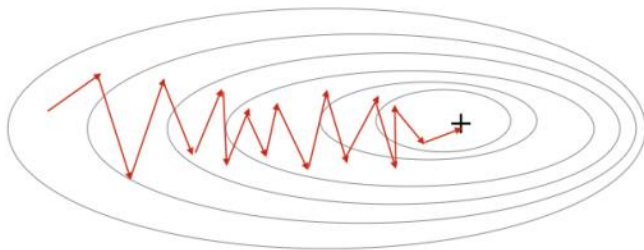




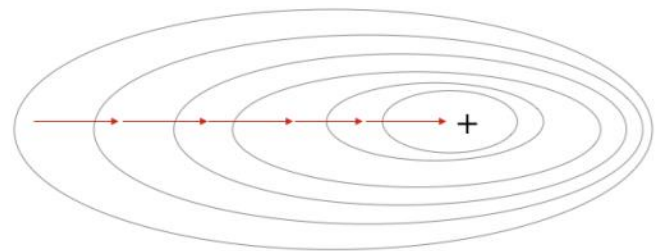
✓ برای سوال اول قسمت الف، تمامی مراحل به طور کامل اجرا شده و فرمول‌های استفاده شده در ابتدا ذکر شده است. نکته قابل توجه نحوه محاسبه مقادیر گرادیان است، در واقع مرحله جاگذاری در فرمول، با استفاده از ماشین حساب و کد پایتون اینجانب انجام شد، اما مراحل کامل و جامع گویای روش حل است. در نهایت نیز شاهد کم شدن مقدار Loss هستیم و بهتر شدن وزن‌ها.

✓ یکی از معایب بزرگ روش گرادیان عدم توجه به تغییرات هر راستا است، یعنی ممکن است گرادیان در یک جهت تغییرات شدید و نوسان داشته و اما در جهتی دیگر آهسته باشد. یکی دیگر از مشکلات آن امکان توقف الگوریتم در نقاط Local Minimum است که اگر با استفاده از Mini Batch و تغییر مناسب Learning Rate می‌توان آن را بهتر کنترل کرد تا امکان گیر کردن در این نقاط کمتر شود. اما در روش گرادیان تصادفی یک مشکل مطرح آن است که به دلیل تغییر وزن‌ها پس از بررسی تعدادی از داده‌ها (یک نمونه، یا استفاده از Mini Batch و چند نمونه)، ممکن است حرکت ما به سمت منطقه Minimum دچار Noise و تغییرات بسیار زیادی شود، اگرچه در انتها به آن می‌رسیم. شکل زیر گویای این مسئله است:

Stochastic Gradient Descent



Gradient Descent



برای حل مشکلات مربوط چند راهکار پیشنهاد شده مانند در نظر گرفتن سرعت مرحله قبل برای کنترل این مرحله، اگر سرعت زیاد، آن را بیشتر و اگر سرعت کم، آن را کم می‌کنیم. این روش Momentum نام دارد. روش دیگر آن است که نقطه آینده را نیز مد نظر بگیریم و گرادیان را در آنجا نیز حساب کنیم، تا ببینیم تا همان سرعت در آن نقطه گرادیان چند خواهد شد. این روش Nesterov است. روش‌های دیگر نیز برای حل مشکلات گفته شده وجود دارد، مثلاً Learning Rate را با توجه به مقدار قبل هر پارامتر انتخاب کنیم تا مشکل گفته شده تفاوت در جهت راستا حل شود.

