



دانشگاه صنعتی اصفهان

دانشکده مهندسی برق و کامپیوتر

## دستورکار آزمایشگاه هوش محاسباتی

جلسه ۵  
رگرسیون لاجستیک

استاد درس: دکتر مهران صفایانی

# فصل ۵

## رگرسیون لاجستیک

### اهداف این جلسه

شما در این جلسه یاد خواهید گرفت که :

- چگونه با استفاده از رگرسیون خطی، طبقه‌بندی کنید.
  - با استفاده از `logistic regression` ، `gradian descent` را پیاده‌سازی و اشکال‌زدایی کنید
  - `logistic regression` را با `linear regression` مقایسه کنید.
  - روش نیوتون را برای `logistic regression` پیاده‌سازی کنید.
- در این جلسه از پایگاه داده `height-weight-genders.csv` استفاده خواهیم کرد همچنین کدهای نمونه و کمکی برای شما آماده شده است که راهنماهای خوبی برای شما خواهند بود.

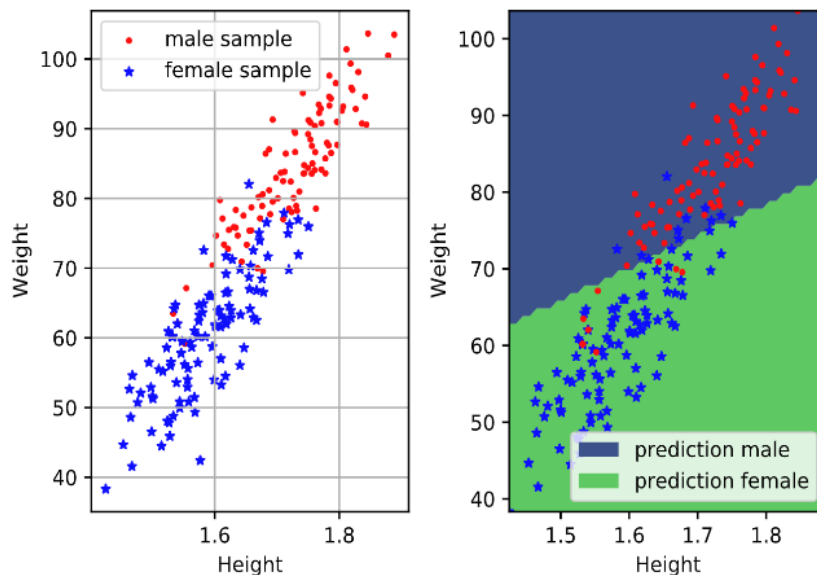
## ۱.۵ طبقه‌بندی با استفاده از رگرسیون خطی

ما سعی خواهیم کرد که با استفاده از رگرسیون خطی، عمل طبقه‌بندی را انجام دهیم. اگرچه این کار به طور کلی ایده‌ی خوبی نیست، اما برای داده‌های ساده، کار خواهد کرد. به طور مشخص، ما از داده‌های قد-وزن که در جلسه‌ی دوم داشتیم، برای پیش‌بینی جنسیت استفاده خواهیم کرد. برای مصورسازی بهتر، ما بصورت تصادفی ۲۰۰ نمونه‌ی داده را از داده‌ها، انتخاب خواهیم کرد.

### تمرین اول

طبقه‌بندی با استفاده از رگرسیون خطی

- از تابع `least_squares(y, tx)` که در تمرین قبلی داشتیم، برای محاسبه‌ی وزنهای  $w$  بر روی داده‌های وزن-قد، استفاده کنید. لطفا پیاده‌سازی مربوط به جلسه قبل را در فایل `least_squares.py` کپی کنید.
- نقاط داده و همچنین مرز تصمیم را مانند شکل ۱، با استفاده از `visualization()` تصویرسازی کنید.



شکل ۱.۵: طبقه‌بندی با استفاده از least square

## ۲.۵ رگرسیون لاجستیک

### تمرین دوم

در این تمرین می‌خواهیم رگرسیون لاجستیک<sup>۱</sup> را با استفاده از نزول گرادیان<sup>۲</sup> پیاده سازی کنیم.

- درون فایل این جلسه، تابع `sigmoid()` را تکمیل کنید.
  - درون فایل این جلسه، توابع `calculate_loss()` و `calculate_gradient()` را تکمیل کنید. خروجی تابع اول باید منفی تابع هزینه `log-likelihood` باشد، در حالی که تابع دوم، گرادیان این تابع هزینه را بر اساس پارامترهای  $w$  برمی‌گرداند.
  - تابع `learning_by_gradient_descent()` را برای رگرسیون لاجستیک پیاده سازی کنید. شما باید تابع هزینه و گرادیان نسبت به  $w$  را محاسبه کنید.
  - پیش‌بینی‌های بدست آمده را بر روی نمودار نشان دهید، خروجی شما باید چیزی شبیه به قسمت سمت راست تصویر<sup>۱</sup> باشد.
  - آیا نتایجی که بدست آورده‌اید، منطقی هستند؟
- حال که نزول گرادیان را داریم، به راحتی می‌توانیم روش نیوتون را پیاده‌سازی کنیم

### تمرین سوم

در این تمرین قصد داریم روش نیوتون را برای رگرسیون لاجستیک پیاده‌سازی کنیم.

- درون فایل مربوط به این جلسه، در ابتدا تابع `calculate_hessian()` را تکمیل کنید سپس تابع `logistic_regression()` را با استفاده از `calculate_loss()`، `calculate_gradient()` و `calculate_hessian()` کامل کنید. تابع `logistic_regression()` باید `loss`، گرادیان و Hessian را خروجی دهد. توجه داشته باشید که این تابع نباید پارامترهای  $w$  را آپدیت کند یا بعنوان خروجی، چاپ کند.

### تمرین چهارم

در این تمرین قصد داریم مفهوم رگرسیون لاجستیک همراه با جریمه<sup>۳</sup> را مورد بررسی قرار دهیم.

- درون فایل این جلسه، تابع `penalized_logistic_regression()` را تکمیل کنید. برای این کار شما باید `regularization term` که برابر  $\lambda \|w\|^2$  است را اضافه کنید. برای چک کردن معقول بودن، مقدار  $\lambda$  را به یک مقدار خیلی تنظیم کنید و ببینید آیا خروجی مشابه قبل است یا خیر. هنگامی که این کار را انجام دادید، تابع `learning_by_penalized_gradient()` را تکمیل کنید و مقدار  $\lambda$  را افزایش داده و چک کنید آیا  $\|w\|$  کاهش می‌یابد یا خیر؟

<sup>۱</sup> Logistic Regression  
<sup>۲</sup> gradient descent  
<sup>۳</sup> penalty

- نتیجه را با خروجی نزول گرادیان چک کنید. برای اشکال یابی کردن، مقدار و نُرم گرادیان را در هر تکرار چاپ کنید. تمامی این مقادیر باید در هر تکرار، کاهش یابند.