

Examen Sorpresa: Optimització en Machine Learning

Instruccions Generals

1. Aquest qüestionari consisteix en dues tasques relacionades amb l'optimització en Machine Learning: implementació de **Gradient Descent** (Descens del Gradient) i **Grid Search** (Cerca Exhaustiva).
2. Cada estudiant treballarà amb un **fitxer de dades únic** que ja us ha estat proporcionat (e.g., `student_12345_data.csv`).
3. Heu de completar les tasques següents i respondre les preguntes incloses al final de cada tasca.
4. Envieu:
 - El codi Python complet i funcional.
 - Les gràfiques requerides.
 - Les respostes a les preguntes reflexives.
5. Podeu utilitzar llibreries com NumPy, Matplotlib i Scikit-learn.

Tasca 1: Descens del Gradient per Regressió Lineal

Descripció del problema:

- Utilitzeu el **Descens del Gradient** per ajustar un model de regressió lineal al vostre conjunt de dades.
- L'objectiu és minimitzar l'**error quadràtic mitjà (MSE)** entre els valors previstos i els valors reals.

Passos a seguir:

1. Carregueu el vostre fitxer de dades (e.g., `student_12345_data.csv`) i dividiu-lo en X (característiques) i y (objectiu).
2. Afegiu un terme de biaix (*bias*) a les característiques (X).
3. Implementeu l'algorisme de Descens del Gradient:
 - Comenceu amb valors aleatoris per als paràmetres (θ).

- Actualitzeu iterativament els paràmetres per minimitzar el MSE.
- Utilitzeu una taxa d'aprenentatge (η) i un nombre màxim d'iteracions.

4. Mostreu una gràfica:

- Punts de dades (\mathbf{X} vs. \mathbf{y}).
- La línia de regressió ajustada (valors previstos pel model).

Lliurables:

- Codi Python per la implementació del Descens del Gradient.
- Gràfica que mostra la línia de regressió.
- Responen les preguntes següents:

Preguntes:

1. Quin és el valor final de l'**MSE** després d'executar el Descens del Gradient?
2. Com afecta la **taxa d'aprenentatge** (η) al procés d'optimització?
3. Què passa si reduïu el nombre màxim d'iteracions?
4. Per què és necessari afegir un terme de biaix (*bias*)?

Tasca 2: Cerca Exhaustiva (Grid Search) per a SVM

Descripció del problema:

- Utilitzeu **Grid Search** per optimitzar els hiperparàmetres d'un classificador SVM.
- Converteix el teu conjunt de dades de regressió en un problema de classificació binària:
 - Assigna un valor objectiu de 1 si l'objectiu original (\mathbf{y}) és superior a la mediana.
 - Assigna un valor objectiu de 0 si és inferior o igual a la mediana.

Passos a seguir:

1. Carregueu el vostre fitxer de dades i preproceseu-lo per a classificació binària.
2. Dividiu les dades en conjunts d'entrenament i prova (80% entrenament, 20% prova).
3. Utilitzeu `GridSearchCV` per trobar la millor combinació d'hiperparàmetres (\mathbf{C} i `kernel`):
 - \mathbf{C} : {0.1, 1, 10}.
 - `kernel`: {'linear', 'rbf'}.
4. Entreneu el model SVM amb els millors paràmetres i evalueu-ne el rendiment.
5. Mostreu una gràfica:

- La frontera de decisió del SVM ajustat.

Lliurables:

- Codi Python per la implementació de Grid Search.
- Gràfica que mostra la frontera de decisió del classificador.
- Responen les preguntes següents:

Preguntes:

1. Quins són els millors hiperparàmetres seleccionats per Grid Search?
2. Quina és l'**exactitud (accuracy)** del model al conjunt de prova?
3. Com afecta l'increment de la mida de la graella (més valors de \mathbb{C}) al temps de càlcul?
4. Expliqueu la diferència entre els kernels lineal i RBF. Quin ha tingut millor rendiment i per què?

Criteris d'Avaluació

- **Correcció del codi (40%):**
 - Implementació correcta de Descens del Gradient i Grid Search.
 - Resultats precisos.
- **Gràfiques i visualitzacions (20%):**
 - Línia de regressió i frontera de decisió representades correctament.
- **Respostes reflexives (30%):**
 - Respostes que mostrin una comprensió clara dels conceptes.
- **Qualitat del codi (10%):**
 - Codi ben documentat, llegible i modular.