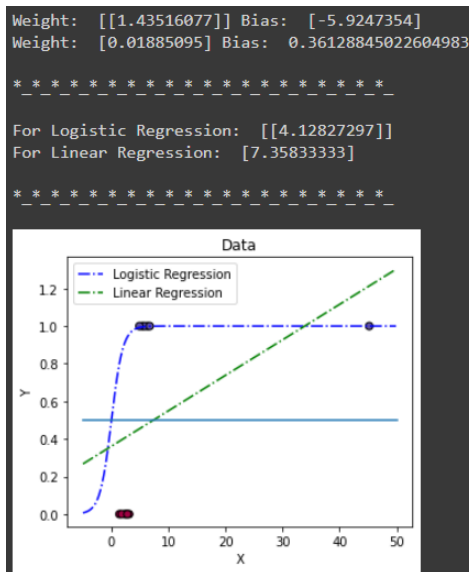


روش حل این سوال به این صورت است که ابتدا Dataset را با Linear و Logistic Regression

Regression آموزش می دهیم و سپس از وزن ها و Bias محاسبه شده در هر الگوریتم استفاده کرده و X ای که احتمالا 0.5 دارد را محاسبه می کنیم. در مدل Linear با توجه به فرمول: $Y = WX + B$ کافی است که 0.5 و بقیه اعداد حساب شده را جایگذاری کنیم. در مدل Logistic پس از محاسبه فرمول خطی بالا آن را به

Sigmoid ورودی می دهیم بنابراین $Sigmoid(Z) = 0.5$ که در این صورت پس از ساده کردن باید $Y = 0$ و $X = -B/W$ و اینگونه محاسبه می کنیم. نتایج به شکل زیر بود:

و می بینیم که مرز تصمیم در Lin جلوتر از Log است به این معنا که تعداد بیشتری را جزو کلاس 0 حساب می کند (نسبتی).





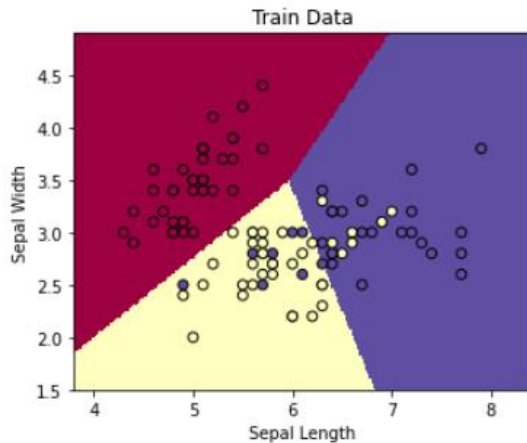
✓ داده‌های Iris, یک Dataset شامل 150 نمونه گل که سه دسته هستند, و هر گل 4 ویژگی دارد و یک اسم گروه که به آن تعلق دارد.

✓ دقت فازها متفاوت است. به این معنا که در هر بار اجرای برنامه جواب‌های مختلفی برای Accuracy می‌گیریم. در این اجرا به شکل زیر است:

```
Train Accuracy: 0.819047619047619
Test Accuracy: 0.8
```

می‌توان این نتیجه را گرفت که در مرحله

آموزش تعداد بیشتری از داده‌ها قابل تفکیک و در محدوده گروه اصلی خود قرار داشتند. دلیل این امر تصادفی انتخاب شدن داده‌های تست و آموزش است, ممکن داده‌هایی که برای با توجه به Feature خود در مرز گروه درست بوده در داده آموزش بیشتر شود. عکس زیر گویای مطلب است:

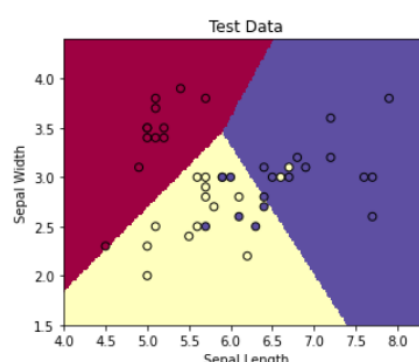
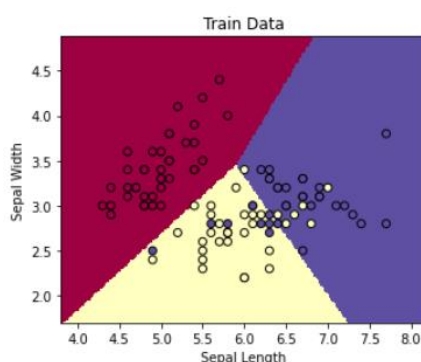


