گزارش فاز دوم پروژه ماشین لرنینگ
پیدا کردن ستارگان متغیر کهکشان M33

امیررضا بهرامی مقدس
یوسف جمشیدی
سوگل سنجری پور

در این فاز قصد داریم الگوریتم های متفاوت ماشین لرنینگ را روی داده هایمان که پیشتر تهیه کرده ایم به دست آوریم. ابتدا داده هایمان را که در فاز قبلی کاهش داده و داده های پرت را شناسایی و حذف کرده بودیم, مجدد به ترتیب کاهشی ایندکس L(معیاری از متغیر بودن یا نبودن ستاره) مرتب کردیم. در این مرحله دیتا را محدودتر میکنیم. به این ترتیب که هر 812 ستاره متغیر را انتخاب میکنیم. از بین بقیه ستارگان نامتغیر 4000 ستاره انتخاب میکنیم. حالا کل مراحل تست و آموزش را با این دیتا ست تمیزتز شده انجام میدهیم. از کل داده ها 20% برای تست، 20% از باقی مانده برای اعتبارسنجی(validation) و بقیه برای آموزش ماشین استفاده شد. در این روند، از stratify استفاده کردیم تا در تقسیم بندی نسبت بین تعداد ستارگان متغیر و نامتغیر حفظ شود.

الگوریتم هایی که برای این فاز استفاده کردیم به ترتیب الگوریتم KNN, Random Forest و SVMو Naive Bayes.

الگوریتم KNN:

در بازشناخت الگو کی-نزدیکترین همسایه (k-nearest neighbors algorithm) :یک متد آمار پارمتری است که برای طبقه بندی آماری و رگرسیون استفاده می شود. در هر دو حالت k شامل نزدیک ترین مثال آموزشی در فضای داده ای می باشد و خروجی آن بسته به نوع مورد استفاده در طبقه بندی و رگرسیون متغیر است. در حالت طبقه بندی با توجه به مقدار مشخص شده برای کی، به محاسبه فاصله نقطه ای که میخواهیم برچسب آن را مشخص کنیم با نزدیک ترین نقاط میپردازد و با توجه به تعداد رای حداکثری این نقاط همسایه، در رابطه با برچسب نقطه مورد نظر تصمیم گیری می‌کنیم. برای محاسبه این فاصله میتوان از روش های مختلفی استفاده کرد که یکی از مطرح ترین این روش ها، فاصله اقلیدسی است. در حالت رگرسیون نیز میانگین مقادیر بدست آمده از کی خروجی آن می باشد. از آنجا که محاسبات این الگوریتم بر اساس فاصله است نرمال سازی داده‌ها می‌تواند به بهبود عملکرد آن کمک کند.

فاز یادگیری (training phase) الگوریتم، شامل ذخیره‌سازی بردارهای ویژگی و برچسب دسته نمونه‌های اولیه است. در فاز طبقه‌بندی، k یک ثابت توسط کاربر تعریف می‌شود و بردار بدون برچسب (نقطه تست) از دسته ای است که بیشترین تعداد را در k نزدیک‌ترین همسایه آن نقطه داشته باشد. به این ترتیب برچسب نقطه تست نیز مشخص می‌شود.

داده هایمان را به دو کلاس متغیر و نامتغیر تقسیم میکنیم و آنها را به صورت بردارهای 12 بعدی میبینیم. مسئله عملا 12 درجه آزادی دارد که تمام فاکتورها مستقل از هم هستند. میانگین بردارهای هر کدام از دو گروه را به دست می آوریم. در نهایت دو بردار 12 بعدی داریم به عنوان مرکز کلاس ها. سپس فاصله داده های تستمان تا این دو را میسنجیم و بقیه نمونه ها را برچسب میزنیم.

