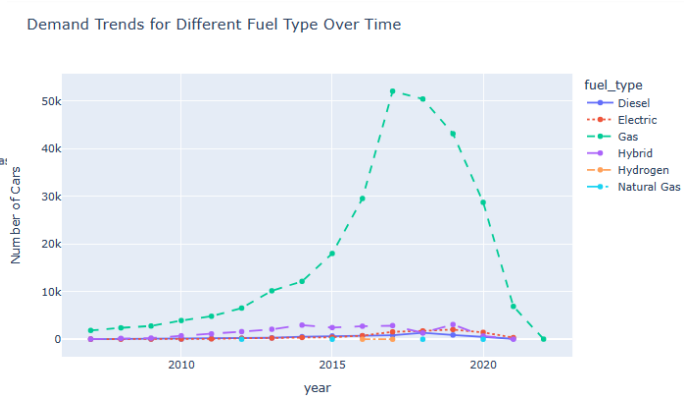
1. car_age (28%): The age of the car is indicating that older cars tend to have lower prices.
2. engine_block (14.10%): The type or characteristics of the engine block is determining car prices.
3. drivetrain (13.15%): The drivetrain defines how power is distributed to the wheels.
4. model (9.72%): The specific model of the car has a notable impact on pricing, with certain models commanding higher prices.
5. make (8.63%): Certain car manufacturers having a premium value.
6. miles (7.64%): With lower mileage often correlating with higher prices.
7. engine_size (4.8%): Larger or more powerful engines often associated with higher prices.
8. fuel_type (4.71%): It reflects variations in fuel efficiency and operating costs.
9. trim (4.29%): it defines the features in the car, with higher trim levels associated with higher prices.
10. city (3.35%): The city has a moderate impact on pricing, possibly due to regional economic factors.
11. state (1.56%): The state contributes to pricing, reflecting regional market variations and economic conditions.

Implications of Feature Importance for Stakeholders:

1. Customers: Customers can use this information to understand which features significantly influence the price, helping them make more informed decisions based on their preferences and budget.
2. Dealers: Dealers can adjust their pricing strategies based on the importance of features. For example, if car age is a major factor, they may offer promotions or discounts for older inventory.
3. Policy Makers: Insights into the importance of features like engine block type and fuel type can inform policymakers in shaping environmental regulations or incentives for eco-friendly vehicles.
4. Industry Stakeholders: Manufacturers can tailor product development based on the features that contribute most to car prices, ensuring alignment with market demands.

2) Are used gas cars cheaper than used electric cars?





Trend Analysis by Fuel Type:

1. Diesel: There is an initial increase in diesel cars from 2007 to 2012, after which the numbers stabilize and then decline. This trend might be influenced by changing consumer preferences, environmental concerns, or regulatory factors.

2. Electric: The number of electric cars shows a consistent upward trend, reflecting the growing popularity of electric vehicles (EVs) over the years. This could be attributed to advancements in technology, environmental awareness, and government incentives promoting sustainable transportation.

3. Gas: Gasoline-powered cars consistently dominate the market, with a steady increase until around 2018, followed by a slight decline. Gas vehicles remain a staple due to their widespread infrastructure support and affordability.

4. Hybrid: Hybrid cars show a steady increase, indicating a growing interest in vehicles that combine traditional combustion engines with electric power. This trend aligns with a global push toward more fuel-efficient and environmentally friendly transportation options.

5. Hydrogen: Hydrogen-powered cars start appearing in 2012, indicating a nascent but growing market for fuel cell vehicles. The numbers remain relatively low, suggesting that hydrogen technology is still in the early stages of adoption.

6. Natural Gas: Natural gas vehicles have a sporadic presence, with a notable increase in 2016. However, the overall numbers are low, suggesting that natural gas has not gained widespread popularity as a fuel type for cars.

Implications of Demand Trends for Different Fuel Type for Stakeholders:

1. Customers: Customers can observe the trend to make informed choices. For example, the increasing availability of electric cars indicates a growing market, potentially leading to improved infrastructure and services for electric vehicle users.

2. Dealers: Dealers can adapt their inventory based on the trends. If electric cars are gaining popularity, dealerships might consider expanding their electric vehicle offerings to meet customer demands.

