گزارش تمرین سوم درس داده‌کاوی

«آشنایی با نرم‌افزار Weka»

گردآورنده: سعید دادخواه

استاد: دکتر ناظرفرد

بهمن ۱۳۹۵

بخش اول: ساخت فایل arff

برای تبدیل فایل txt به فایل weka به فرمت arff از نرم‌افزار weka استفاده خواهد شد. بعد از اجرای نرم‌افزار weka در Weka GUI Chooser در شاخه Tools از گزینه Arff Viewer برای این کار استفاده می‌شود. البته برای اجرای همه مراحل به جای استفاده از GUI از یک اسکریپت برای انجام همه مراحل استفاده خواهد شد و برای انجام این مرحله از کلاس weka.core.converters.CSVLoader استفاده می‌شود.

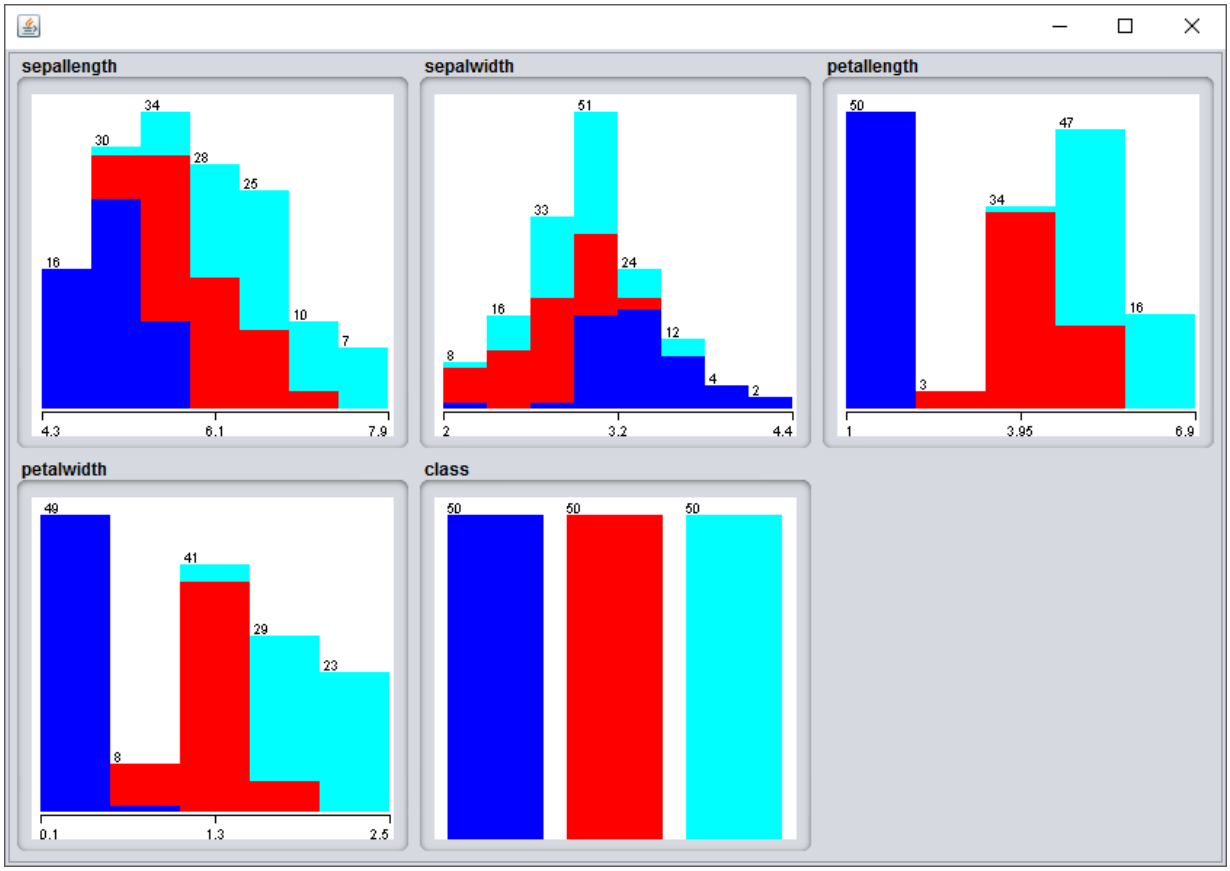
در صورتی که متد آغازین کلاس فوق اجرا شود برنامه به این شکل عمل خواهد کرد که آدرس یک فایل csv به عنوان ورودی می‌گیرد. ستون‌های این فایل csv باید دارای عنوان باشند. پس ابتدا یک برنامه به زبان پایتون اجرا می‌کنیم که یک فایل csv می‌سازد و عناوین ستون‌ها را نوشته و فایل sample.txt را به صورت csv می‌خواند و به ادامه فایل قبل اضافه می‌کند. پس از ساختن فایل csv کلاس بالا می‌تواند آن را خوانده و فایل arff را تولید کند.

