ANWAY KAPOOR
ROHITH K GEORGE
MANAS TEWARI

CAR SALES DASHBOARD

BUSINESS INTELLIGENCE

RA2211027010014
RA2211027010044
RA2211027010062

Abstract

- Goal: To create a fully interactive and error-proof visualization dashboard for a car sales dataset.
- Purpose: Enable rapid Exploratory Data Analysis (EDA), allowing users to quickly identify trends, distributions, and relationships within the data without writing code.
- Key Features: Filtering by manufacturer/category, displaying key statistics, and generating various plot types (Bar Chart, Scatter Plot, Distribution Plot, Pie Chart).
- Technology: Streamlit (or similar web framework) for deployment, Python/Pandas for data handling, and Plotly/Matplotlib/Seaborn for visualization.

Dashboard Architecture & Tools

- Backend: Python (handling data manipulation and logic).
- Frontend/Deployment: Streamlit (for creating the interactive web application).
- Libraries:
- pandas (Data loading and manipulation)
- plotly or matplotlib/seaborn (For generating visualizations)
- streamlit (For the user interface and interactivity)
- User Flow:
- User selects Filters (e.g., Manufacturer).
- Dashboard dynamically updates Key Statistics and all plots.
- User explores different variable relationships.
- User can download filtered data for external analysis.

