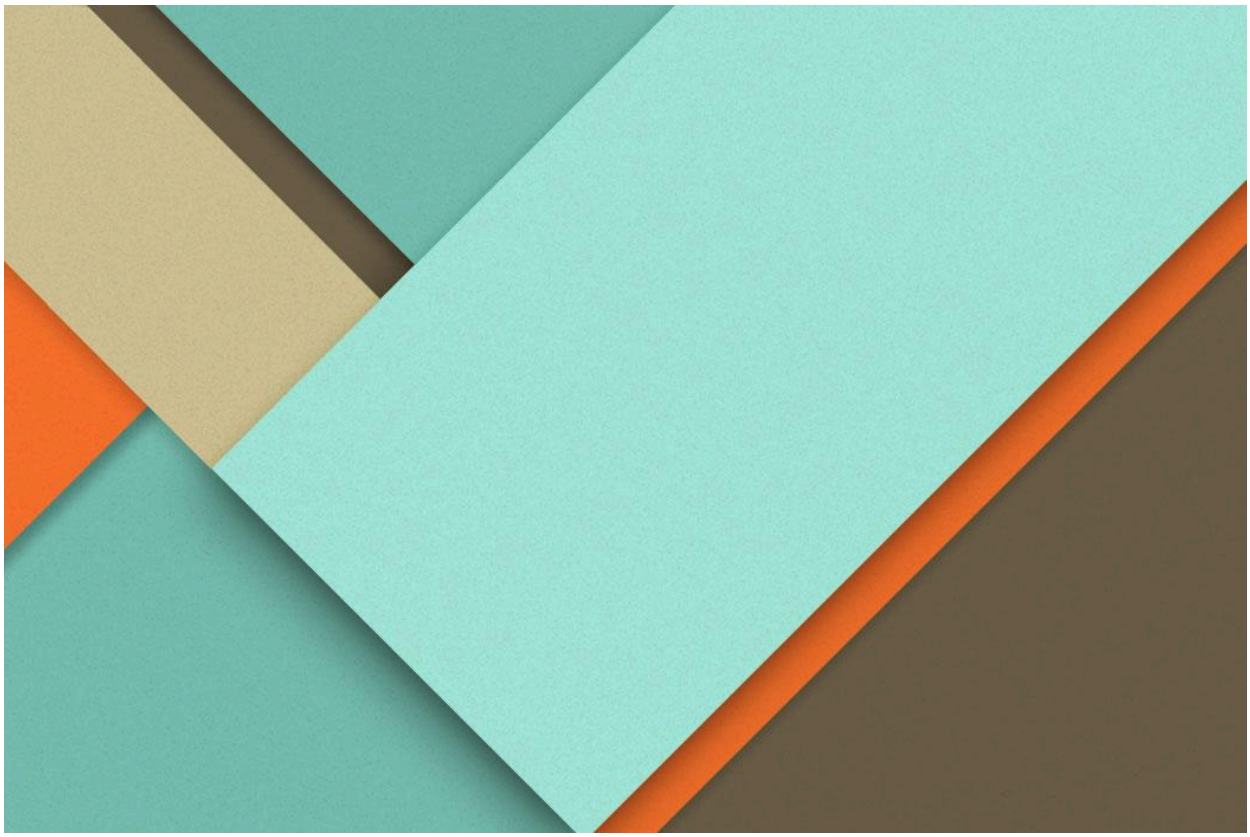


عنوان پروژه:

پیش‌بینی قیمت فروش خودروهای دست دوم با استفاده از مدل رگرسیون خطی

مقدمه اول

هدف پروژه

در این پروژه، ما قصد داریم یک مدل پیش‌بینی قیمت فروش خودروها را توسعه دهیم. این پروژه از طرف یک شرکت خودروسازی یا یک پلتفرم خرید و فروش خودرو انجام شده است. هدف اصلی این پروژه ارائه یک ابزار دقیق و قابل اعتماد برای تخمین قیمت خودروهای دست دوم است تا به خریداران و فروشندگان کمک کند قیمت مناسبی برای معاملات خود تعیین کنند.