الگوریتم Random Forest:

در کل، معمولا درخت تصمیمی که بیش از حد عمیق باشد الگوی دقیق نخواهد داشت: دچار بیش برارزش شده , و دارای سوگیری پایین و واریانس بالا میباشد. جنگل تصادفی روشی است برای میانگین گیری با هدف کاهش واریانس با استفاده از درخت های تصمیم عمیقی که از قسمت های مختلف داده آموزشی ایجاد شده باشند. در این روش معمولا افزایش جزئی سوگیری و از دست رفتن کمی از قابلیت تفسیر اتفاق افتاده اما در کل عملکرد مدل را بسیار افزایش خواهد داد. در این روش با بهره گیری از معیارهایی مثل gini و entropy و به کمک مینیمم کردن آنها, پارامترهای ایده آل برای تصمیمات را انتخاب کردیم. عملا سوالهایی که کمترین احتمال ایجاد خطا در فرآیند را دارند در نظر میگیریم.

الگوریتم Naive bayes:

در یادگیری ماشین به گروهی از دسته بندی کننده های ساده بر پایه احتمالات گفته می‌شود که با فرض استقلال متغیرهای تصادفی و براساس قضیه بیز ساخته می‌شوند. به‌طور ساده روش بیز روشی برای دسته‌بندی پدیده‌ها، بر پایه احتمال وقوع یا عدم وقوع یک پدیده‌است. این روش از ساده‌ترین الگوریتم‌های پیش‌بینی است که دقت قابل قبولی هم دارد.

الگوریتم SVM:

این روش از جملهٔ روش‌های نسبتاً جدیدی است که در سال‌های اخیر کارایی خوبی نسبت به روش‌های قدیمی‌تر برای طبقه‌بندی نشان داده‌است. مبنای کاری دسته بندی کننده کنندهٔ SVM دسته بندی خطی داده ها است و در تقسیم خطی داده‌ها سعی می‌کنیم خطی را انتخاب کنیم که حاشیه اطمینان بیشتری داشته باشد. حل معادله پیدا کردن خط بهینه برای داده‌ها به وسیله روش‌های QP که روش‌های شناخته شده‌ای در حل مسائل محدودیت‌دار هستند صورت می‌گیرد. قبل از تقسیمِ خطی برای اینکه ماشین بتواند داده‌های با پیچیدگی بالا را دسته‌بندی کند داده‌ها را به وسیلهٔ تابعِ phi به فضای با ابعاد خیلی بالاتر می‌بریم. برای اینکه بتوانیم مسئله ابعاد خیلی بالا را با استفاده از این روش‌ها حل کنیم از قضیه دوگانی لاگرانژ برای تبدیلِ مسئلهٔ مینیمم سازی مورد نظر به فرم دوگانی آن که در آن به جای تابع پیچیدهٔ phi که ما را به فضایی با ابعاد بالا می‌برد، تابعِ ساده‌تری به نامِ تابع هسته که ضرب برداری تابع phi است ظاهر می‌شود استفاده می‌کنیم. از توابع هسته مختلفی از جمله هسته‌های نمایی، چند جمله ای و سیگموید می‌توان استفاده نمود. ماتریس الگو را آماده می‌کنیم تابع کرنلی را برای استفاده انتخاب می‌کنیم .پارامتر تابع کرنل و مقدار Cرا انتخاب می‌کنیم .برای محاسبهٔ مقادیر α_i الگوریتم آموزشی را با استفاده از حل کننده‌های QP

اجرا می‌کنیم .داده‌های جدید با استفاده از مقادیر α_i و بردارهای پشتیبان می‌توانند دسته بندی شوند. آموزش نسبتاً ساده است .برخلاف شبکه های عصبی در ماکزیمم محلی گیر نمی‌افتد .برای داده‌های با ابعاد بالا تقریباً خوب جواب می‌دهد .مصالحه بین پیچیدگی دسته‌بندی‌کننده و میزان خطا به‌طور واضح کنترل می‌شود .به یک تابع کرنل خوب و انتخاب پارامتر C نیاز دارد.

همچنین از این الگوریتم برای انتخاب اینکه کدام یک از الگوریتم های ما دقت بهتری داشتند استفاده کردیم. نتیجه این بود که تمامی الگوریتم ها دقت 100 داشتند به جز KNN که دقتش 99.7% بود.