Анализа на ефикасноста на автомобили со модели на регресија

Симона Ристовска

25 јуни, 2025

Абстракт

Во овој проект се анализираат податоци за автомобили со цел да се испита како различни карактеристики, како моќноста на моторот, тежината и забрзувањето, влијаат врз потрошувачката на гориво. Со помош на регресиски модели се обидуваме да изградиме предвидувања за ефикасноста на горивото и да идентификуваме кои фактори имаат најголемо влијание.

Овој проект ја следи структурата на Lab 3 од книгата *"An Introduction to Statistical Learning"*, прилагоден на специфичните податоци и цели на анализата.

1 Преземање и подготовка на податоците

```
df = ISLP.load_data("Auto")
df = df.dropna().reset_index(drop=True)
df.head()
```

2 Едноставна линеарна регресија

Модел за зависност помеѓу mpg и horsepower:

$$\text{mpg} = \beta_0 + \beta_1 \, horsepower + \varepsilon$$

2.1 Код

```
y = df['mpg']
X1 = sm.add_constant(df['horsepower'])
model_lin = sm.OLS(y, X1).fit()
model_lin.summary()
```

2.2 Анализа на резултатите

Коефициентот на intercept (const = 39.9359) укажува на очекувана вредност на mpg каде horsepower е нула, односно базична ефикасност од околу 39.94 mpg. Коефициентот за horsepower (–0.1578) е негативен и статистички значаен (t = –24.489, p<0.001), што укажува дека секоја единица моќност го намалува просечното трошење гориво за околу 0.16 mpg. $R^2 = 0.606$ покажува дека околу 60.6% од варијансата на mpg се објаснува со horsepower како единствен предиктор.

Доколку го разгледаме 95% интервалот на доверба за наклонот (–0.171 до –0.145), можеме да бидеме доста сигурни во негативната врска помеѓу horsepower и mpg. Високата вредност на Ф-статистиката, 599.7 (p≈7.03e–81) ја потврдува општата статистичка значајност на моделот, односно дека вклучувањето на horsepower значајно го намалува остатокот на грешката во предвидувањето.

3 Мулти линеарна регресија

Ги вклучуваме и weight и acceleration:

$$\text{mpg} = \beta_0 + \beta_1 horsepower + \beta_2 weight + \beta_3 acceleration + \varepsilon$$

3.1 Код

```
X_lin = df[["horsepower", "weight", "acceleration", "displacement"]]
X_lin = sm.add_constant(X_lin)
model_multi = sm.OLS(y, X_lin).fit()
model_multi.summary()
```

3.2 Анализа на резултатите

При вклучување на четири предиктори, R^2 се зголемува од 0.606 кај едноставната линеарна регресија на околу 0.707, што укажува дека моделот објаснува околу 70.7% од варијансата на *mpg*. Пресекот (const ≈ 45.25) претставува очекувано ниво на *mpg* кога сите предиктори се нула.

Коефициентот за *horsepower* е негативен (≈ -0.0436, $p \approx 0.009$), што значи дека зголемување на моќноста за една единица просечно го намалува *mpg* за околу 0.044. Коефициентот за *weight* е исто така негативен и статистички многу значаен (≈ -0.0053, $p < 0.001$), што укажува дека поголемата маса исто така го намалува *mpg*.

За разлика од тоа, коефициентите за *acceleration* (≈ -0.023, $p \approx 0.854$) и *displacement* (≈ -0.0060, $p \approx 0.372$) се статистички незначајни, што покажува дека нивното влијание врз *mpg* не може со сигурност да се тврди во овој модел.

F-статистиката е 233.4 ($p < 10^{-101}$), што ја потврдува значајноста на сите предиктори во моделот. Интервалите на доверба за значајните коефициенти се тесни, што дополнително ја зајакнува довербата во нивните проценки.

4 Полиномијална линеарна регресија

Ги вклучуваме horsepower (со полиномски член), weight, acceleration и displacement:

$$mpg = \beta_0 + \beta_1 horsepower + \beta_2 horsepower^2 + \beta_3 weight + \beta_4 acceleration + \beta_5 displacement + \varepsilon$$

4.1 Код

```
poly = PolynomialFeatures(degree=2, include_bias=False)
Xp = poly.fit_transform(df[["horsepower"]])
cols = poly.get_feature_names_out(["horsepower"])
Xp = pd.DataFrame(Xp, columns=cols, index=df.index)

Xp["weight"] = df["weight"]
Xp["acceleration"] = df["acceleration"]
Xp["displacement"] = df["displacement"]

Xp = sm.add_constant(Xp)
model_poly = sm.OLS(df["mpg"], Xp).fit()
print(model_poly.summary())
```