موضوع پروژه

موضوع این پروژه توسعه و ارزیابی یک مدل رگرسیون خطی برای پیش‌بینی قیمت فروش خودروها است. با استفاده از داده‌های مربوط به مشخصات فنی و ویژگی‌های خودروها، مدل ما قادر خواهد بود قیمت هر خودرو را با دقت قابل قبولی پیش‌بینی کند.

اهمیت پروژه

پیش‌بینی دقیق قیمت خودروهای دست دوم اهمیت بسیاری دارد. این پروژه می‌تواند به خریداران و فروشندگان کمک کند تا تصمیمات بهتری بگیرند و از تقلب و قیمت‌گذاری ناعادلانه جلوگیری کنند. همچنین، این مدل می‌تواند به شرکت‌های خودروسازی و پلتفرم‌های خرید و فروش خودرو در بهینه‌سازی فرآیندهای قیمت‌گذاری و بازاریابی کمک کند.

بستر پروژه

بستر این پروژه شامل استفاده از کتابخانه‌های مختلف پایتون برای پیش‌پردازش داده‌ها، آموزش مدل و ارزیابی عملکرد آن است. دیتاست اصلی این پروژه از فایل `cardekho.csv` بارگذاری شده است و شامل اطلاعات متنوعی از خودروها است.

5. مطالبی که قرار است ارائه شوند

در این پروژه، مطالب زیر ارائه خواهند شد:

- توضیح مسئله و هدف پروژه
- بررسی و پیش‌پردازش داده‌های موجود
- توسعه مدل رگرسیون خطی برای پیش‌بینی قیمت فروش خودروها
- ارزیابی عملکرد مدل با استفاده از معیارهای مختلف
- نتیجه‌گیری و پیشنهادات برای بهبود مدل

این پروژه یک مثال عملی از کاربردهای یادگیری ماشین در صنعت خودروسازی و بازارهای خرید و فروش خودرو است و نشان می‌دهد که چگونه می‌توان با استفاده از داده‌های موجود و مدل‌های آماری، پیش‌بینی‌های دقیقی انجام داد که به تصمیم‌گیری‌های بهتر کمک می‌کند.

مقدمه دوم: تحقیق‌های دیگر در این زمینه

تحقیقات پیشین

تحقیق‌های متعددی در زمینه پیش‌بینی قیمت خودروهای دست دوم انجام شده است. این تحقیق‌ها می‌توانند به دو دسته کلی تقسیم شوند:

1. **مدل‌های سنتی آماری:**
   - **رگرسیون خطی:** بسیاری از تحقیقات اولیه از مدل‌های رگرسیون خطی برای پیش‌بینی قیمت خودروها استفاده کرده‌اند. این مدل‌ها به دلیل سادگی و قابلیت تفسیر بالا، بسیار محبوب بوده‌اند.
   - **رگرسیون چندگانه:** برخی از مطالعات با استفاده از رگرسیون چندگانه و در نظر گرفتن تأثیرات متقابل بین ویژگی‌های مختلف خودروها، دقت پیش‌بینی‌ها را افزایش داده‌اند.
2. **مدل‌های یادگیری ماشین و هوش مصنوعی:**
   - **درخت‌های تصمیم:** این مدل‌ها به دلیل قابلیت تفسیر و توانایی در شناسایی روابط پیچیده بین ویژگی‌ها، مورد توجه قرار گرفته‌اند.
   - **ماشین‌های بردار پشتیبان (SVM):** این مدل‌ها به خصوص در مواردی که داده‌ها پیچیده و چند بعدی هستند، نتایج خوبی ارائه داده‌اند.
   - **شبکه‌های عصبی مصنوعی (ANN):** با افزایش قدرت محاسباتی، استفاده از شبکه‌های عصبی برای پیش‌بینی قیمت خودروها نیز رایج شده است. این مدل‌ها با توانایی یادگیری الگوهای پیچیده، دقت پیش‌بینی‌ها را بهبود بخشیده‌اند.