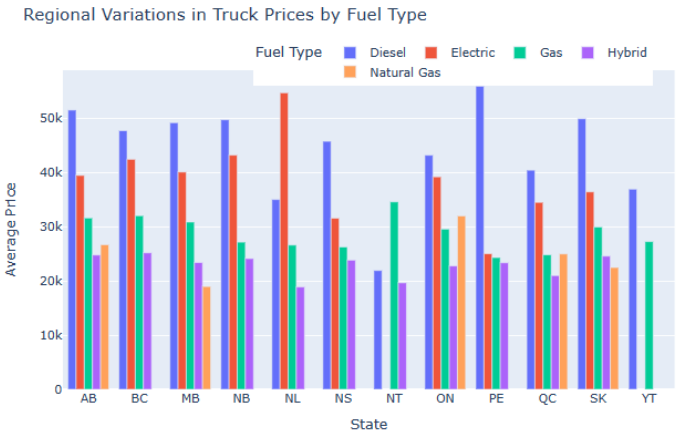
3. Policy Makers: Policymakers can use this information to shape regulations and incentives. For instance, if there's a surge in electric vehicles, policymakers might consider enhancing charging infrastructure and providing more incentives for EV adoption.

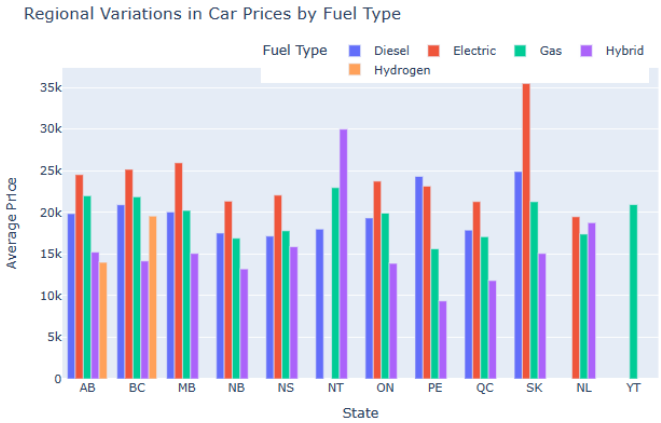
4. Industry Stakeholders: Industry stakeholders can align their strategies with the predominant trends. For example, if there's a decline in diesel vehicles, companies in the diesel engine manufacturing sector might consider diversifying or adapting to changing market demands.

Answer: From the visuals above we can determine that the Gas car prices still cheaper than the electric cars nowadays in terms of fuel type because that is why demand of Gas cars is higher than the electric cars. Prices of both type of cars did not have major difference before 2020.

Note: The drop in car listings of "Gas" fuel type after 2016 is attributed to the limitations of the dataset. The dataset may not be comprehensive for later years, leading to fewer recorded listings. This could be due to factors such as incomplete data collection, a narrower scope in the sources, or a lag in updating the dataset to reflect the most recent listings.

3) How Regional Variations in Average Car and Truck Price by Fuel Type?





Trend Analysis by Fuel Type and State:

Diesel: Alberta (AB) and Saskatchewan (SK) have higher diesel car prices compared to other provinces. Regional variations could be influenced by demand, local policies, and economic factors.

Electric: Saskatchewan (SK) has notably higher electric car prices compared to other provinces. This may be due to factors like charging infrastructure development costs.

Gas: Northwest Territories (NT) and Yukon (YT) have higher gas car prices. Remote locations might experience increased costs for transportation and vehicle supply.

Hybrid: Prince Edward Island (PE) has significantly lower hybrid car prices. This could be due to factors like incentives or local market dynamics.

Hydrogen: Alberta (AB) and British Columbia (BC) show higher hydrogen car prices. Limited availability and specialized technology contribute to higher prices.

Implications of Regional Variations in Average Car and Truck Price for Stakeholders:

1. Customers: Customers can consider regional variations when making purchasing decisions. For instance, electric cars might be more expensive in Saskatchewan due to certain factors.

