Maskininlärning Lab PM

IT Högskolan
raphael.korsoski@iths.se

Introduktion

Denna labb består av två moment, ett för att få Godkänt och ett för att få Väl Godkänt. Det första momentet innefattar att använda maskininlärningstekniker och skriva en enklare teknisk rapport som en jupyter notebook kring ett givet problem. Det momentet fokuserar på utförandet och bedöms endast med Godkänt efter godkänd inlämning.

Tänk på att en teknisk rapport inte skrivs i den ordning den presenteras. Kronologin hur resultaten nåddes är inte viktigt utan rapporten skrivs som om resultaten var kända hela tiden. Läs alltså igenom hela PM:et och gör uppgifterna i den ordning som gör det lättast att komma vidare. Fyll i de olika avsnitten i rapporten allt eftersom ny information blir tillgänglig.

Rapporten skall innehålla en data-analys, en modelldesign inklusive hyperparameteroptimisering, utvärdering och slutligen en paketering och demo. Dessa motsvarar ungefär avsnitten i IMRaD metoden.

Det andra momentet innefattar en mer utforskande och experimentell aktivitet som ger möjlighet till kreativitet och självständig utveckling. Alternativt kan en vedertagen teknik (content filtering eller collaborative filtering) undersökas. Det momentet bedöms endast för det högre betyget VG och lägger tyngd på arbetsinsatsen och ambitionsnivån. Självständigt arbete är här avgörande. Systemets effektivitet är inte det viktiga, utan viktigt är data-analysen och att resonera kring utfallet.

1 Klassificering av hjärt- och kärlsjukdom (G)

I det här momentet används ett dataset med data för hjärt-kärlsjukdom. Börja med att ladda ned datamängden från Kaggle och läs beskrivningen av innehållet. Notera att denna datamängd innehåller många felaktigheter, exempelvis finns negativa blodtryck och blodtryck som är omöjligt höga.

1.1 EDA

Använd pandas, matplotlib och seaborn för att besvara följande frågor för datasetet:

a) Hur många är positiva för hjärt-kärlsjukdom och hur många är negativa?

b) Hur stor andel har normala, över normala och långt över normala kolesterolvärden?

c) Hur ser åldersfördelningen ut?

d) Hur stor andel röker?

e) Hur ser viktfördelningen ut?

f) Hur ser längdfördelningen ut?

g) Hur stor andel av kvinnor respektive män har hjärt-kärlsjukdom?

Sammanfatta dina resultat och presentera dem i en notebook. Använd markdown boxar för löpande text.

1.2 Modelldesign

· Skapa en heatmap av korrelationer i datan.

Se om du hittar features som är starkt korrelerade, dvs betydligt skiljt från 0 med positiv tal eller features som är starkt negativt korrelerade. Detta är en ledning för kommande uppgifter och är en del av att designa modellen, men skall presenteras som del av data-analysen. Lägg till heatmap och resonera kring vad den visar tillsammans med den tidigare data-analysen du gjort.

1.2.1 Feature Engineering

I detta avsnitt skall du dokumentera vad du *gjorde* i löpande text. Ta inte med misstag eller sidospår, utan bara vad som faktiskt ledde till det senare resultatet.

Skapa en feature för BMI (Body Mass Index), läs på om formeln på wikipedia.

a) Släng de samples med orimliga BMIer och outliers. Notera att detta kan vara svårt att avgöra i vilket range av BMIer som vi ska spara. Beskriv hur du kommer fram till gränserna, med resonemang eller referens.

b) Skapa en kategorisk BMI-feature med kategorierna: normal range, overweight, obese (class I), obese (class II), obese (class III).

c) Undersök om kategorin är relevant, dvs dess korrelationer. Uppdatera data-analysen om du hittar något intressant.

Skapa en feature för blodtryckskategorier enligt tabellen i denna artikel.

a) Släng bort samples med orimliga blodtryck och outliers. Likt förra uppgiften är det inte trivialt att sätta gränserna. Beskriv hur du kommer fram till gränserna.

b) Skapa en kategorisk feature med relevanta kategorier.

c) Undersök om den nya kategorin är relevant, dvs har den någon nyttig korrelation? Uppdatera data-analysen om du hittar något intressant.