برای انجام این مراحل باید پوشه‌های data و پوشه حاوی make_weka.bat و add_header.py در کنار یکدیگر قرار گیرند و فایل sample.txt نیز در پوشه data باشد. با اجرای make_weka.bat ابتدا کد پایتون add_header.py اجرا می‌شود و فایل sample.csv در پوشه data ایجاد می‌شود که همان فایل txt است با این تفاوت که عناوین ستون‌ها نیز به آن‌ها اضافه شده است. پس از آن کلاس بالا اجرا می‌شود و فایل sample.arff از روی فایل csv در همان پوشه data ساخته می‌شود.

بخش دوم: Visualize All

با انتخاب گزینه Visualize All پنجره زیر باز می‌شود. در این نمودارها به هر کلاس یک رنگ اختصاص داده می‌شود و چگونگی توزیع هر یک از کلاس‌ها در هرکدام از ویژگی‌ها نمایش داده می‌شود. برای نمایش این موارد در ویژگی‌های عددی از نمودار هیستوگرام و برای مقادیر غیرعددی از نمودار میله‌ای استفاده می‌شود.

نمودار آخر که کلاس هرکدام از نمونه‌ها را نمایش می‌دهد کاملا مشخص می‌کند که از هر کلاس پنجاه نمونه داریم. از این پس کلاس‌های مربوط به رنگ آبی، قرمز و آسمانی به ترتیب کلاس اول تا سوم نامیده خواهند شد. نمودار اول نشان می‌دهد که به‌طوری کلی کلاس اول در این ویژگی کمتر از کلاس دوم و مخصوصا کلاس سوم است. نمودار دوم نشان می‌دهد که این ویژگی نمی‌تواند نقش موثری در جداسازی کلاس سوم از دیگر کلاس‌ها داشته باشد ولی می‌تواند به تشخیص کلاس اول و دوم کمک کند. نمودار سوم و چهارم نشان می‌دهند که این ویژگی‌ها در تشخیص کلاس اول می‌توانند بسیار خوب عمل کنند. در مورد دو کلاس دیگر نیز این ویژگی‌ها می‌توانند موثر واقع شوند.