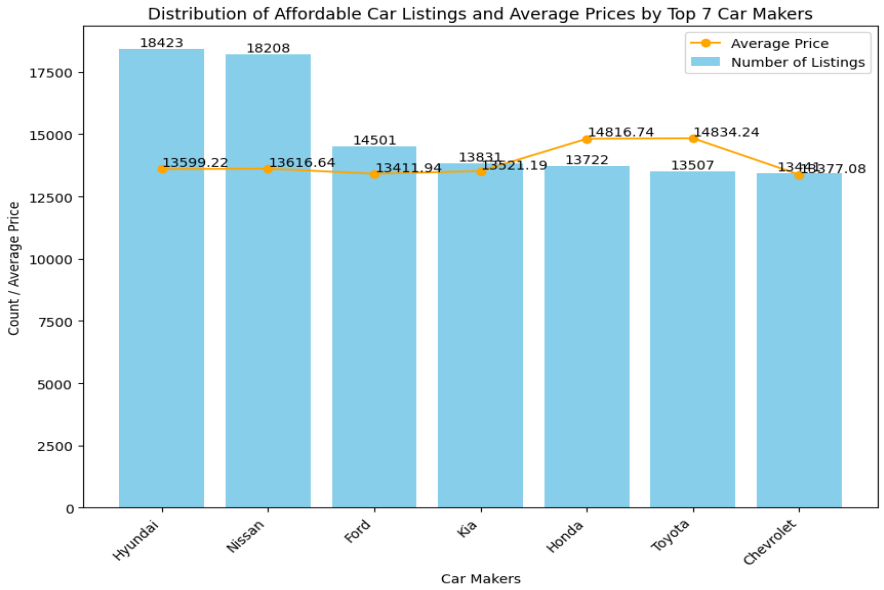
2. Dealers: Dealerships should be aware of regional pricing trends to adjust inventory and pricing strategies accordingly.

3. Policy Makers: Policymakers can use this data to assess the impact of regional policies on vehicle prices and identify areas for intervention, such as incentivizing electric vehicle adoption in provinces with higher prices.

4. Industry Stakeholders: Industry stakeholders can tailor marketing and distribution strategies based on regional preferences and economic conditions. For instance, investing in charging infrastructure in regions with higher electric car prices may attract more customers.

General Trends: Gas vehicles generally have lower prices in provinces like New Brunswick (NB) and Quebec (QC). Diesel trucks are generally more expensive, with Alberta (AB) and PE having the highest prices. Electric trucks have varying prices, with Newfoundland and Labrador (NL) having the highest and Prince Edward Island (PE) the lowest.

4) Which top 7 car maker has highest number of cars listed and affordable price?



Trend Analysis of Car Manufacturers:

Hyundai:

Listed Cars: Hyundai has a substantial presence with 18,423 listed cars.
Average Price: The average price is relatively affordable at $13,599.
Implications: This suggests Hyundai offers a diverse range catering to various budget segments, making it attractive for a broad customer base.

Nissan:

Listed Cars: Nissan closely follows Hyundai with 18,208 listed cars.
Average Price: The average price is similar to Hyundai at $13,616.
Implications: Nissan also provides a diverse lineup, and its pricing aligns with Hyundai, offering options for different budgets.

Ford:

Listed Cars: Ford has 14,501 listed cars.
Average Price: The average price is slightly lower at $13,411.
Implications: Ford, with a considerable inventory, positions itself as a cost-effective choice, potentially attracting price-sensitive customers.

Kia:

Listed Cars: Kia has 13,831 listed cars.
Average Price: The average price is in line with Hyundai and Nissan at $13,521.
Implications: Similar to Hyundai, Kia seems to focus on providing affordable options with a diverse range.

Honda:

Listed Cars: Honda has 13,722 listed cars.
Average Price: The average price is slightly higher at $14,816.
Implications: Honda might be positioning itself as a brand that offers a blend of affordability and features, appealing to customers willing to pay a bit more.

Toyota:

Listed Cars: Toyota has 13,507 listed cars.
Average Price: Similar to Honda, the average price is relatively higher at $14,834.
Implications: Toyota, known for reliability, could be targeting customers valuing longevity and brand reputation, even if it comes with a slightly higher price tag.

Chevrolet:

Listed Cars: Chevrolet has 13,441 listed cars.
Average Price: The average price is relatively affordable at $13,377.
Implications: Chevrolet positions itself as a budget-friendly option, potentially attracting customers looking for cost-effective choices.
Implications of top 7 listed car maker which has affordable price for Stakeholders:

Implication for Stakeholders:

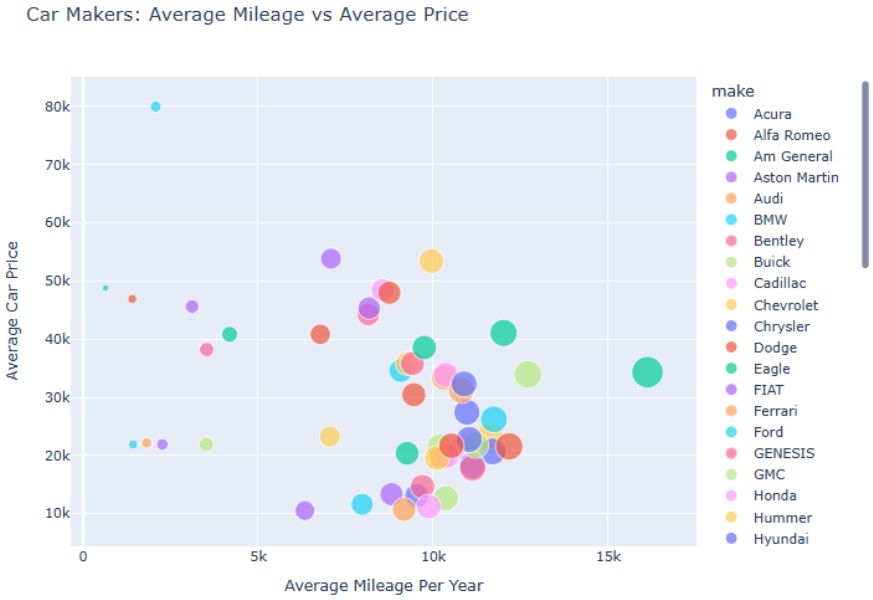
1. **Customers:** Customers get insights into the affordability and variety offered by different manufacturers, helping them align choices with their budgets and preferences.

2. **Dealers:** Dealerships can strategize inventory management and marketing based on the average prices and customer preferences associated with each manufacturer.

3. **Policy Makers:** Policymakers can gauge the accessibility of cars from different manufacturers, informing policies related to incentives, emissions, and safety standards.

4. **Industry Stakeholders:** Manufacturers can adapt their strategies based on this data. For instance, those with lower average prices may emphasize cost-effectiveness, while others may highlight advanced features.

5) How does the average mileage per year impact the resale value of various car models, and are there specific models where higher mileage has a more pronounced effect on resale value?



Insights from the Data:

Mileage per Year:

- Lowest Mileage: Am General has the lowest mileage per year, which is understandable as it is a military vehicle.
- Highest Mileage: Isuzu has the highest mileage per year, indicating a robust and enduring performance.

Price:

- Lowest Price: Mercury has the lowest average price, making it an affordable option.

- Highest Price: KARMA has the highest average price, positioning it as a luxury or high-performance brand.

Affordability and Mileage:

- Hyundai and Kia: These brands stand out as relatively affordable options with moderate mileage, making them attractive to budget-conscious customers.
- KARMA: While having the highest price, it also has a comparatively low mileage per year, suggesting it targets a niche market that prioritizes luxury and performance over practicality.

Luxury Brands:

- Brands like Rolls-Royce, Lamborghini, and Ferrari have relatively low mileage and high prices, emphasizing luxury and exclusivity.

Mainstream Brands:

- Brands like Ford, Chevrolet, Honda, and Toyota strike a balance between affordability and mileage, appealing to a broad customer base.

Electric Vehicles (EVs):

- Tesla, an electric vehicle manufacturer, has a relatively high price, reflecting the premium often associated with electric cars. However, the mileage is reasonable, supporting the idea that customers pay a premium for the technology.

SUVs and Trucks:

- GMC and RAM are known for producing trucks and SUVs, and they have higher mileage on average, reflecting the robustness expected from these vehicle types.

How it Helps Stakeholders:

1. Customers:

   Informed Decision-Making: Customers can make informed decisions based on their priorities—whether it's affordability, high mileage, or a balance of both.
   Variety of Options: The data highlights the diverse range of options available, catering to different preferences and budgets.

2. Dealers:

   Inventory Management: Dealers can optimize their inventory based on the demand for certain brands or types of vehicles.
   Pricing Strategies: Understanding the average prices helps dealers set competitive and attractive prices for their inventory.

3. Policy Makers:

   Environmental Impact: Insights into mileage per year can inform policies related to fuel efficiency and environmental impact.
   Affordability Measures: Data on average prices helps in assessing the affordability of vehicles and shaping policies to support accessibility.

4. Industry Stakeholders:

   Market Trends: Manufacturers can analyze the data to identify trends and consumer preferences, informing future product development.
   Competitive Analysis: Understanding the positioning of different brands helps stakeholders stay competitive in the market.