4.2 Анализа на резултатите

Вклучувањето на полиномен член и дополнителните предиктори доведува до зголемување на R^2 на 0.751, што укажува дека моделот објаснува околу 75% од варијансата на *mpg*. Пресекот (const ≈ 65.32) претставува очекувано ниво на *mpg* кога сите предиктори се нула. Коефициентите за *horsepower* (≈ -0.380, $p < 0.001$) и неговиот полиномски член $horsepower^2$ (≈ 0.0011, $p < 0.001$) укажуваат на нелинеарна врска, каде што негативниот ефект на *horsepower* се помалку влијае кај поголеми вредности.

Коефициентот за *weight* е негативен (≈ -0.0015, $p \approx 0.08$), но со повисока p-вредност што ја намалува неговата статистичка значајност. Забрзувањето има негативен и значаен ефект (≈ -0.481, $p < 0.001$), а исто така и *displacement* (≈ -0.019, $p = 0.003$), што покажува дека зголемувањето на овие карактеристики е поврзано со опаѓање на ефикасноста на горивото.

Погледот на метриките за мултиколинеарност покажува Condition Number $\approx 2.7 \times 10^5$, што сигнализира на изразена мултиколинеарност, особено поради полиномскиот член и поврзаноста на *horsepower*, *displacement* и *weight*.

Заклучокот е дека и покрај високата мултиколинеарност, моделот значајно ја подобрува објаснетата варијанса на *mpg*, но потребно е внимание при интерпретација на резултатите (најверојатно).

5 ANOVA споредба на моделите

Со цел да се процени дали вклучувањето на полиномскиот член и дополнителните променливи значајно го подобрува моделот, се прави ANOVA анализа за споредба на едноставниот линеарен модел (`model_lin`), полиномскиот модел (`model_poly`) и мултипроменливиот модел (`model_multi`).

5.1 ANOVA: линеарен наспроти полиномски модел

```
anova_lm(model_lin, model_poly)
```

Резултатите покажуваат дека додавањето на полиномниот член за *horsepower* доведува до значајно намалување на грешките на моделот(residuals) (SSR се намалува од 9385.92 на 5922.45), со F статистика од приближно 56.43 и p-вредност $\approx 1.83 \times 10^{-37}$. Ова јасно укажува дека полиномниот модел значајно подобро ги опишува податоците во споредба со едноставниот линеарен модел.

5.2 ANOVA: споредба на сите три модели

```
anova_lm(model_lin, model_poly, model_multi)
```

Во споредба на сите три модели, првата фаза (од `model_lin` кон `model_poly`) покажува исти резултати како претходно ($F \approx 48.01$, $p \approx 9.26 \times 10^{-33}$). Вториот чекор (од `model_poly` кон `model_multi`) дава $F \approx 58.61$, но со негативна разлика во степените на слобода и ss_{diff}, што е индикатор дека мултилинеарната регресија всушност нема дополнителни параметри во однос на полиномниот модел како што се очекува во класична ANOVA, бидејќи тие модели се поставени со различна структура.

Сепак, значајното намалување на SSR при додавање на полиномите и дополнителни предиктори укажува дека секој следен модел ја намалува варијансата на остатоците и овозможува подобро прилагодување на податоците. *mpg*.

Резултатите покажуваат значајно намалување на остатоците ($p<0.001$), што го оправдува додавањето на квадратичниот термин.

6 Оценка на моделот на тест сет

Со цел да се процени колку добро моделот се генерализира на нови податоци, податоците се поделени на тренинг и тест сет во сооднос 70:30. Се користи истиот полиномијален модел со предиктори *horsepower* и неговиот полиномен член, како и *weight*, *acceleration* и *displacement*.

6.1 Код

```
X_full = Xp.copy()
y = df["mpg"]
```

```
3
4 X_train, X_test, y_train, y_test = train_test_split(
5     X_full, y, test_size=0.3, random_state=42
6 )
7
8 mdl = sm.OLS(y_train, sm.add_constant(X_train, has_constant='add')).
     fit()
9 y_pred = mdl.predict(sm.add_constant(X_test, has_constant='add'))
10
11 mse = mean_squared_error(y_test, y_pred)
12 rmse = np.sqrt(mse)
13 print(f"Test MSE: {mse:.3f}")
14 print(f"Test RMSE: {rmse:.3f}")
```

6.2 Резултати

На тест сетот, добиени се следните метрики:

$$\text{Test MSE} \approx 17.71, \quad \text{Test RMSE} \approx 4.21$$

Ова значи дека просечната апсолутна грешка при предвидување на *mpg* е околу 4.2 единици.