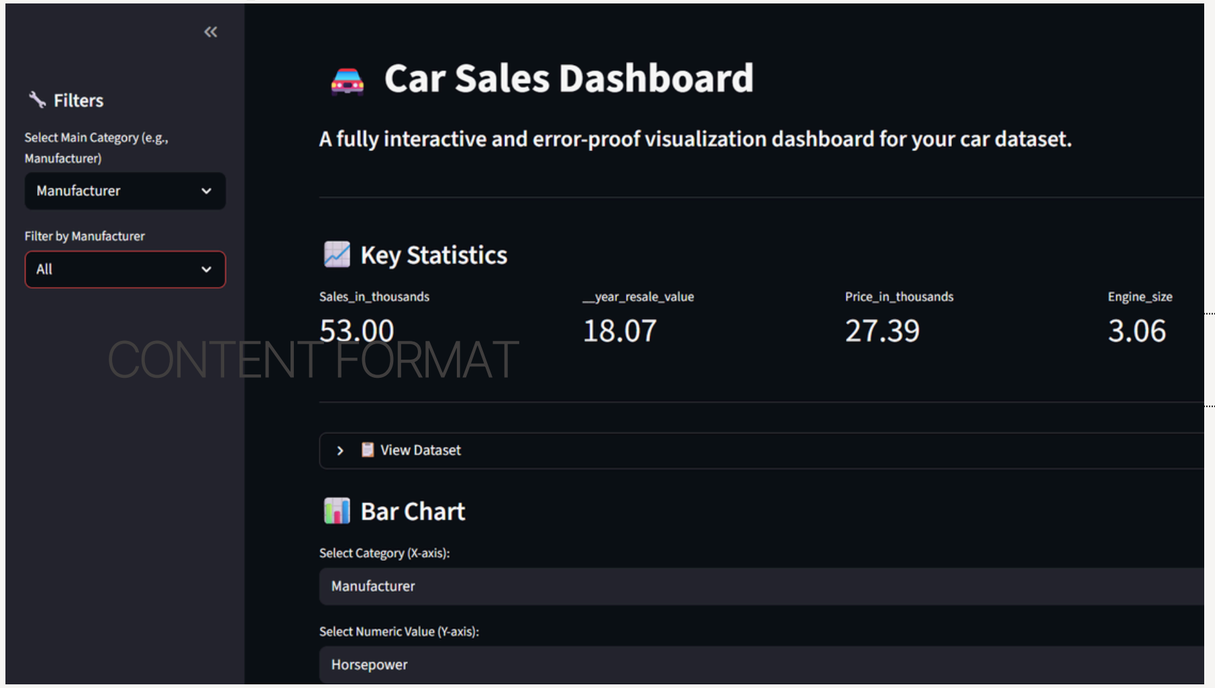
Dashboard Output

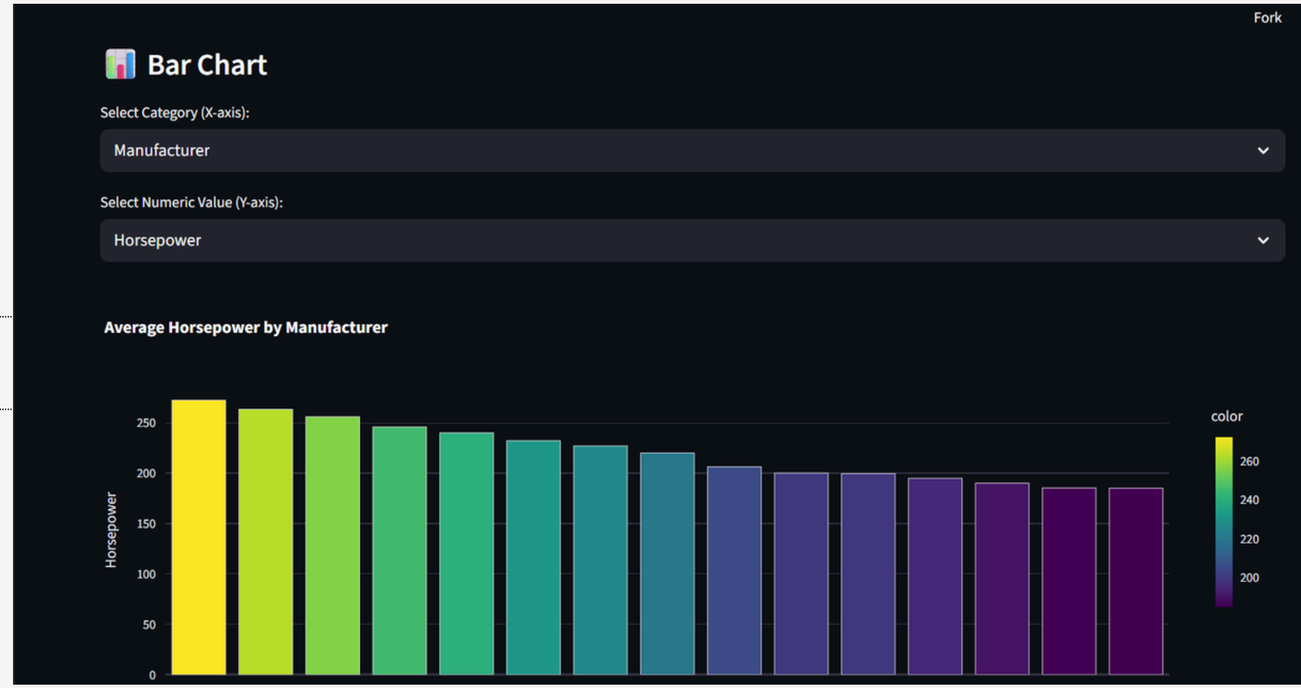
- Baseline Context: The Key Statistics panel provides overall averages for Sales (53.00K), Price (27.39K), and Resale Value (18.07K), establishing the market benchmark for the dataset.
- Market Skew: The Distribution Plot of sales reveals a heavy positive skew. This is a crucial finding: the market is dominated by a few high-volume vehicles, while the majority of models sell significantly fewer than 100,000 units.
- Performance Benchmarking: The Bar Chart quickly ranks manufacturers by metrics like Horsepower. We can see that performance-focused brands like Porsche hold the lead in average horsepower.
- Portfolio Recency: The Pie Chart gives insight into the recency of the product lineup by showing the distribution of the latest model launches, confirming if the portfolio is being refreshed consistently.