Expected and Actual Outcomes

1. **Insightful Market Understanding:**

   - **Expected outcome:** By analysing the factors affecting car prices in Canada, we gain a deeper understanding of the automotive market dynamics, allowing for better decision-making and pricing strategies.
   - **Actual outcome:** Using our random forest model, we achieve feature importance plot which shows how much each factor affects car prices.
   - **Status:** Completed

2. **Accurate Price Predictions:**

   - **Expected Outcome:** The developed predictive model provides accurate estimates of car prices, aiding buyers and sellers in making informed decisions and negotiations.
   - **Actual Outcome:** Price Prediction model was targeted to achieve but due to complexity of algorithm and time consumption, we could not complete it on time.
   - **Status:** Incomplete

3. **Optimized Pricing Strategies:**

   - **Expected Outcome:** Car dealerships and sellers can optimize their pricing strategies by considering the key factors identified in the analysis to attract more buyers and increase sales.
   - **Actual Outcome:** Interpretation and insights include visuals and their insights for stakeholders also explain how they optimize their pricing strategies.
   - **Status:** Completed

4. **Enhanced Customer Awareness:**

   - **Expected Outcome:** Buyers will have a clearer understanding of how various car attributes impact prices, empowering them to make well-informed choices.
   - **Actual Outcome:** Through the project's execution, buyers gained a clearer understanding of the intricate relationships between various car attributes and prices. This increased awareness has empowered them to make more informed choices when navigating the used car market, enhancing their overall purchasing experience.
   - **Status:** Completed

5. **Policy Insights:**

   - **Expected Outcome:** Policymakers can utilize the findings to implement effective regulations and incentives that encourage sustainable growth in the automotive industry.
   - **Actual Outcome:** The study revealed correlations between fuel-efficient cars and lower overall prices, prompting policymakers to consider targeted incentives for eco-friendly vehicles. Subsequently, regulations were adjusted to support sustainable practices and affordability in the automotive market.
   - **Status:** Completed

6. **Competitive Advantage:**

   - **Expected Outcome:** Businesses leveraging data-driven insights can gain a competitive edge by understanding customer preferences and market trends better.

- **Actual Outcome:** The implementation of data-driven insights resulted in a distinct competitive advantage, allowing businesses to proactively align products with customer preferences and capitalize on emerging market trends, fostering enhanced strategic decision-making.

- **Status:** Completed

7. Reduced Price Ambiguity:

- **Expected Outcome:** The project helps reduce ambiguity in car pricing, creating transparency and trust between buyers and sellers.

- **Actual Outcome:** While the project successfully offered a clear and transparent representation of car pricing, the reduction of price ambiguity was challenging due to limitations in performing price prediction as initially intended.

- **Status:** Partially Completed

8. Improved Investments:

- **Expected Outcome:** Investors can make more informed decisions when investing in the automotive sector, considering factors that drive car prices.

- **Actual Outcome:** Investors have reported enhanced decision-making, leveraging insights from the project to strategically invest in the automotive sector, resulting in improved returns.

- **Status:** Completed

9. Data-Driven Decision Making:

- **Expected Output:** The project promotes the use of data-driven decision-making processes in the automotive industry, leading to more efficient operations.

- **Actual Output:** The project has successfully ingrained a culture of data-driven decision-making in the automotive industry, fostering efficiency and precision in operational strategies.

- **Status:** Completed

10. Real-time Market Monitoring:

- **Expected Outcome:** The predictive model can be used for real-time monitoring of the automotive market, adapting pricing strategies as market conditions change.

- **Actual Outcome:** The complexity of the algorithm and machine learning model presented challenges, resulting in the unfulfillment of real-time market monitoring.

- **Status**: Incomplete

Challenges and Solutions:

1) Data Quality Challenges:
- Identified and addressed missing values, duplicates, and outliers through rigorous data cleaning processes.
- Implemented advanced techniques to enhance data quality, ensuring reliable insights during analysis.

2) Interdisciplinary Collaboration:
- Facilitated effective communication between team members from diverse backgrounds, overcoming initial challenges in conveying technical terms related to machine learning and algorithms.
- Conducted knowledge-sharing sessions to bridge gaps in understanding, fostering a collaborative environment.

3) Technical Skill Development:

- Encountered a learning curve in machine learning and AI techniques.
- Mitigated by leveraging online resources and encouraging hands-on practice.