Tips: Efter du valt gränser och skapat kategorier, kolla vilka kategorier som faktiskt förekommer i din data.

1.2.2 Skapa två dataset

Skapa en kopia av din dataframe.

- På ena dataframen: ta bort följande features: `ap_hi, ap_lo, height, weight, BMI` och gör one-hot encoding på BMI-kategori, blodtryckskategori samt kön
- På andra dataframen: ta bort följande features: `BMI-kategori, blodtryckskategori, height, weight` och gör one-hot encoding på kön

Alltså en datamängd med kategorisk data tillagd och en med endast BMI tillagd.

1.2.3 Utförande

Välj tre eller fler algoritmer. För varje algoritm:

a) Använd gridsearchCV för att skala och hyperparameteroptimisera varje algoritm

b) Utvärdera resulterande modell

c) Kolla hyperparametrarna som ledde till bäst resultat

d) Samla data om utfallet för senare presentation

Upprepa detta för båda datamängderna. Välj datamängd och modell utifrån dina resultat.

Dokumentera resultaten och motivera valen.

1.3 Paketering och demo

1.3.1 Spara modell

Börja med att plocka ut 100 slumpmässigt valda rader från ditt dataset. Exportera dessa 100 samples i `test_samples.csv`. Därefter tar du den bästa modellen och träna på all data vi har förutom de 100 datapunkterna du plockade ut. Spara därefter modellen i en .pkl-fil med hjälp av `joblib.dump()`. För modellen kan du behöva använda argumentet `compress` för att komprimera om filstorleken för stor.

1.3.2 Ladda modellen

Skapa ett nytt skript: `production_model.py`, ladda in `test_samples.csv` och din modell. Använd `joblib.load()` för att ladda in en .pkl-fil. Gör prediction på de 100 datapunkterna och exportera en fil `prediction.csv` som ska innehålla kolumnerna med ifyllda värden:

probability class 0

probability class 1

prediction

2 Filmrekommendationer (VG)

Rekommendationssystem är en idag avgörande funktion för sociala medier och distribution av musik, film, spel med mera. Denna uppgift ger möjlighet till självständigt utforskande av problemet att rekommendera filmer. Dokumentera ditt arbete och hur du tänker; bara en hög kod räcker inte. Denna uppgift innebär en hel del research på egen hand. Du kommer även behöva göra avgränsningar för att hinna på utsatt tid. För att utföra detta moment, välj en av följande:

- Gör en utforskande analys av movielens datan, välj några begränsningar och implementera en egen modell. Behöver användaren mata in egna preferenser eller kan den rekommendera "blint" givet några exempel? (**)
- Beskriv och implementera en vedertagen teknik för rekommendationer. De två dominerande är content-filtering och collaborative-filtering. (***)

Antalet (*) är min uppskattning av hur mycket arbete vardera uppgift innebär.

2.1 Förberedelser

Ladda ned ml-latest.zip under sektionen"recommended for education and development" från movielens.

Undersök datamängden. Filerna `genome_*` är utdatan från ett annat maskininlärningsprojekt och kan ignoreras för denna uppgift. Filen `link.csv` innehåller korsreferenser mellan olika filmsajter och kan också ignoreras.

Filmer och användare har unika identifierare i datan och korsrefereras mellan filerna. Fundera på hur du vill bygga en gemensam datamängd att använda.

Undersök `movies.csv` filen och särskilt kolumnen `genres`. Fundera på vad den innebär med avseende på likhet och hur den kan kategoriseras.

Undersök `ratings.csv` och fundera över vad distributionen av värden innebär. *Tips:* Gruppera efter användare

Undersök `tags.csv` och fundera på om du kan använda den datan på något sätt.

2.2 Utförande

Kom ihåg att dokumentera ditt arbete och att i slutändan samanfatta det som är relevant. Precis som i första uppgiften så är inte kronologin det intressanta utan vad de faktiska resultaten blev. Skriv som om du visste hela tiden vad utfallet skulle bli.

Det viktiga för betygsättningen är inte "rätt svar". Datapunkterna och typerna är högst subjektiva i denna uppgift. För att få det högre betyget skall du visa självständighet, förståelse och utförlighet. Glöm inte att vara noggrann med hänvisningar om du använt kod eller formler som du inte skrivit själv.