در اینجا مشاهده می‌کنیم که در نقاط بهتر برای تشخیص گروه مرتبط هستند, در حالی که در Test Data پراکندگی داده‌ها بسیار بیشتر است و با توجه به Boundary مشخص شده پس از Fit شدن, امکان خطا در این Test بیشتر شده.

در این عکس Boundaries به طور تقریبی مشخص شده و در Test Data نیز اختلافات بررسی شده, با توجه به تعداد نمونه کمتر, نسبت خطا در Test به Train بیشتر است. در حالی که در Train پخش

داده بهتری داریم. این اتفاق ممکن در اجرای دوباره برای Test بیافتد چون انتخاب و تفکیک داده اولیه به Train و Test تصادفی است.

در اینجا برعکس اتفاق بالا افتاده, که تا حد زیادی واضح است.



```
Train Accuracy: 0.7904761904761904
Test Accuracy: 0.8222222222222222
```

✓ در نهایت نیز Confusion Matrix را داریم که با توجه به تعداد کلاس‌ها است، یعنی اگر دو کلاس باشد مربعی 2×2 و در نمونه ما که 3 کلاس دارد مربعی 3×3 است. این ماتریس نشان می‌دهد که نتیجه پیشبینی شده توسط مدل آموزش داده شده، تا چه اندازه داده‌ها را در گروه درست قرار داده، و در صورت اشتباه، با کدام گروه اشتباه شده. پس از بررسی این ماتریس مطلب سوال قبل نیز بیشتر قابل فهم می‌شود. در این ماتریس فهمیدن اینکه کدام دو گروه اطلاعات مشابه دارند و امکان اشتباه در آن دو زیاد است راحت است. به شکل زیر دقت کنید:

Train Accuracy: 0.8095238095238095
Test Accuracy: 0.8666666666666667

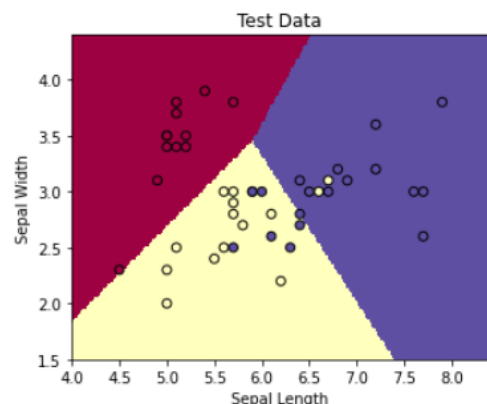
دقت برابر:

این جدول بیان می‌کند که از داده‌های گروه صفرم (0) تماماً درست پیشبینی شده ردیف 0 ستون 0.

در داده‌های گروه اول 13 مورد صحیح در دسته خود قرار گرفته، 0 مورد به اشتباه در گروه 0 و 6 مورد در گروه 2 پیشبینی شده.

در داده‌های گروه دوم 12 مورد درست، و 2 مورد غلط در گروه 1 پیشبینی شده. در مجموع نیز 45 داده Test به شکل زیر بوده است

(اشتباهات مشخص است). برای Training Data هم همین تحلیل را خواهیم داشت.



- ✓ <https://towardsdatascience.com/step-by-step-tutorial-on-linear-regression-with-stochastic-gradient-descent-1d35b088a843>
- ✓ https://scikit-learn.org/stable/auto_examples/linear_model/plot_iris_logistic.html
- ✓ https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.train_test_split.html
- ✓ https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html
- ✓ https://scikit-learn.org/stable/modules/generated/sklearn.metrics.plot_confusion_matrix.html