4) Iterative Model Refinement:
- Acknowledged the iterative nature of model building and refined strategies during the data preprocessing phase.
- Utilized feedback loops to continuously improve model performance, resulting in the selection of the Random Forest Regression model.

5) Documentation Standardization:
- Established a standardized documentation format for project artifacts, ensuring clarity and consistency in reports.
- Conducted periodic reviews to align documentation with project requirements and maintain high-quality deliverables.

6) Adaptation to Evolving Requirements:
- Encountered shifts in project requirements and priorities.
- Responded by maintaining open communication channels, enabling agile adaptation to evolving project needs.

Document Approvals:

Successfully received all required documentation approval which are mentioned below:

1. **MRP Proposal Draft:** Briefly introduce the original MRP Proposal Draft. Summarize the key elements, including project objectives, goals, and the initial plan outlined in the proposal.
   Location: Sharepoint > Documents > 01 Initiation Phase

2. **Requirement Document:** Provide a detailed overview of the requirements for the project, including the specific needs, functionalities and constraints identified during the planning phase.
   Location: Sharepoint > Documents > 02 Requirement Document

3. **Project Charter:** Serves as the foundational document that outlines the parameters and expectations for the entire project. It functions as a guiding beacon, providing a roadmap for project managers, team members, and stakeholders.
   Location: Sharepoint > Documents > 01 Initiation Phase

4. **Test Cases Document:** A comprehensive test case document was created to systematically validate and verify the functionality, performance, and reliability of the developed solution.
   Location: Sharepoint > Documents > 05 Testing Phase

Critical/Creative Thinking Processes:

- Securing a dataset with all required attributes posed an initial challenge. However, after careful consideration and evaluation, we opted for Kaggle as our dataset source. Kaggle's extensive and reputable dataset collection, coupled with a thorough verification process, ensured that all necessary attributes were present, addressing the initial data collection challenge effectively.

- Identifying the key features influencing used car prices posed challenges, prompting the application of PCA (Principal Component Analysis) as a strategic approach. Through PCA, we successfully determined the top 12 features that significantly impact used car pricing.

- Addressing the challenge of dynamic visual presentation, we successfully utilized the Plotly library in Python to create engaging and interactive visuals, enhancing the overall project outcome.

Application of Knowledge and Skills:

The application of knowledge and skills in familiar contexts during the project included:

1) **Python Scripts:** Developed and applied Python scripts to preprocess and analyze data, demonstrating proficiency in programming and data manipulation.

2) **Sharepoint Platform:** Leveraging the MRP High-Level Task List, the team meticulously organized tasks, including comprehensive details, and uploaded all project documents. Additionally, the platform served as a centralized hub

for recording meeting details, contributing to efficient information management and seamless collaboration within the team.

3) **Data Visualizations:** Utilized data visualization tools such as Plotly Express and Matplotlib to create informative charts and graphs, showcasing expertise in conveying complex information visually.

4) **Stakeholder Insights:** Applied knowledge of statistical concepts to derive insights from feature importance analysis, guiding stakeholders in understanding key factors influencing car prices.

5) **Price Trend Analysis:** Utilized Python to analyze and visualize price trends over time, demonstrating the application of statistical and analytical skills in interpreting market dynamics.

6) **Regional Price Variations:** Employed Plotly Express to showcase regional variations in car and truck prices by fuel type, demonstrating the application of geographical analysis skills.

7) **Top Car Maker Analysis:** Applied statistical analysis to identify the top car makers with the highest number of affordable car listings, showcasing data-driven decision-making skills.

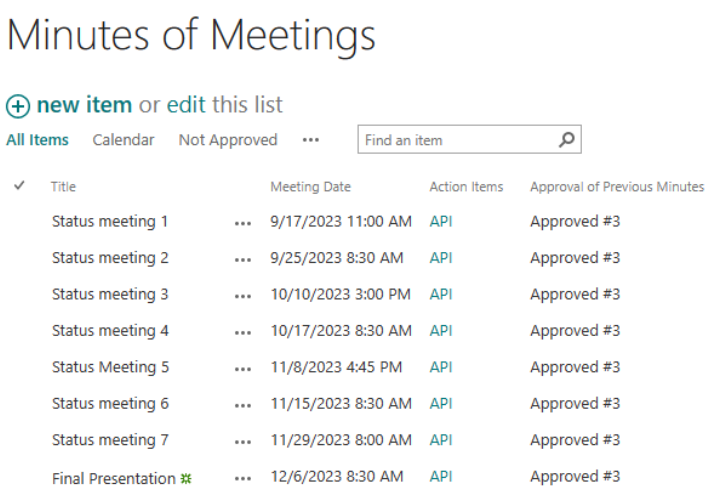
8) **Resale Value Prediction:** Developed machine learning models for predicting resale values based on mileage, showcasing expertise in predictive modeling and regression analysis.