6.3 Споредба со baseline модел

Како референтна линија, се користи модел кој секогаш ја предвидува средната вредност од *mpg* во тренинг сетот.

```
1 y_mean = y_train.mean()
2 baseline_mse = ((y_test - y_mean)**2).mean()
3 baseline_rmse = np.sqrt(baseline_mse)
4 print("Baseline RMSE:", baseline_rmse)
```

Добиената baseline грешка е:

$$\text{Baseline RMSE} \approx 7.30$$

Со тоа, моделот на линеарна регресија го намалува RMSE од околу 7.30 на 4.21, што претставува значително подобрување и покажува дека моделот добро ги користи променливите за предвидување на *mpg*.

7 Предвидувања и leverage анализа

7.1 Предвидувања за нови вредности

За да се провери како моделот предвидува mpg за нови вредности на *horsepower*, се изведени предвидувања за $hp = 50, 150, 250$.

```
new_hp = pd.DataFrame({"horsepower": [50, 150, 250]})
poly = PolynomialFeatures(degree=2, include_bias=False)
hp2 = poly.fit_transform(new_hp[["horsepower"]])
cols = poly.get_feature_names_out(["horsepower"])
new_X_hp2 = pd.DataFrame(hp2, columns=cols)
new_X_hp2 = sm.add_constant(new_X_hp2)

pred_hp2 = model_poly_2.get_prediction(new_X_hp2)
print("Mean estimates:", pred_hp2.predicted_mean)
print("95% CIs:", pred_hp2.conf_int())
print("95% PIs:", pred_hp2.conf_int(obs=True))
```

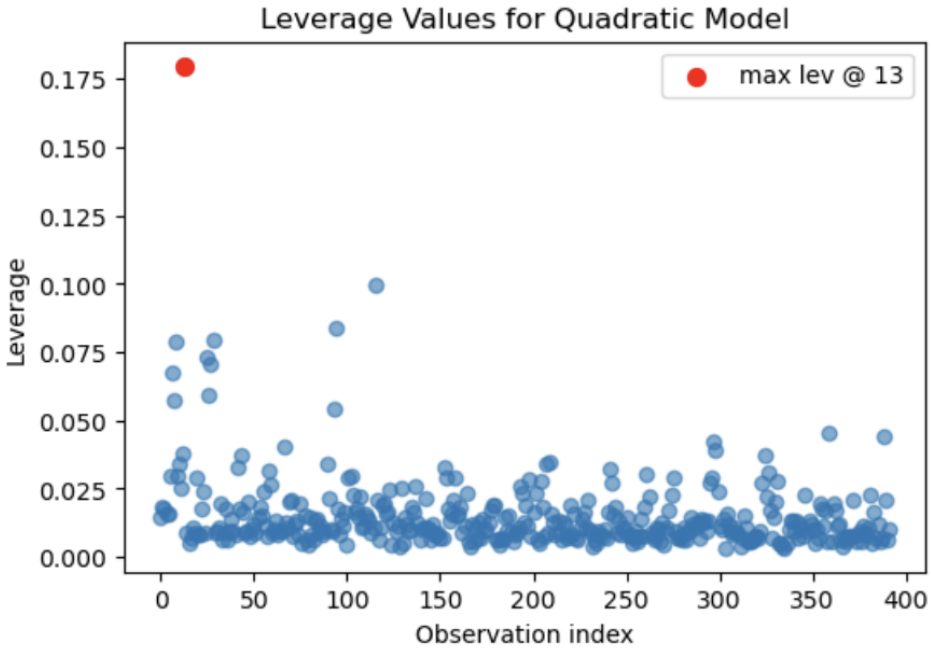
Добиени се просечни предвидувања на mpg:

$$\approx [36.67, 14.66, 17.26]$$

за $horsepower = [50, 150, 250]$, со интервали на доверба кои покажуваат пошироки граници за екстремни вредности.

7.2 Анализа на leverage вредности

Извршена е анализа на leverage вредностите за да се утврди кои набљудувања најмногу влијаат врз приспособувањето на моделот.