دسته‌بندی تحقیق‌ها

تحقیقات در این زمینه را می‌توان به دو دسته اصلی تقسیم کرد:

1. **تحقیقات بر اساس ویژگی‌های خودرو:**
   - این دسته از تحقیقات تمرکز بر ویژگی‌های فنی خودروها مانند نوع موتور، سال تولید، کیلومتر پیموده شده و مصرف سوخت دارند.
   - برخی از تحقیقات نیز به بررسی تأثیر برند و مدل خودرو بر قیمت فروش پرداخته‌اند.
2. **تحقیقات بر اساس روش‌های پیش‌بینی:**
   - در این دسته، تحقیقات به بررسی و مقایسه روش‌های مختلف پیش‌بینی می‌پردازند.
   - برخی از مطالعات به ارزیابی عملکرد مدل‌های مختلف بر روی داده‌های یکسان و مقایسه دقت آن‌ها اختصاص یافته‌اند.

اهمیت تحقیق‌های پیشین

تحقیقات پیشین نشان داده‌اند که استفاده از مدل‌های یادگیری ماشین می‌تواند دقت پیش‌بینی قیمت خودروها را بهبود بخشد. این تحقیقات همچنین نشان داده‌اند که پیش‌پردازش دقیق داده‌ها و انتخاب ویژگی‌های مناسب می‌تواند تأثیر زیادی بر عملکرد مدل‌ها داشته باشد.

در این پروژه، ما با استفاده از مدل رگرسیون خطی و بهره‌گیری از نتایج تحقیقات پیشین، سعی داریم یک مدل دقیق و کارآمد برای پیش‌بینی قیمت فروش خودروهای دست دوم توسعه دهیم.

چکیده پروژه

مسئله: پیش‌بینی قیمت فروش خودروها با استفاده از داده‌های مشخصات فنی و برند خودرو.

کاربرد: این مدل می‌تواند برای خریداران و فروشندگان خودرو مفید باشد تا قیمت مناسب برای خرید و فروش خودروهای دست دوم را تخمین بزنند.

دیتاست: دیتاست مورد استفاده شامل اطلاعات مربوط به خودروها از جمله نام، برند، سال تولید، قیمت فروش، کیلومتر پیموده شده، نوع سوخت، نوع فروشنده، گیربکس، مالکیت، مصرف سوخت، و قدرت موتور است. این دیتاست از فایل `cardekho.csv` بارگذاری شده است.

مدل: مدل مورد استفاده، رگرسیون خطی است که با استفاده از کتابخانه `scikit-learn` و همچنین یک پیاده‌سازی سفارشی از رگرسیون خطی در فایل `LinearRegression.py` اجرا شده است.

نتایج:

1. **پیش‌پردازش داده‌ها:** داده‌ها با حذف مقادیر خالی و داده‌های تکراری تمیز شدند. نام خودروها به برندهایشان ساده‌سازی شدند و برندهای کمتر محبوب به عنوان "دیگر" برچسب‌گذاری شدند. داده‌های نامتعارف و خارج از محدوده نیز حذف شدند.
2. **نرمال‌سازی:** داده‌ها برای بهبود عملکرد مدل نرمال‌سازی شدند.
3. **آموزش و ارزیابی مدل:** مدل رگرسیون خطی با استفاده از داده‌های تمیز شده آموزش داده شد و عملکرد آن با استفاده از معیار میانگین خطای مطلق (MAE) ارزیابی شد.

نتیجه نهایی نشان‌دهنده عملکرد قابل قبول مدل در پیش‌بینی قیمت فروش خودروها بود، اما ممکن است نیاز به بهبود هایی برای دقت بیشتر وجود داشته باشد.

