

Scommesse - Guida passo-passo (macOS, principianti)

Questa guida ti accompagna, senza tecnicismi inutili, a usare il mini-motore di scommesse che abbiamo creato. È pensata per chi usa **macOS** e **Visual Studio Code (VS Code)**.

1) Cosa c'è nel progetto (panoramica semplice)

File principali - `README_Scommesse.md` - la spiegazione rapida del flusso. - `config.json` - impostazioni (soglie, stake, blending). - `fixtures.csv` - partite del giorno + quote/linee. - `features.csv` - fattori (xG, infortuni, calendario, meteo...). - `predictions.csv` - le probabilità del tuo modello (1, X, 2, Over/Under). - `scommesse_pipeline.py` - genera le **pick**, aggiorna **metriche** e crea **report HTML**. - `bets_log.csv` - registro giocate (cosa, quota, stake, esito, CLV...). - `metrics_daily.csv` - performance per giorno (ROI, Brier, Logloss...). - `manage_scommesse_csv.py` - **menu interattivo** per svuotare/riempire i CSV. - `Makefile` - scorciatoie (venv, install, run, metrics, report...). - `.github/workflows/scommesse.yml` - automazione su GitHub (opzionale).

2) Preparazione ambiente su macOS (senza stress)

2.1 Installa Visual Studio Code

1. Vai su code.visualstudio.com e scarica **Visual Studio Code**.
2. Apri l'app e consenti l'apertura se macOS chiede conferma.

2.2 Apri la cartella del progetto

1. In VS Code: **File → Open Folder...**
2. Seleziona la cartella dove hai messo i file (`README_Scommesse.md`, `scommesse_pipeline.py`, ecc.).

2.3 Installa l'estensione Python in VS Code

- In VS Code: **Extensions** (icona quadratini) → cerca **"Python" (Microsoft)** → **Install**.
- Facoltativo ma utile: **Pylance**.

2.4 Crea l'ambiente Python (virtualenv)

Apri il **Terminale integrato**: **View → Terminal** e digita:

```
python3 -m venv .venv
source .venv/bin/activate
pip install --upgrade pip
```

Se preferisci usare il Makefile (più semplice):

```
make venv
```

2.5 Installa le dipendenze

Con l'ambiente attivo:

```
pip install pandas numpy matplotlib
```

oppure, via Makefile:

```
make install
```

2.6 Seleziona l'interprete Python giusto in VS Code

In basso a destra (o in Command Palette  → "Python: Select Interpreter"), scegli **.venv/bin/python**.

Se vedi errori "module not found", hai probabilmente il venv disattivato: riesegui `source .venv/bin/activate` e riprova.

3) Gestione rapida dei CSV con menu (consigliato)

Il menu evita errori di formattazione.

3.1 Avvio menu CSV

Nel terminale VS Code (con venv attivo):

```
python manage_scommesse_csv.py
```

Vedrai un menu con opzioni: - 1: backup dei CSV - 2: svuota `fixtures.csv` - 3: svuota `features.csv` - 4: aggiungi righe a `fixtures.csv` - 5: aggiungi righe a `features.csv` - 6: riempi esempi DEMO (2 match d'esempio) - 7: mostra anteprima dei file - 0: esci

3.2 Comandi rapidi (senza menu)

- Svuota entrambi i CSV, ma tieni le intestazioni:

```
python manage_scommesse_csv.py --clear
```

- Inserisci **dati demo** pronti (utile per fare prove):

```
python manage_scommesse_csv.py --demo
```

Consiglio: usa **prima** `--demo`, prova la pipeline, poi svuota e inserisci i tuoi incontri reali.

4) Come compilare i CSV (senza sbagliare)

4.1 fixtures.csv (partite e quote)

Colonne chiave - `match_id` (generato automaticamente dal menu quando inserisci una partita) → formato consigliato: `YYYYMMDD_HOME_AWAY_LEAGUE` (maiuscolo, spazi con `_`). - `league` (es. "Serie A"). - `date` e `time_local` (formato `YYYY-MM-DD`, `HH:MM`). - `home`, `away` (nomi squadre coerenti tra i file). - `odds_1`, `odds_x`, `odds_2` (quote 1X2 al momento della **presa**). - `line_ou` (es. 2.5 o 3.0) + `odds_over`, `odds_under`.

4.2 features.csv (fattori)

Compila almeno: - `xG rolling`: `xG_for_5_home`, `xG_against_5_home`, `xG_for_5_away`, `xG_against_5_away` - riposo: `rest_days_home`, `rest_days_away` - infortuni chiave (flag 0/1): `injuries_key_home`, `injuries_key_away` - contesto: `derby_flag` (0/1), `europe_flag_home` / `away` (0/1), `meteo_flag` (0/1) - stile: `style_ppda_home`, `style_ppda_away` - viaggio: `travel_km_away`

Le colonne devono combaciare **esattamente** con l'intestazione: il menu e gli script si aspettano quei nomi.