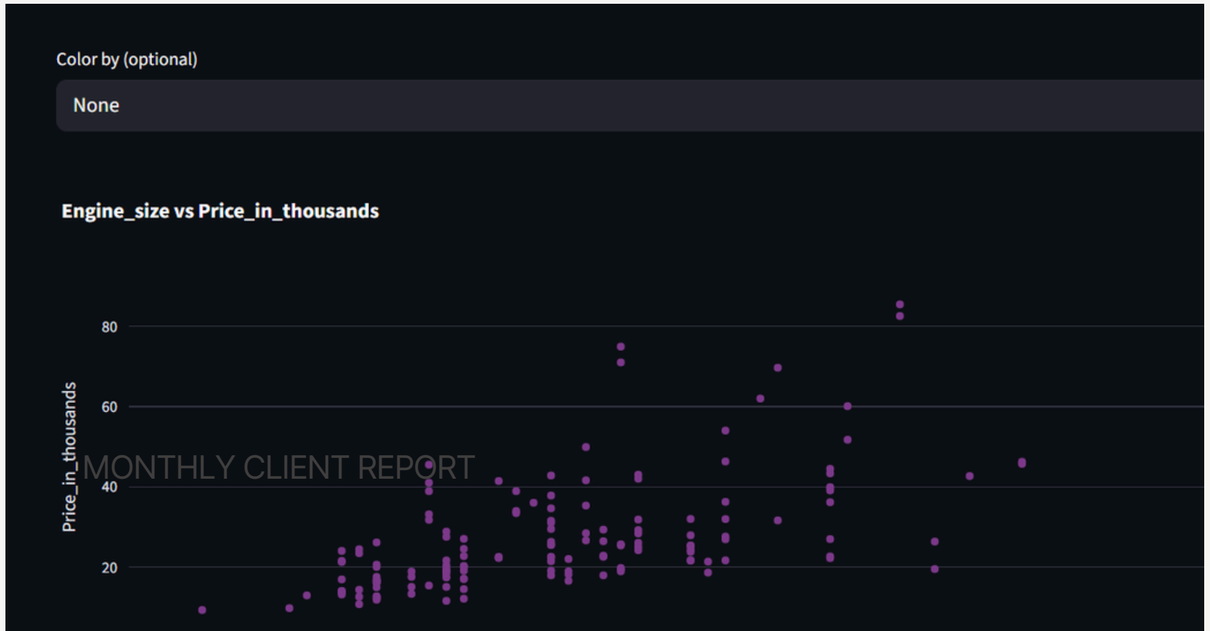
Data Analysis

- Price Drivers are Multifaceted: The Scatter Plot (Engine Size vs. Price) shows a correlation, but it's not perfect. While larger engines generally mean higher prices, the wide scatter suggests that brand positioning, luxury features, and vehicle type are equally strong drivers of the final price tag.
- Implication: Pricing models must account for intangible factors beyond simple physical specifications.
- Segment Isolation: The dynamic filtering capability is the key analytical tool. Analysts can instantaneously isolate and compare segments (e.g., filter for 'BMW' or 'Trucks'). This immediately reveals that segment's specific average sales, price, and distribution, contrasting it against the overall market benchmark.
- Optimizing Sales Strategy: Given the highly skewed sales distribution, business strategy must clearly define two tracks:

Snap Shots







Snap Shots

Code Blame 114 lines (97 loc) · 4.2 KB

```python
import streamlit as st
import pandas as pd
import plotly.express as px

# --- PAGE CONFIG ---
st.set_page_config(page_title="Car Sales Dashboard", page_icon="🚗", layout="wide")

# --- LOAD DATA ---
@st.cache_data
def load_data():
    df = pd.read_csv("Car_sales.csv")
    return df
df = load_data()

st.title("🚗 Car Sales Dashboard")
st.markdown("#### A fully interactive and error-proof visualization dashboard for your car dataset.")

st.markdown("---")

# --- AUTO-DETECT COLUMNS ---
numeric_cols = df.select_dtypes(include=["int64", "float64"]).columns.tolist()
categorical_cols = df.select_dtypes(include=["object"]).columns.tolist()

# --- SIDEBAR FILTERS ---
st.sidebar.header("🔧 Filters")
```



```python
st.markdown("---")

# --- DATA PREVIEW ---
with st.expander("📄 View Dataset"):
    st.dataframe(df)

# --- VISUALIZATION 1: BAR CHART ---
if len(categorical_cols) > 0 and len(numeric_cols) > 0:
    st.subheader("📊 Bar Chart")
    x_col = st.selectbox("Select Category (X-axis):", categorical_cols)
    y_col = st.selectbox("Select Numeric Value (Y-axis):", numeric_cols)
    bar_data = df.groupby(x_col)[y_col].mean().sort_values(ascending=False).head(15)
    fig_bar = px.bar(
        bar_data,
        x=bar_data.index,
        y=bar_data.values,
        title=f"Average {y_col} by {x_col}",
        labels={"x": x_col, "y": y_col},
        color=bar_data.values,
        color_continuous_scale="Viridis"
    )
    fig_bar.update_layout(template="plotly_white")
    st.plotly_chart(fig_bar, use_container_width=True)

# --- VISUALIZATION 2: SCATTER PLOT ---
if len(numeric_cols) >= 2:
    st.subheader("📉 Scatter Plot")
```

```python
        title=f"{scatter_x} vs {scatter_y}",
        color_discrete_sequence=px.colors.qualitative.Bold
    )
    fig_scatter.update_layout(template="plotly_white")
    st.plotly_chart(fig_scatter, use_container_width=True)

# --- VISUALIZATION 3: HISTOGRAM ---
if len(numeric_cols) > 0:
    st.subheader("📦 Distribution Plot")
    hist_col = st.selectbox("Select Numeric Column", numeric_cols, key="hist")
    fig_hist = px.histogram(df, x=hist_col, nbins=25, color_discrete_sequence=["#1f77b4"])
    fig_hist.update_layout(template="plotly_white", title=f"Distribution of {hist_col}")
    st.plotly_chart(fig_hist, use_container_width=True)

# --- VISUALIZATION 4: PIE CHART ---
if len(categorical_cols) > 0:
    st.subheader("🍩 Category Distribution")
    pie_col = st.selectbox("Select Categorical Column", categorical_cols, key="pie")
    pie_data = df[pie_col].value_counts().head(10)
    fig_pie = px.pie(
        values=pie_data.values,
        names=pie_data.index,
        title=f"Top {pie_col} Distribution",
        color_discrete_sequence=px.colors.qualitative.Pastel
```