فایل آپلود شده شامل موارد زیر است:

1. **cardekho.csv**: دیتاست مورد استفاده
2. **LinearRegression.py**: اسکریپت مربوط به مدل رگرسیون خطی
3. **main.ipynb**: نوت‌بوک اصلی که احتمالاً شامل اجرای مدل و تحلیل نتایج است

گزارش نهایی پروژه

مقدمه

این پروژه به هدف پیش‌بینی قیمت فروش خودروها بر اساس ویژگی‌های مختلف آنها انجام شده است. در این گزارش، مراحل مختلف پیش‌پردازش داده‌ها، تحلیل داده‌ها، مدل‌سازی و ارزیابی مدل‌ها به تفصیل توضیح داده شده است.

کتابخانه‌ها و تنظیمات اولیه

در ابتدای کد، کتابخانه‌های مورد نیاز برای این پروژه وارد شده‌اند. این کتابخانه‌ها شامل `pandas` برای پردازش داده‌ها، `numpy` برای عملیات عددی، `LabelEncoder` برای کدگذاری برچسب‌ها، `Linear Regression` برای ساخت مدل رگرسیون خطی، `matplotlib.pyplot` برای ترسیم نمودارها و `warnings` برای مدیریت هشدارها هستند. همچنین، کتابخانه‌های `train_test_split` و `mean_absolute_error` از `sklearn` برای تقسیم داده‌ها و ارزیابی مدل‌ها استفاده شده‌اند.

```
import pandas as pd
import numpy as np
from sklearn.preprocessing import LabelEncoder
from LinearRegression import LinearRegression
import matplotlib.pyplot as plt
import warnings
warnings.filterwarnings('ignore', category=FutureWarning)
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_absolute_error
from sklearn.linear_model import LinearRegression as lr
```

تعریف تابع نرمال‌سازی داده‌ها

یک تابع به نام `normalize_data` تعریف شده که داده‌ها را نرمال‌سازی می‌کند. این تابع مقادیر حداقل و حداکثر را برای هر ویژگی محاسبه کرده و سپس داده‌ها را در بازه [0, 1] قرار می‌دهد.

```
def normalize_data(X):
    min_vals = np.min(X, axis=0)
    max_vals = np.max(X, axis=0)
    normalized_X = (X - min_vals) / (max_vals - min_vals)
    return normalized_X
```

پیش‌پردازش داده‌ها

در این بخش، داده‌ها از فایل `cardekho.csv` خوانده شده و بررسی اولیه روی آنها انجام شده است. سپس مقادیر نال حذف و داده‌های تکراری حذف شده‌اند

```
df = pd.read_csv('cardekho.csv')
df.dropna(inplace=True)
df.drop_duplicates(inplace=True)
```

پردازش نام برند خودرو

نام برند خودرو از ستون `name` استخراج شده و فقط ده برند برتر نگه داشته شده‌اند. سایر برندها با مقدار `other` جایگزین شده‌اند.