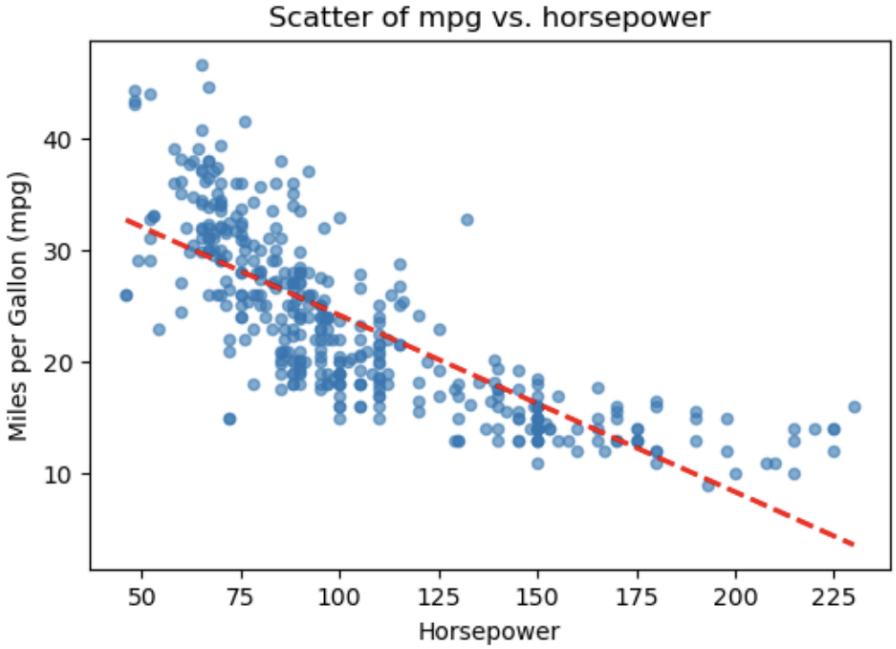


Слика 1: Leverage вредности за квадратичниот модел. Најголема leverage има набљудувањето со индекс ≈ 13.

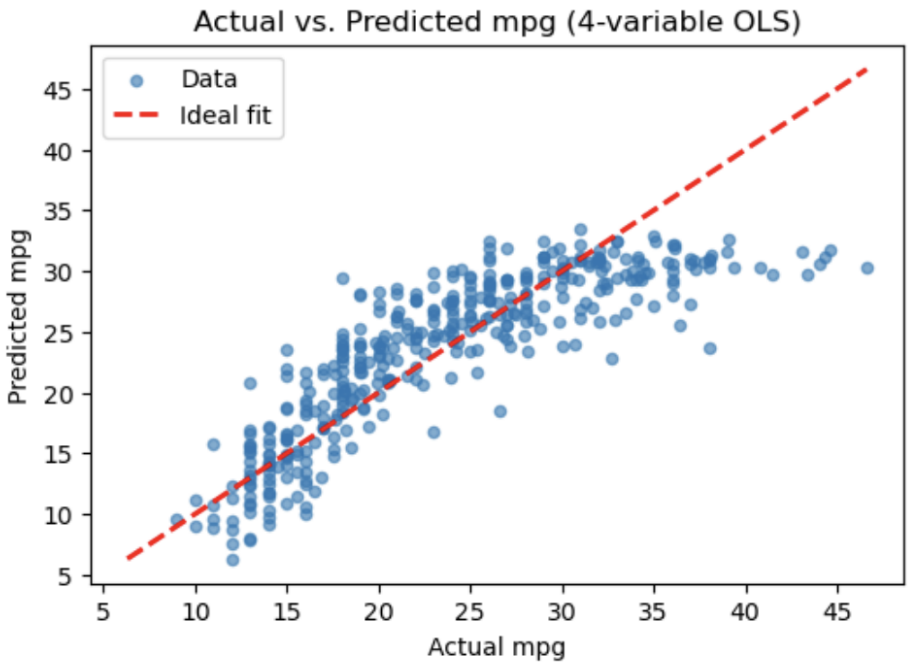
Од графикот се гледа дека најголема leverage има набљудувањето околу индекс 13, што значи дека оваа точка најсилно влијае на фитираната линија и може потенцијално да биде влијателна при проценатa на коефициентите.