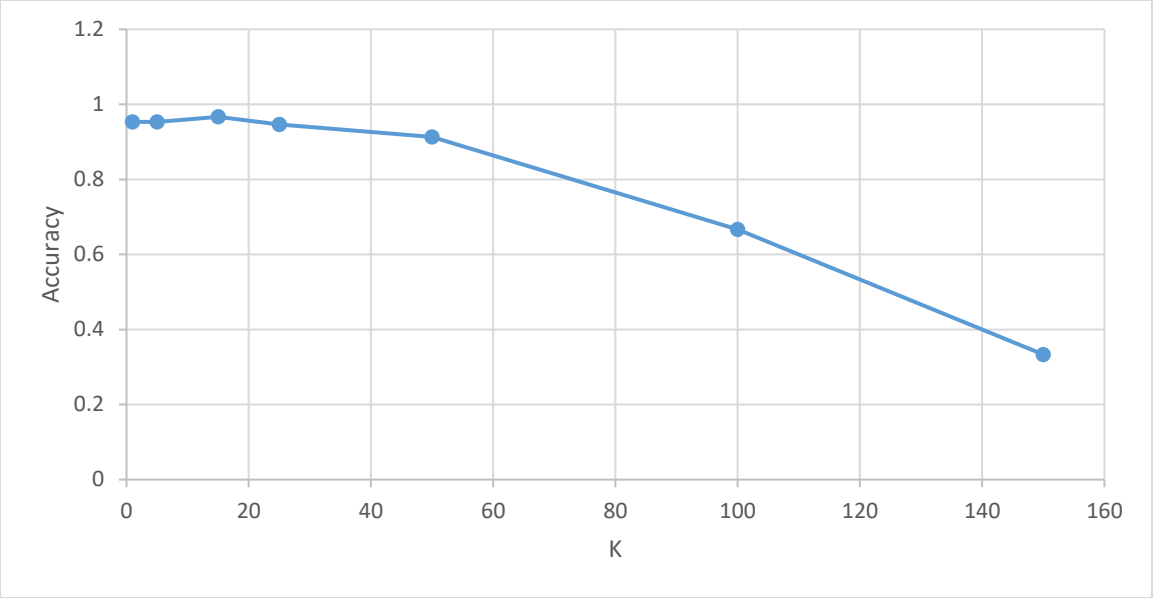


بخش سوم: نرمال‌سازی و گسسته‌سازی

با توجه به دستورات آورده شده در صورت پروژه مراحل زیر انجام شدند. برای نرمال‌سازی در تنظیمات فیلتر مقدار scale برابر دو و مقدار translation برابر منفی یک در نظر گرفته شدند. برای گسسته‌سازی نیز مقدار attribute indices برابر سه تا چهار و bins برابر پنج در نظر گرفته شدند. داده‌های به دست آمده در فایل iris-norm-disc.arff در پوشه data در دسترس هستند.

بخش چهارم: تاثیر k در الگوریتم K-NN

همانطور که در نمودار زیر مشاهده می‌شود با افزایش k دقت کاهش می‌یابد که این کاهش دقت به خاطر افزایش بایاس است. نکته دیگری که از این نمودار می‌توان دریافت کرد کاهش دقت با نزدیک شدن k به مقادیر کوچک است که به دلیل افزایش واریانس است. البته به دلیل ویژگی‌های مجموعه داده‌ها این مقدار کاهش دقت چشمگیر نیست و کاهش دقت کمی را می‌بینیم.



بخش پنجم: مقایسه درخت یک‌سطحی و چندسطحی

نتایج به شکل زیر هستند.

الگوریتم	دقت	TPR	FPR	Precision
DecisionStump	۶۶,۶۷٪	۰,۶۶۷	۰,۱۶۷	۰,۵۰۰
J48	۹۶٪	۰,۹۶۰	۰,۰۲۰	۰,۹۶۰

ماتریس درهم‌ریختگی برای الگوریتم‌ها به شکل زیر است.

کلاس	DecisionStump			J48		
	a	b	c	a	b	c
a	۵۰	۰	۰	۴۹	۱	۰
b	۰	۵۰	۰	۰	۴۷	۳
c	۰	۵۰	۰	۰	۲	۴۸

اگر معیارها را به معیارهای خوبی و بدی تقسیم کنیم از معیارهای گفته شده دقت، TPR و Precision معیارهای خوبی هستند یعنی با افزایش این مقادیر یعنی الگوریتم بهتر عمل می‌کند و برعکس FPR معیار بدی است و با افزایش مقدار آن یعنی الگوریتم بدتر عمل کرده است. DecisionStump درختی با یک سطح است. انتظار می‌رود برای مجموعه داده‌های پیچیده درخت‌های چندسطحی بهتر از درخت‌های یک‌سطحی عمل کنند چون می‌توان

گفت اگر الگوریتم ساخت درخت برای درخت‌های یک‌سطحی و چندسطحی یکسان باشد در بدترین حالت درخت‌های چندسطحی مانند درخت‌های یک‌سطحی عمل خواهند کرد. پس می‌توان انتظار داشت معیارهای خوبی برای J48 که درخت چندسطحی است و معیارهای بدی برای DecisionStump بیشتر باشد. همانطور که مشاهده می‌شود دقت، TPR و Precision الگوریتم J48 بسیار بیشتر از DecisionStump است و مقدار FPR نیز کمتر است. ماتریس درهم‌ریختگی نیز نشان‌گر همین نکته‌ها است.

نکته دیگری که می‌توان از ماتریس درهم‌ریختگی استدلال کرد این است که کلاس a تفاوت بیشتری با کلاس‌های b و c دارد و راحت‌تر می‌توان نمونه‌های مربوط به آن را تشخیص داد. در بین دسته‌بندی‌های اشتباه بیشتر اشتباه‌ها در جداسازی کلاس b و c رخ می‌دهد به‌طوری که در J48 تمامی تشخیص‌های غلط کلاس b به کلاس c و تمامی تشخیص‌های غلط کلاس c به کلاس b نسبت داده شده‌اند. در DecisionStump این قضیه به‌طور جدی‌تری خود را نمایش می‌دهد به این صورت که الگوریتم به خوبی کلاس a را تشخیص می‌دهد ولی قدرت جداسازی کلاس‌های b و c را ندارد و در صورتی که یک نمونه را متعلق به کلاس a نداند آن را به عنوان نمونه‌ای از کلاس b تشخیص می‌دهد.

