Subject: AI-Hw4
Date

محمد صادقی / 810100175

Q1)

وجود نداشتن ویژگی ها: روش های پر کردن داده های ناقص مانند میانگین گیری، جایگزینی با مُد، الگوریتم های پیش بینی مانند شبکه های عصبی و رگرسیون و یا حذف داده های ناقص از دیتاست

نامتوازن بودن توزیع داده گان: استفاده از روش هایی که مبتنی بر هزینه اند و به نمونه های مختلف وزن های متفاوتی می دهند؛ یا روش های متعادل سازی مانند oversampling و undersampling می توانند به نرمال کردن توزیع داده گان در کلاس ها کمک کنند.

وجود نویز: استفاده از روش های پیش پردازش مانند فیلتر های تونیری، فیلتر میانگین، فیلتر گوس و یا بهره بردن از الگوریتم های مقاوم به نویز مانند رگرسیون و یا NN

ویژگی های همبسته: استفاده از روش های کاهش بُعد مثل PCA و یا حذف یکی از ویژگی های همبسته می تواند تاثیر این correlate را کم کند.

Subject:
Date

Q2)

با اضافه شدن یک ویژگی جدید می‌توان از رگرسیون چند جمله‌ای استفاده کرد:

نمره آزمون = B_0 + (B_1 × آزمون دادن) + (B_2 × مطالعه)

برای ضرایب مناسب دو روش وجود دارد:

1) Least Square method → $\beta = X^T y^{-1} (X^T X)$

X = ماتریس ویژگی‌ها ، y = بردار مقادیر حقیقی ، β = بردار ضرایب

2) decent Gradiant =>

در این روش از یک نقطه تصادفی شروع به update کردن ضرایب می‌کنیم تا مجموع مربعات اختلاف را کمینه کنیم! در روش اول با مشتق‌گیری نسبت به ویژگی‌ها به دنبال کمینه شدن مجموع هستیم ولی در روش دوم با آپدیت کردن مقادیر ویژگی‌ها به این minimum می‌رسیم.

روش‌های دیگر:

روش‌های رگرسیون وجود دارد که علاوه بر کمینه کردن مجموع مربعات اختلاف، به منظور پیشگیری از overfitting برای ضرایب بزرگ جریمه محاسبه می‌شود.

Subject:
Date

Q3)

	خروجی درست	خروجی غلط
حقیقی غلط	200	30
حقیقی درست	20	309

$$Recall = \frac{TP}{TP+FN} = \frac{300}{300+20} = 0,94$$

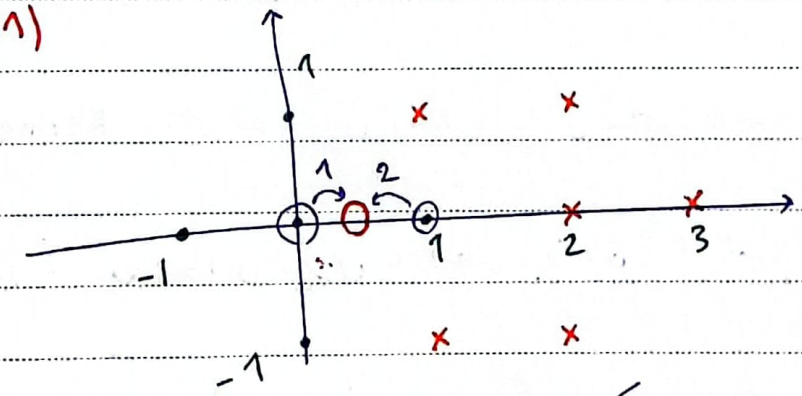
$$precision = \frac{TP}{TP+FP} = \frac{300}{300+30} = 0,91$$

$$accuracy = \frac{TP+TN}{all} = \frac{300+200}{550} = 0,91$$

$$F_1\text{-}score = 2\frac{Pr \times Re}{Pr+Re} = 2\frac{0,91 \times 0,94}{0,91+0,94} = 2\frac{0,8554}{1,85} \simeq 0,925$$

KNN

Q1)

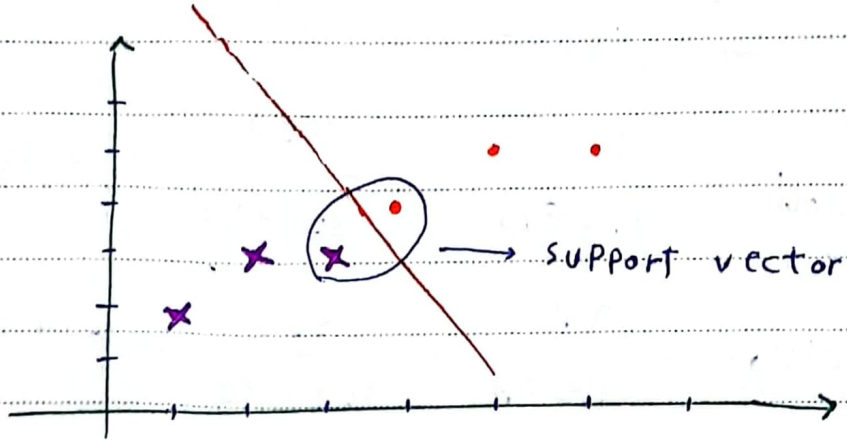


سومین همسایه هر چه باشد، چون دوتای اول در کلاس دایره هستند، داده مانیز در دسته 2 یا همان • قرار می‌گیرد!

Subject:
Date

⊕ SVM

به نقاطی که برای تعیین margin بین دسته های مختلف مورد استفاده قرار می‌گیرند support vector می گوییم:



در داده های پرتراکم با حجم زیاد، داده‌های نامتوازن و توزیع، دادگان دارای ویژگی های همبسته و پیچیده و در جاهایی که کاهش بعد منجر به از دست رفتن دیتا شود، این روش مناسب نمی باشد!

Kernel ⟸ به ما این امکان را می دهد که با استفاده از توابع مشخص، داده ها را به یک فضای ویژگی با ابعاد بالا نگاشت کنیم به طوری که دادگان در این فضا بصورت خطی جدا پذیر شوند؛ سپس SVM را روی داده های نگاشت شده اعمال می‌کنیم. در واقع یک خط در ابعاد بالا، برابر یک نقطه در فضای 3 بعدی است!

Subject:
Date

① Hard-SVM ⟹ هیچ داده ای در margin قرار نمی گیرد و فرض می شود

داده ها بصورت خطی جدا پذیرند؛ اگر واقعا خطی جدا نشوند

این نوع SVM دچار overfitting خواهد شد.

② Soft-SVM ⟹ برخلاف قبل اجازه می دهد در margin داده قرار

بگیرد و محدوده آنها طوری تغییر می دهد که پس از

classification مقدار خطا و overfitting به حداقل برسد و طبعا یک سری داده را

نویز و داخل margin تشخیص می دهد.