حذف داده‌های پرت

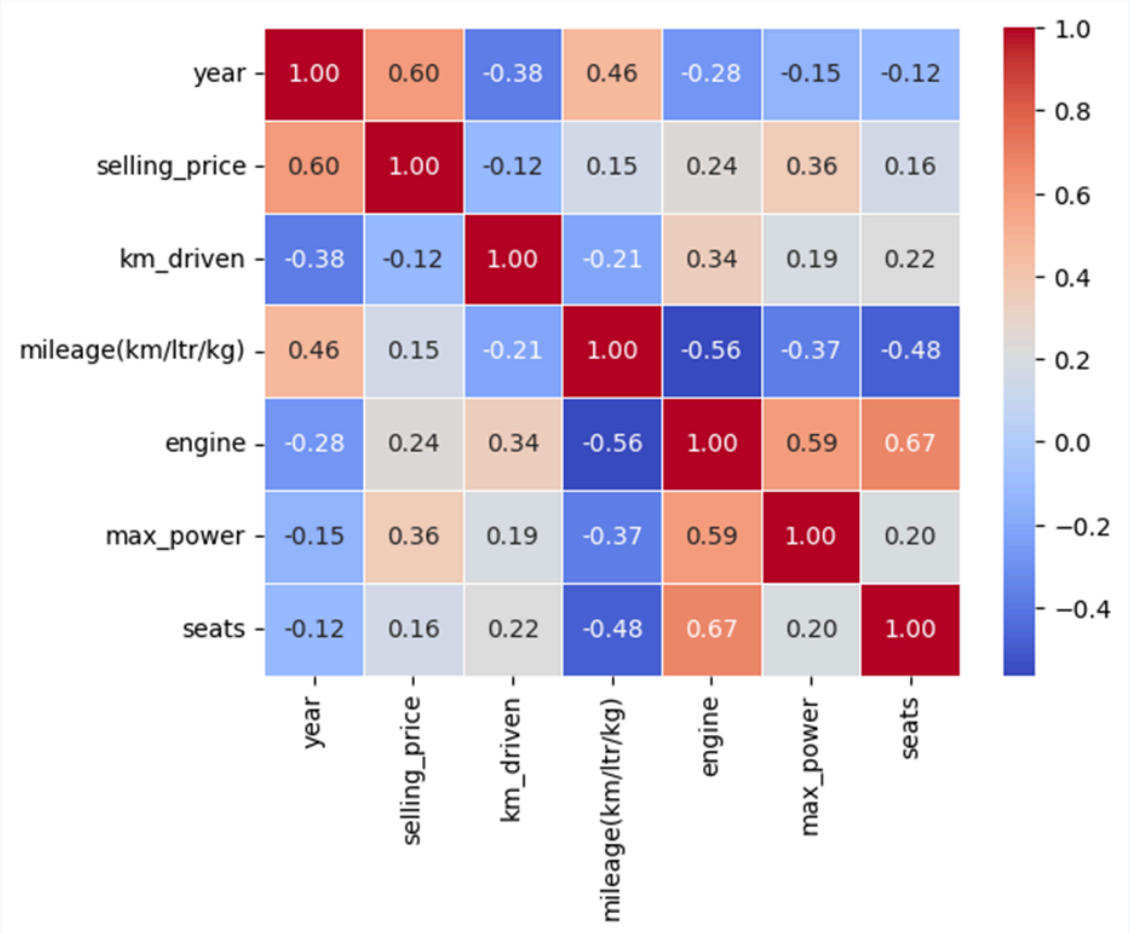
برای حذف داده‌های پرت از ستون `selling_price`، داده‌هایی که خارج از بازه مشخصی هستند حذف شده‌اند.

تبدیل نوع داده‌ها

نوع داده‌های ستون `max_power` به عددی تغییر کرده است.

محاسبه همبستگی‌ها

برای بررسی همبستگی بین ویژگی‌ها، یک نمودار heatmap رسم شده است.



کدگذاری ویژگی‌های متنی

برای تبدیل ویژگی‌های متنی به عددی از روش `one-hot encoding` استفاده شده است.

تقسیم داده‌ها به مجموعه‌های آموزشی و آزمایشی

داده‌ها به دو مجموعه آموزشی و آزمایشی تقسیم شده‌اند و داده‌های ویژگی‌ها نیز نرمال‌سازی شده‌اند.

آموزش مدل‌ها

دو مدل رگرسیون خطی تعریف و آموزش داده شده‌اند. یکی از این مدل‌ها توسط خودمان تعریف شده و دیگری از کتابخانه `sklearn` استفاده شده است.