8 Визуелизации на моделите

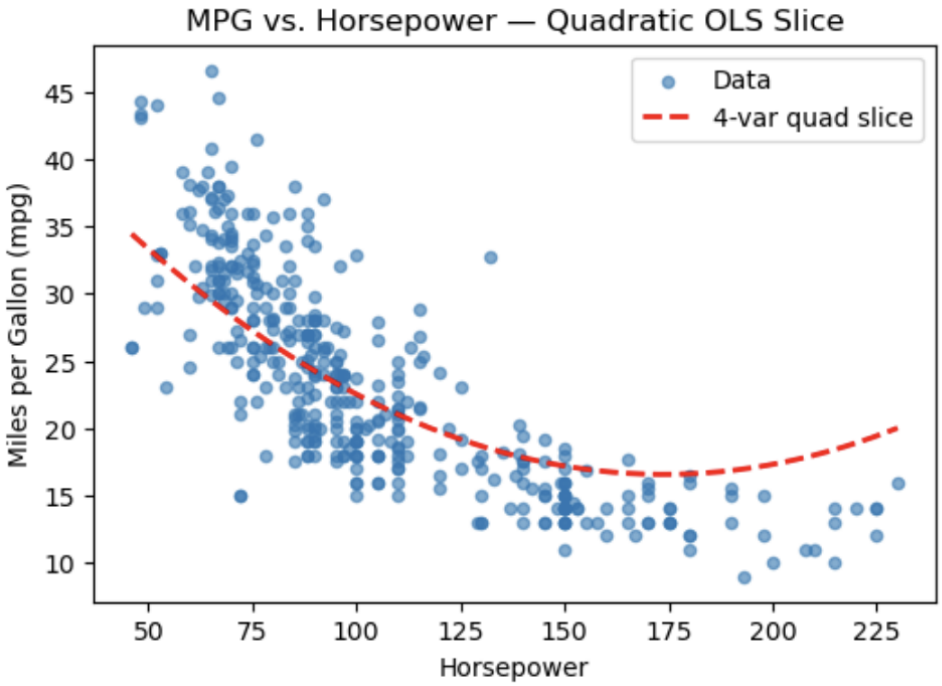
Во ова поглавје се прикажуваат графички визуализации на фитање на различните регресиски модели. За секоја од нив се прикажува scatter plot на податоците и линија или крива на предвидување добиена од соодветниот модел. Ова овозможува подобар визуелен увид во тоа колку моделите добро ги опишуваат податоците.



Слика 2: Едноставна линеарна регресија: mpg vs horsepower



Слика 3: Мулти-променлива регресија: предвидени mpg врз основа на horsepower, weight и acceleration



Слика 4: Полиномијална регресија од втор степен: mpg vs horsepower (крива)

9 Проверка на мултиколинеарност со VIF

Со цел да се процени степенот на мултиколинеарност помеѓу предикторите во моделот, се пресметуваат Variance Inflation Factors (VIF) за секоја независна променлива.

9.1 Код

```
from statsmodels.stats.outliers_influence import
    variance_inflation_factor

vifs = [variance_inflation_factor(Xp.values, i)
        for i in range(1, Xp.shape[1])]

vif_df = pd.DataFrame({
    'feature': Xp.columns[1:],
    'VIF': vifs
})

vif_df
```

9.2 Анализа на резултатите

Вредностите на VIF ги покажуваат колку варијансата на проценетиот коефициент на даден предиктор е зголемена поради мултиколинеарност со останатите предиктори. Вообичаено, VIF поголем од 5 или 10 се смета за знак на потенцијален проблем со мултиколинеарност.

Вредностите на VIF кои ги добивме се:

- *horsepower* има многу висока вредност на VIF од ≈ 70.7, што укажува на силна мултиколинеарност.
- $horsepower^2$ има исто така висока вредност од ≈ 42.5, што е очекувано поради неговата конструкција како полиномски член.
- *weight* има VIF од ≈ 14.0, што исто така укажува на значителна мултиколинеарност.
- За *acceleration* (≈ 3.19) и *displacement* (≈ 11.4), вредностите се умерени, со *displacement* над прагот од 10, што сугерира дека постои зависност со останатите променливи, но не на ниво како кај *horsepower*.

Ова потврдува дека при интерпретација на индивидуалните коефициенти треба да се биде претпазлив, особено поради силната мултиколинеарност помеѓу *horsepower*, неговиот полином и *weight*.

10 Заклучок

11 Заклучок

Овој проект дава општ преглед на начините на кои карактеристиките на автомобилите влијаат врз нивната ефикасност на гориво. Преку примена на различни регресиски модели и споредбени анализи, добиени се вредни увиди за зависностите меѓу променливите.

Резултатите покажуваат дека комплексните модели генерално обезбедуваат подобро прилагодување кон податоците, но истовремено носат и предизвици како мултиколинеарност и влијателни точки кои треба внимателно да се разгледаат.