بخش ششم

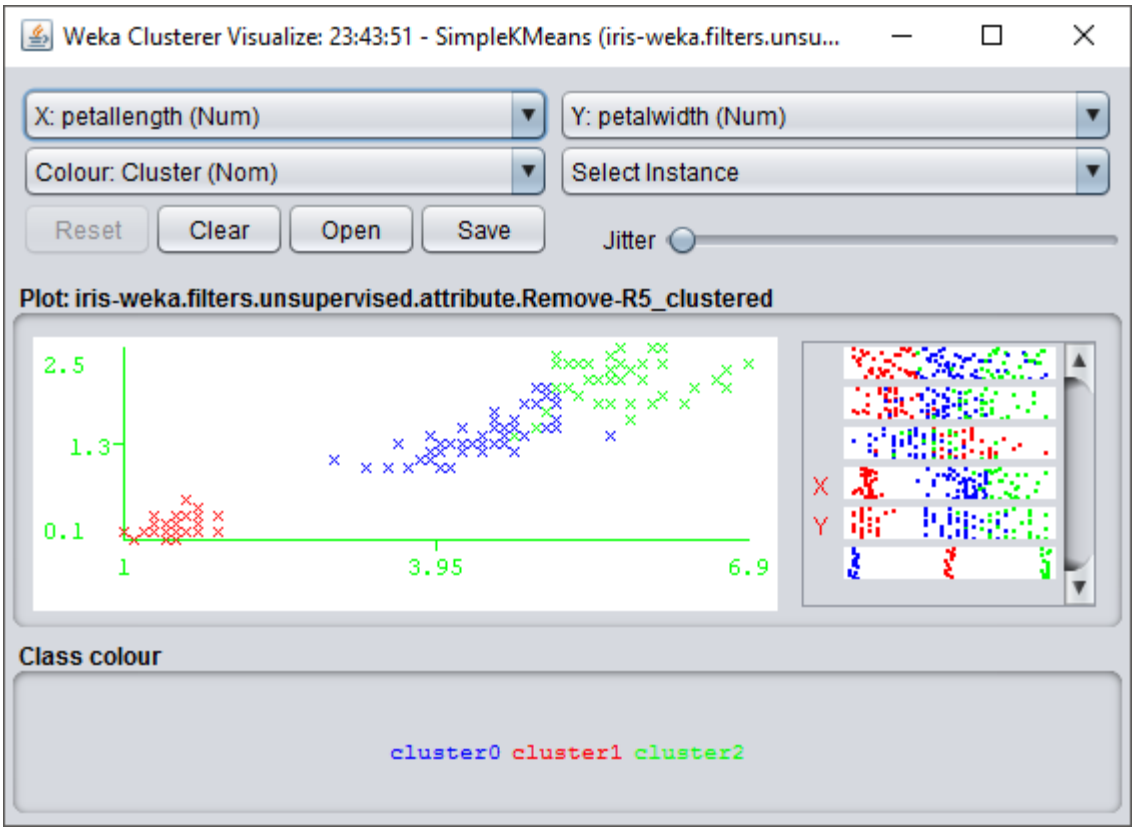
رگرسیون

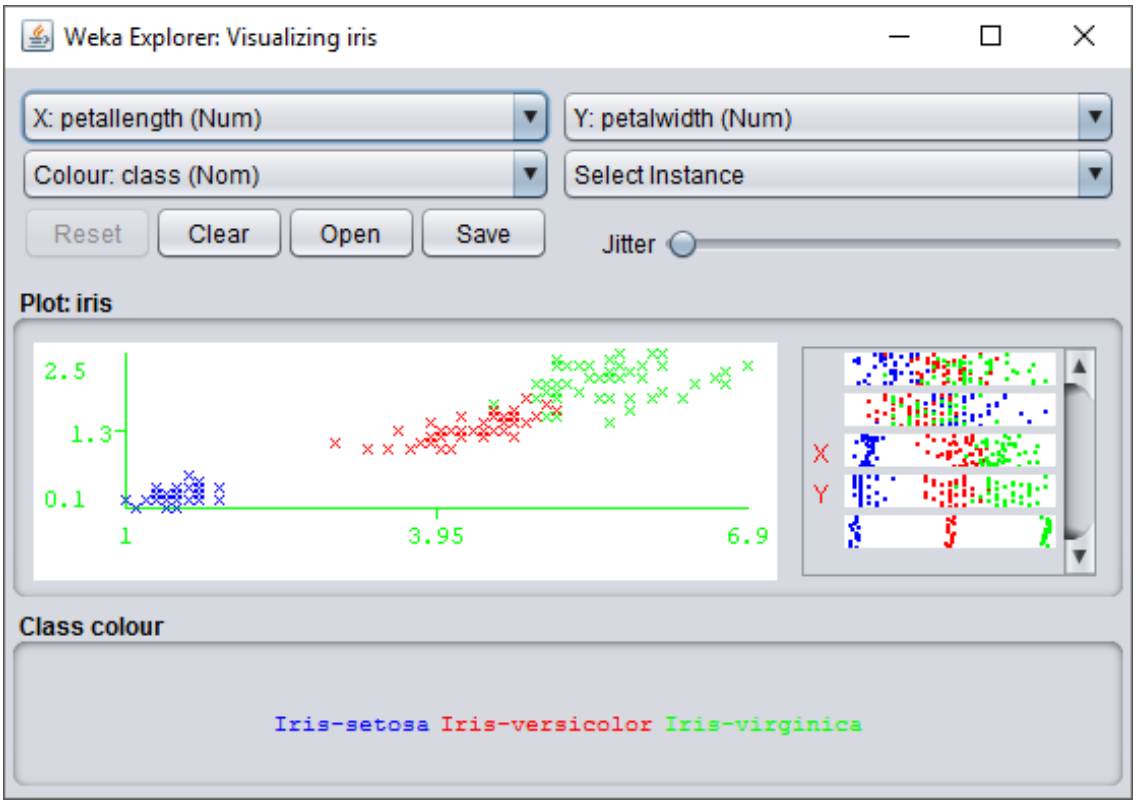
$$PetalWidth = -0.2103 \times SepalLength + 0.2288 \times SepalWidth + 0.5261 \times PetalLength - 0.2487$$

RMSE برای این مدل در صورت استفاده از روش 10 Fold مقدار ۰,۱۹۶۴ است.

خوشه‌بندی

تصویر اول از تصاویر زیر تصویر به‌دست آمده از خوشه‌بندی است و تصویر دوم نیز تصویر به دست آمده از نمایش داده‌های اصلی است. شباهت واضحی بین دو تصویر وجود دارد. هرچند که برای دو کلاسی که در مرکز و بالا سمت راست نمودار قرار گرفته‌اند در برخی موارد کلاس به درستی تشخیص داده نشده است ولی برای کلاسی که در نزدیکی مبدا قرار گرفته است خوشه‌بندی کاملا صحیح عمل کرده است.


Weka Clusterer Visualize: 23:43:51 - SimpleKMeans (iris-weka.filters.unsu...
X: petallength (Num)
Y: petalwidth (Num)
Colour: Cluster (Nom)
Select Instance
Reset
Clear
Open
Save
Jitter