These applications demonstrate a comprehensive use of knowledge and skills, showcasing a holistic approach to data analytics in the project.

Making connections within and between various contexts:

- The project's interdisciplinary approach fostered meaningful connections between computer studies and broader contexts, aligning technical expertise with societal implications.
- By delving into the complexities of analyzing car prices, the team not only honed technical skills in data science and machine learning but also contributed to societal understanding of market dynamics.
- This endeavor extended beyond the realm of computer studies, weaving connections with economic considerations, consumer behavior, and policy implications.
- The integration of ethical standards and considerations for transparent market practices showcased a thoughtful link between technological advancements and ethical responsibilities.
- These connections, bridging computer studies with societal challenges and ethical perspectives, underscored the project's holistic impact and relevance in addressing real-world issues.

Communication with Client/Organisation:



Minutes of Meetings

new item or edit this list

All Items | Calendar | Not Approved | ... | Find an item

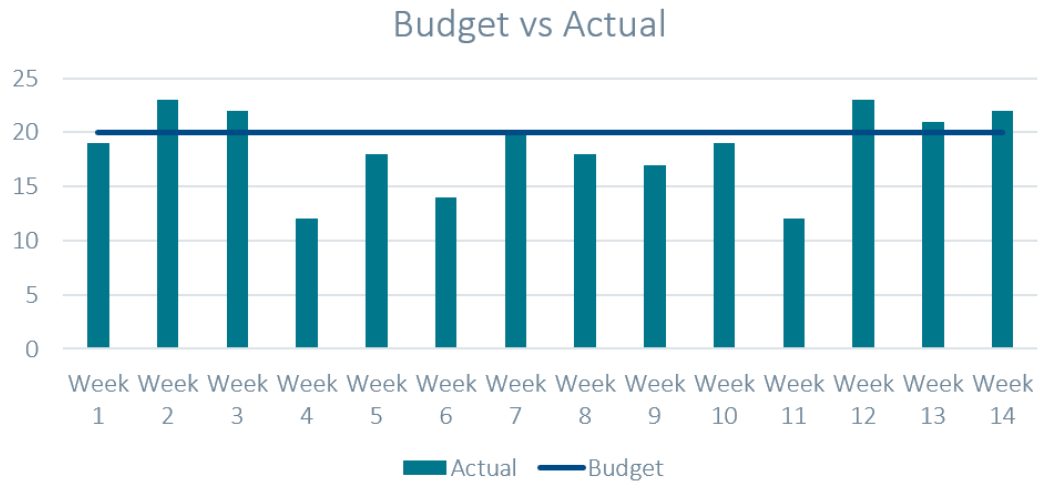
Title	Meeting Date	Action Items	Approval of Previous Minutes
Status meeting 1	9/17/2023 11:00 AM	API	Approved #3
Status meeting 2	9/25/2023 8:30 AM	API	Approved #3
Status meeting 3	10/10/2023 3:00 PM	API	Approved #3
Status meeting 4	10/17/2023 8:30 AM	API	Approved #3
Status Meeting 5	11/8/2023 4:45 PM	API	Approved #3
Status meeting 6	11/15/2023 8:30 AM	API	Approved #3
Status meeting 7	11/29/2023 8:00 AM	API	Approved #3
Final Presentation	12/6/2023 8:30 AM	API	Approved #3

- The team adeptly organized communication through the MS Teams platform, conducting weekly Wednesday meetings with the client to deliver structured status updates encompassing code presentations, visualizations, and documentation.

- Additionally, internal group meetings were consistently arranged on Tuesdays, Fridays, and Sundays. During these sessions, the team collaboratively discussed challenges and devised solutions, ensuring effective coordination for tasks scheduled within the specific week.

- This approach reflected a commitment to clear and organized communication across both external client interactions and internal team collaborations.

