

Interpolation Tables Examples

Interpolation Tables Examples

Файлы: `examples/interpolation_tables/interpolation_table.py`, `models_evaluation.py`,
`models_evaluation_ray.py`.

“ **Проверено на локальном Docker-стеке:** все три скрипта
отрабатывают до конца. Недостающие данные и ошибка выбора метрики
(см. ниже) исправлены.

?????????: ?????????????? ????
??????

Каждый скрипт читает `examples/interpolation_tables/data/table_example.csv`:

```
csv_path = os.path.join(os.path.join(os.path.dirname(__file__), "data"), "table_example.csv")
df = pd.read_csv(csv_path, sep=";")
```

Папки `examples/interpolation_tables/data/` в репозитории не было. Добавлен
`examples/interpolation_tables/data/table_example.csv` — регулярная сетка 5×4 (20 строк): `x` ∈
{1000, 3000, 5000, 7000, 9000}, `y` ∈ {200, 600, 1000, 1400} (охватывает точку запроса
(4723, 937) из `interpolation_table.py` и даёт минимум 4 точки на ось, нужные
`RegularGridInterpolator` с `method="cubic"`), `z` — гладкая функция от `x/y`, плюс колонка
`target` (для `models_evaluation*.py`, где `X = df[["x", "y", "z"]]`, `y = df["target"]`) с небольшим
шумом.

interpolation_table.py

```
interp = TableInterpolator(df, x_col="x", y_col="y", z_col="z")
value = interp.get(4723, 937)
Xg, Yg, Zg = interp.get_grid(step_x=50, step_y=50)
```

```
interp.plot_surface(step_x=50, step_y=50)
```

Прямое повторение примера из README/документации модуля — единственный кусок кода в проекте, который дословно совпадает с `make_docs/source/introduction/what_is_mlumina.rst`.

`models_evaluation.py` — ?????????? ????????? ??? Ray

Последовательно демонстрирует все функции модуля `interpolation_tables.models_evaluation`, на train/test-разбиении первых 15 / оставшихся строк CSV:

1. `evaluate_model(LinearRegression(), X_train, y_train, X_test, y_test)` — обучение + метрики одной модели.
2. `grid_search_model(DecisionTreeRegressor, {"max_depth": (2,5,1), "max_features": (1,3,1)}, ..., optimize_metric="mse")` — перебор сетки для одного класса модели.
3. `compare_multiple_models([...], ..., optimize_metric="mse")` — сравнение 4 моделей (SVR, `DecisionTreeRegressor`, `LinearRegression`, `XGBRegressor`), каждая со своей сеткой гиперпараметров; замеряется время выполнения.
4. `hyperopt_search_model(DecisionTreeRegressor, {"max_depth": (2,10,1), "min_samples_split": (2,20,1)}, ..., max_evals=40, optimize_metric="mse")` — ТРЕ-поиск вместо полного перебора.

`models_evaluation_ray.py` — ????????????????????

То же самое, но `compare_multiple_models_ray([...], ..., optimize_metric="mse")` — каждая из 4 моделей оптимизируется в отдельной Ray-задаче.

?????????????: ?????????? ?????????? ????????????????

Раньше пример использовал `optimize_metric="r2"`. Но `grid_search_model` / `compare_multiple_models_ray` внутри всегда выбирают результат с **наименьшим** значением метрики (`if score < best_score`) — это корректно для `mse` / `rmse` / `mae` (меньше — лучше), но **не корректно для `r2`** (для R^2 лучше — *больше*, в идеале ближе к 1; с `"r2"` выбиралась бы модель с наименьшим/наиболее отрицательным R^2 , т.е. фактически худшая по этой метрике). Заменено на `optimize_metric="mse"` — с ним обе версии (без Ray и с Ray) стабильно

выбирают одну и ту же (действительно лучшую) модель `LinearRegression`.

Ни `grid_search_model`, ни `compare_multiple_models`/`compare_multiple_models_ray` по-прежнему не поддерживают направление оптимизации (`mode="min"/"max"`) — используйте только `mse`/`rmse`/`mae`-подобные метрики в `optimize_metric`, не `r2`.

??? ????????

```
python examples/interpolation_tables/interpolation_table.py
python examples/interpolation_tables/models_evaluation.py
python examples/interpolation_tables/models_evaluation_ray.py # требует ray.init() внутри
скрипта – работает и локально (без кластера)
```

Revision #2

Created 2026-09-23 06:54:42 UTC by Маслов Алексей

Updated 2026-09-23 07:53:06 UTC by Маслов Алексей