Plot: iris-weka.filters.unsupervised.attribute.Remove-R5_clustered
2.5
1.3
0.1
1
3.95
6.9
X
Y
Class colour
cluster0 cluster1 cluster2


Weka Explorer: Visualizing iris
X: petallength (Num)
Y: petalwidth (Num)
Colour: class (Nom)
Select Instance
Reset
Clear
Open
Save
Jitter
Plot: iris
2.5
1.3
0.1
1
3.95
6.9
X
Y
Class colour
Iris-setosa Iris-versicolor Iris-virginica

بخش هفتم: ویژگی‌های ارزشمند

با استفاده از InfoGainAttributeEval اگر از همه مجموعه داده‌ها استفاده کنیم به ترتیب petallength و petalwidth رنک‌های ۱,۴۱۸ و ۱,۳۷۸ را به خود اختصاص می‌دهند و اگر از 10 Fold استفاده کنیم هر دو رنک ۱,۵ با خطای ۰,۵ را به دست می‌آورند. با استفاده از CorrelationAttributeEval اگر از همه مجموعه داده‌ها استفاده کنیم همان دو ویژگی مقادیر ۰,۶۱۵ و ۰,۵۹۲ و اگر از 10 Fold استفاده کنیم مقادیر ۰,۶۱۵ و ۰,۵۹۲ با خطای ۰,۰۰۲ را به خود اختصاص می‌دهند.