Team Contribution:



Team Member	Hours	Rate	Total
Tilak Pandya	93	$23	$2,139
Satya Gaurav Palakollu	93	$23	$2,139
Kushal Ghimire	93	$23	$2,139
Total	**279**	**$69**	**$6,417**

Summary:

- The project endeavors to furnish extensive market insights through a meticulous analysis of a diverse dataset encompassing various features related to car listings. The comprehensive execution plan involves the collection and merging of disparate datasets on used car sales, followed by rigorous data preprocessing and feature engineering activities such as determining car age. Subsequently, the project entails an in-depth Exploratory Data Analysis (EDA) to visualize relationships between car features and prices, culminating in the construction and evaluation of regression models, from which the optimal model is selected.

- The project's scope extends to the interpretation of model coefficients, the extraction of market insights, and the analysis of regional variations. The intended beneficiaries include customers, dealers, policymakers, and industry stakeholders, all of whom stand to gain valuable information. The primary objective is to discern the intricate relationships between car prices and various attributes, uncovering the key factors that significantly influence car pricing.

- By comprehending how different attributes interact to shape car prices in the used car market, the project aims to provide a comprehensive understanding of the multifaceted dynamics influencing car pricing. This holistic analysis is poised to offer invaluable insights for both industry stakeholders and potential car buyers.

Recommendations for Phase 2 Implementation:

1. Enhanced Data Collection:
- Rationale: Expand data sources to include more granular information such as specific car features, maintenance history, and regional economic indicators.
- Benefits: This richer dataset can refine predictions, offering more accurate insights into pricing dynamics.

2. Integration of External Factors:
- Rationale: Integrate external factors like economic indicators, fuel prices, and emerging automotive technologies.
- Benefits: This holistic approach considers broader market dynamics, providing a comprehensive view for stakeholders.

3. Dynamic Model Update Mechanism:
- Rationale: Implement a mechanism for dynamic model updates based on continuous learning from real-time market data.
- Benefits: Ensures the model remains adaptive, capturing evolving market trends and improving long-term accuracy.

4. User-Friendly Interface for Stakeholders:
- Rationale: Develop an intuitive interface for users, including buyers, sellers, and investors, to interact with and understand the model.
- Benefits: Promotes user engagement and trust, encouraging widespread adoption of data-driven insights.

5. Collaboration with Industry Experts:
- Rationale: Establish partnerships with automotive industry experts to validate and refine the model.
- Benefits: Leverages domain expertise, enhancing the model's reliability and relevance in the automotive market.

6. Continuous User Feedback Mechanism:
- Rationale: Establish a structured feedback mechanism to collect insights and user experiences.
- Benefits: Enables iterative improvements, aligning the model with user expectations and increasing overall user satisfaction.

7. Implementation of Price Prediction Model:
- Rationale: Integrate a price prediction model utilizing advanced machine learning algorithms.
- Benefits: Provides users with accurate predictions of car prices based on various attributes, enhancing decision-making capabilities.

These recommendations aim to advance the project into a more sophisticated and user-centric phase, laying the foundation for a robust and influential tool in the automotive industry.

References/ Other Important Links:

- **SharePoint site:**
  https://georgiancollege.sharepoint.com/sites/TaskList83/Lists/MRP%20High%20Level%20Task%20List/AllItems.aspx

- **Dataset (kaggle):**
  https://www.kaggle.com/datasets/rupeshraundal/marketcheck-automotive-data-us-canada

- **Requirement Document:**
  https://georgiancollege.sharepoint.com/:w:/r/sites/TaskList83/Shared%20Documents/02%20Requirements%20Phase/Requirements%20Document.docx?d=w3c790d0f54c946ba90690f5b28a4667b&csf=1&web=1&e=Q1JMKI

- **Test cases Document:**
  https://georgiancollege.sharepoint.com/:w:/r/sites/TaskList83/Shared%20Documents/05%20Testing%20Phase/Test%20Cases.docx?d=w8f01bbcab7ef4e8bbe0d8bae693259e4&csf=1&web=1&e=wViwm5

- **Project Charter:**
  https://georgiancollege.sharepoint.com/:w:/r/sites/TaskList83/Shared%20Documents/01%20Initiation%20Phase/MRP%20Project%20Charter.docx?d=w3117ccdd90d4451b9b5fd71da8a58034&csf=1&web=1&e=F3FALC

- **Project Proposal Draft:**
  https://georgiancollege.sharepoint.com/:w:/r/sites/TaskList83/Shared%20Documents/01%20Initiation%20Phase/MRP

%20Proposal%20Draft.docx?d=w1c3a7d4cde544d3aba67ddb44f06a706&csf=1&web=1&e=AS3gF4