پیش‌بینی و ارزیابی مدل‌ها

محاسبه شده است. نتایج نشان می‌دهند **که مدل ما** عملکرد مناسبی (MAE) پیش‌بینی‌ها انجام شده و خطای مطلق متوسط دارد

ترسیم نمودارها

.دو نمودار برای مقایسه تفاوت پیش‌بینی‌ها با مقادیر واقعی و همچنین مقایسه مقادیر پیش‌بینی شده و واقعی ترسیم شده‌اند

نتیجه‌گیری

نتایج نشان می‌دهند که مدل تعریف شده توسط ما عملکرد قابل قبولی دارد و اختلاف زیادی با مدل `sklearn` ندارد. این امر نشان‌دهنده صحت و دقت مدل ما است.

کلاس LinearRegression

```
class LinearRegression:
    def __init__(self, learning_rate=0.4, num_iterations=2000):
        self.learning_rate = learning_rate
        self.num_iterations = num_iterations
        self.weights = None
        self.bias = None

    def fit(self, X, y):
        num_samples, num_features = X.shape
        self.weights = np.zeros(num_features)
        self.bias = 0
        for _ in range(self.num_iterations):
            y_predicted = np.dot(X, self.weights) + self.bias
            dw = (1 / num_samples) * np.dot(X.T, (y_predicted - y))
            db = (1 / num_samples) * np.sum(y_predicted - y)
            self.weights -= self.learning_rate * dw
            self.bias -= self.learning_rate * db

    def predict(self, X):
        return np.dot(X, self.weights) + self.bias
```

تحلیل نتایج

در نهایت پس از تمرین دادن مدل برروی بخش تمرین داده‌ها، و سپس گرفتن بردار پیش‌بینی، به نتایج زیر میرسیم:

Mean Absolute Error of the Model : 58,000

شرح و مقایسه نتایج

در این پروژه همچنین از مدل‌های Linear Regression, SVM, lightgbm Regressorاز کتابخانه sklearn استفاده شده است و مقایسه نتیجه‌های به‌دست آمده به صورت زیر است:

می‌توانیم مشاهده کنیم که مدل ما نتیجه بهتری از مدل SVR ارائه می‌دهد، همچنین نتیجه نزدیکی به مدل Linear Regression از کتابخانه sklearn دارد که نشان می‌دهد به درستی مدل خود را تعریف کرده‌ایم.

اما مدل LGBM نتیجه بهتری از مدل ما داشته است.

برای بهتر کردن نتیجه:

وجود نقاط پرت: برخی نقاط پرت وجود دارند که ممکن است نیاز به بررسی بیشتر داشته باشند تا بتوان دلایل تفاوت‌های بزرگ را شناسایی کرد.

بهبود مدل:

1 . افزایش داده‌های آموزشی:

با افزودن داده‌های بیشتر به مجموعه آموزشی می‌توان مدل‌ها را بهتر آموزش داد و دقت پیش‌بینی‌ها را افزایش داد.

2. استفاده از ویژگی‌های بیشتر:

اضافه کردن ویژگی‌های جدید و مرتبط با قیمت خودروها می‌تواند به بهبود مدل‌ها کمک کند.

3. بهینه‌سازی پارامترهای مدل‌ها:

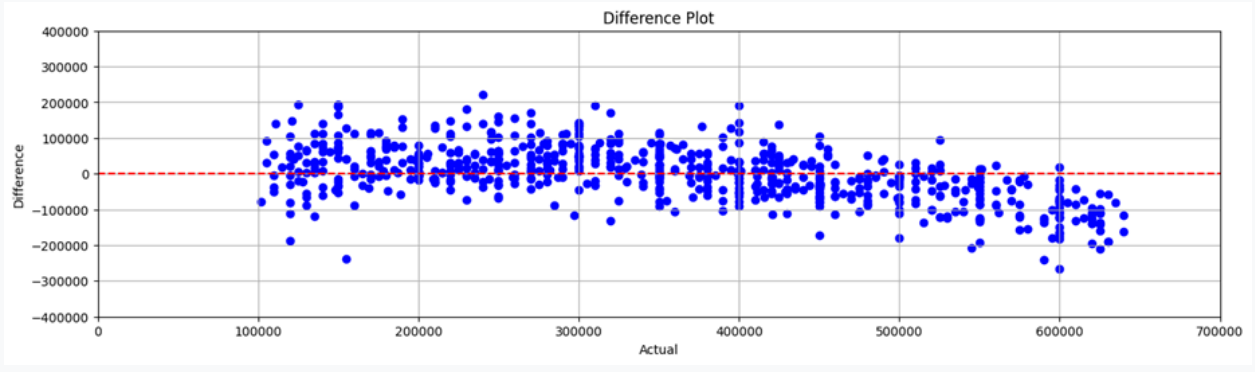
با استفاده از تکنیک‌های بهینه‌سازی هایپرپارامترها مانند جستجوی شبکه‌ای (Grid Search) یا جستجوی تصادفی (Random Search)، می‌توان پارامترهای بهینه مدل‌ها را یافت و دقت آنها را افزایش داد.

4. استفاده از مدل‌های پیچیده‌تر:

مدل‌های پیچیده‌تر مانند شبکه‌های عصبی عمیق (Deep Neural Networks) می‌توانند بهبودهای قابل توجهی در دقت پیش‌بینی‌ها به همراه داشته باشند.

به طور کلی، مدل رگرسیون خطی ساخته شده عملکرد قابل قبولی دارد و می‌توان با بهبودهای بیشتر دقت آن را افزایش داد.

نمودار تفاوت قیمت پیش‌بینی و قیمت واقعی در مقابل قیمت واقعی(Actual vs. Difference)



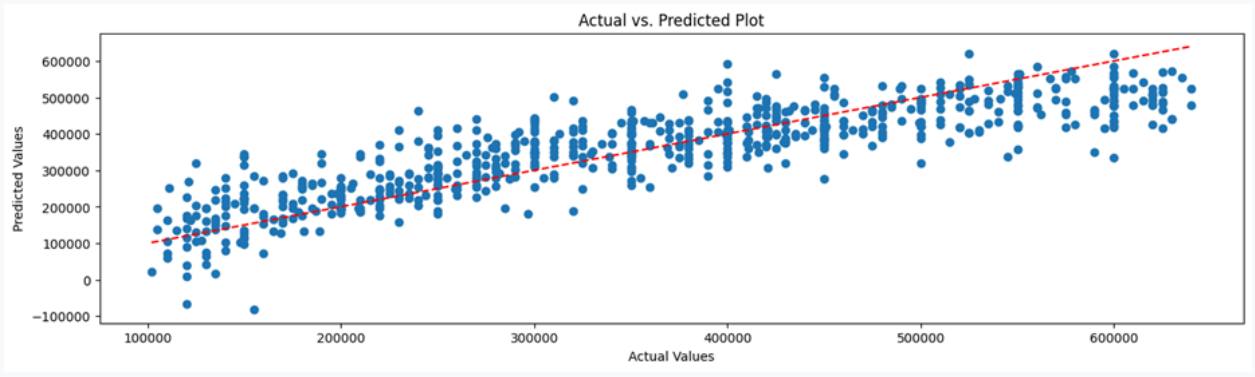

```
difference = pred - y_test

plt.figure(figsize=(16, 9))
plt.subplot(2, 1, 1)
plt.scatter(y_test, difference, color='blue')
plt.xlim(0, 7e5)
plt.ylim(-4e5, 4e5)
plt.axhline(y=0, color='red', linestyle='--')
plt.xlabel('Actual')
plt.ylabel('Difference')
plt.title('Difference Plot')
plt.grid(True)
plt.show()
```

تحلیل نمودار تفاوت

- **توزیع تفاوت‌ها:** نمودار نشان می‌دهد که تفاوت‌ها (error) به طور کلی به صورت تصادفی توزیع شده‌اند. این نشان‌دهنده این است که مدل ما به طور کلی دچار بایاس نیست.
- **نقاط پرت:** برخی نقاط پرت وجود دارند که تفاوت‌های زیادی با مقدار واقعی دارند. این ممکن است به دلیل وجود داده‌های پرت در مجموعه داده‌ها باشد یا ممکن است نشان‌دهنده نقاطی باشد که مدل نتوانسته به خوبی پیش‌بینی کند.