4.3 predictions.csv (le tue probabilità)

- `p1`, `px`, `p2` **devono sommare ≈ 1** (sono probabilità). Suggerimento: 0.54/0.26/0.20 ecc.
- Per i totali: `mu_gol` (media gol), `sigma_gol` (dev. standard), `p_over`, `p_under`.

5) Generare le selezioni (pick), metriche e report

5.1 Crea selezioni (aggiorna `bets_log.csv`)

```
python scommesse_pipeline.py --run
```

Cosa succede: - Confronta le tue probabilità con il mercato (**edge**). - Sceglie il **mercato migliore** (DNB/DC/1X2/UO) secondo le soglie di `config.json`. - Stima lo **stake** (flat; Kelly frazionato solo quando ha senso). - Scrive tutto nel **registro** `bets_log.csv`.

5.2 Aggiorna metriche (ROI, Brier, Logloss)

Dopo aver inserito in `bets_log.csv` i risultati finali (`risultato` tipo `2-1` e `esito` W/L/Push):

```
python scommesse_pipeline.py --update-metrics
```

Aggiorna `metrics_daily.csv` con: - **ROI** (ritorno sull'investimento) - **Brier 1X2** (quanto le probabilità 1X2 erano ben calibrate) - **Logloss O/U** (quanto le probabilità Over/Under erano ben calibrate)

5.3 Crea il report HTML

```
python scommesse_pipeline.py --report
```

Genera `report_scommesse.html` con: - Tabella delle **ultime pick** - Grafico **profitto cumulato** - Grafico **CLV rolling** (se hai anche `quota_closing`) - Tabella **metriche giornaliere**

Il report è un file HTML: doppio click per aprirlo nel browser.

6) Scorciatoie con Makefile (facile facile)

Se non vuoi ricordare i comandi lunghi, usa:

```
make venv      # crea l'ambiente .venv
make install   # installa le librerie
make run        # selezioni → bets_log.csv
make metrics    # aggiorna ROI/Brier/Logloss
make report     # genera report_scommesse.html
make clear      # svuota i CSV con intestazioni
make demo       # carica esempi pronti
```

7) Automazione su GitHub (opzionale)

Se usi GitHub: 1. Fai commit dei file nella tua repo. 2. Il workflow `.github/workflows/scommesse.yml` esegue automaticamente `run → metrics → report` ad ogni **push su main** (o manualmente via *Run workflow*). 3. Troverai il report come **artifact** scaricabile nella pagina delle Actions.

Serve avere i file nella radice della repo o aggiornare i percorsi nel workflow.

8) Risoluzione problemi comuni

- **"command not found: python3"** → Installa Xcode Command Line Tools: apri Terminale e digita `xcode-select --install`, poi riprova.
- **"ModuleNotFoundError: pandas"** → l'ambiente non è attivo. Esegui `source .venv/bin/activate` e `pip install pandas numpy matplotlib`.
- **Quote non numeriche** → nei CSV usa solo numeri (punto decimale), niente simboli.
- **Le somme p1+px+p2 ≠ 1** → ribilancia le probabilità (devono approssimare 1).

- **Il report è vuoto** → probabilmente non hai ancora esiti in `bets_log.csv` o `metrics_daily.csv` è vuoto: esegui `--update-metrics` dopo aver inserito i risultati.
-

9) Glossario mini (senza parolone)

- **Edge**: vantaggio stimato rispetto alle quote del mercato.
 - **DNB (Draw No Bet)**: rimborso in caso di pareggio.
 - **DC (Doppia chance)**: copri due risultati (1X, X2, 12).
 - **Brier score**: misura quanto sono “oneste” le tue probabilità 1X2 (più basso è meglio).
 - **Logloss**: misura la bontà delle probabilità per eventi binari (tipo Over/Under) (più basso è meglio).
 - **CLV (Closing Line Value)**: quanto la tua quota iniziale batte la quota di chiusura (se sì, stai leggendo bene il mercato).
-

10) Checklist giornata tipo

1. **Apri venv**: `source .venv/bin/activate`
2. **Pulisci o carica demo**: `python manage_scommesse_csv.py --clear` oppure `--demo`
3. **Inserisci incontri reali con il menu**
4. **Compila features & predictions**
5. **Genera pick**: `python scommesse_pipeline.py --run`
6. **A fine partite**: aggiorna esiti/chiusure in `bets_log.csv`
7. **Metriche**: `python scommesse_pipeline.py --update-metrics`
8. **Report**: `python scommesse_pipeline.py --report`

Mantieni ordine nei CSV. Il computer è obbediente, ma non legge nel pensiero.