Code Blame 114 lines (97 loc) · 4.2 KB Raw

```python
# --- VISUALIZATION 3: HISTOGRAM ---
if len(numeric_cols) > 0:
    st.subheader("📦 Distribution Plot")
    hist_col = st.selectbox("Select Numeric Column", numeric_cols, key="hist")
    fig_hist = px.histogram(df, x=hist_col, nbins=25, color_discrete_sequence=["#1f77b4"])
    fig_hist.update_layout(template="plotly_white", title=f"Distribution of {hist_col}")
    st.plotly_chart(fig_hist, use_container_width=True)

# --- VISUALIZATION 4: PIE CHART ---
if len(categorical_cols) > 0:
    st.subheader("🍩 Category Distribution")
    pie_col = st.selectbox("Select Categorical Column", categorical_cols, key="pie")
    pie_data = df[pie_col].value_counts().head(10)
    fig_pie = px.pie(
        values=pie_data.values,
        names=pie_data.index,
        title=f"Top {pie_col} Distribution",
        color_discrete_sequence=px.colors.qualitative.Pastel
    )
    st.plotly_chart(fig_pie, use_container_width=True)

# --- DOWNLOAD FILTERED DATA ---
```

Future Implementations

- Advanced Filtering: Implement multi-select filters (e.g., select Acura AND Audi sales simultaneously) or a date-range slider for the latest launch date.
- Advanced Analytics:
- Add a Regression Line to the Scatter Plot to quantify the relationship between variables (e.g., Price vs. Engine Size).
- Implement Clustering Analysis to automatically group similar car models based on a combination of features (e.g., Price, Horsepower, Sales).
- Prediction Feature: Integrate a Machine Learning Model (e.g., Linear Regression) to predict a car's Resale Value or Sales based on its specifications.
- More Context: Add hover tooltips on all plots to display specific model names and exact data points.

Conclusion

- Summary: The Car Sales Dashboard successfully provides a powerful, zero-code tool for exploring complex sales data.
- Impact: It has enabled analysts to quickly:
- Understand Market Distribution: The heavy right skew in sales highlights the importance of volume leaders.
- Analyze Price Drivers: Price is influenced by engine size but also heavily by brand positioning.
- Benchmark Competitors: Performance metrics like Horsepower can be quickly compared across manufacturers.
- Final Statement: This project demonstrates the value of interactive data visualization in transforming raw data into actionable business intelligence

Key Takeaways

- Market Skew is Critical: Sales are heavily skewed, meaning a few volume models drive the majority of market share. Strategic focus must prioritize these high-volume vehicles.
- Price is More Than Specs: The relationship between price and features (like Engine Size) is present but weak. Brand equity and premium features are the dominant drivers for high price points and margins.
- Dynamic Benchmarking: The dashboard enables instant comparison of any manufacturer or segment against the overall market baseline, facilitating rapid competitive analysis.
- Actionable & Accessible: The project successfully delivered a zero-code, interactive platform, making complex data analysis accessible to all decision-makers.

THANKYOU