- **محدوده تفاوت‌ها:** تفاوت‌ها عمدتاً بین -4e5 تا 4e5 قرار دارند. این نشان‌دهنده این است که مدل در برخی موارد تفاوت زیادی با مقادیر واقعی دارد، اما به طور کلی تفاوت‌ها در محدوده قابل قبولی قرار دارند.
- **خط قرمز:** خط قرمز در محور افقی (y=0) نشان‌دهنده جایی است که تفاوت بین مقادیر پیش‌بینی شده و واقعی صفر است. هرچه نقاط نزدیک‌تر به این خط باشند، پیش‌بینی مدل دقیق‌تر است.

نمودار مقادیر واقعی در مقابل پیش‌بینی شده (Actual vs. Predicted Plot)



این نمودار تفاوت بین مقادیر واقعی و پیش‌بینی شده را نشان می‌دهد. محور افقی مقادیر واقعی (`y_test`) و محور عمودی تفاوت بین مقادیر پیش‌بینی شده و واقعی (`difference`) است.

این نمودار مقادیر واقعی (`y_test`) را در مقابل مقادیر پیش‌بینی شده (`pred`) نشان می‌دهد. محور افقی مقادیر واقعی و مح ور عمودی مقادیر پیش‌بینی شده است.

```
plt.figure(figsize=(16, 9))
plt.subplot(2, 1, 2)
plt.scatter(y_test, pred)
plt.plot([min(y_test), max(y_test)], [min(y_test), max(y_test)], color='red', lines
plt.xlabel('Actual Values')
plt.ylabel('Predicted Values')
plt.title('Actual vs. Predicted Plot')
plt.show()
```

تحلیل نمودار مقادیر واقعی در مقابل پیش‌بینی شده

- **توزیع نقاط:** نقاط به طور کلی در اطراف خط قرمز (y=x) توزیع شده‌اند که نشان‌دهنده این است که مدل به طور کلی مقادیر واقعی را به درستی پیش‌بینی کرده است.
- **انحراف‌ها:** هرچه نقاط نزدیک‌تر به خط قرمز باشند، پیش‌بینی مدل دقیق‌تر است. برخی نقاط از خط قرمز دور هستند که نشان‌دهنده تفاوت‌های بزرگ بین مقادیر پیش‌بینی شده و واقعی است.
- **خط قرمز:** خط قرمز (y=x) نشان‌دهنده جایی است که مقادیر پیش‌بینی شده و واقعی برابر هستند. هرچه نقاط نزدیک‌تر به این خط باشند، دقت مدل بیشتر است.
- **محدوده پیش‌بینی‌ها:** مقادیر پیش‌بینی شده در محدوده مقادیر واقعی قرار دارند که نشان‌دهنده این است که مدل به طور کلی روند کلی داده‌ها را درست تشخیص داده است.

نتیجه‌گیری کلی از نمودارها

- **مدل به طور کلی عملکرد مناسبی دارد:** توزیع نقاط در هر دو نمودار نشان می‌دهد که مدل به طور کلی قادر به پیش‌بینی درست مقادیر است.
- **وجود نقاط پرت:** برخی نقاط پرت وجود دارند که ممکن است نیاز به بررسی بیشتر داشته باشند تا بتوان دلایل تفاوت‌های بزرگ را شناسایی کرد.