assignment_2

June 21, 2025

1 Assignment 2

In this assigment, we will work with the *Forest Fire* data set. Please download the data from the UCI Machine Learning Repository. Extract the data files into the subdirectory: `../data/fires/` (relative to `./05_src/`).

1.1 Objective

- The model objective is to predict the area affected by forest fires given the features set.
- The objective of this exercise is to assess your ability to construct and evaluate model pipelines.
- Please note: the instructions are not meant to be 100% prescriptive, but instead they are a set of minimum requirements. If you find predictive performance gains by applying additional steps, by all means show them.

1.2 Variable Description

From the description file contained in the archive (`forestfires.names`), we obtain the following variable descriptions:

1. X - x-axis spatial coordinate within the Montesinho park map: 1 to 9
2. Y - y-axis spatial coordinate within the Montesinho park map: 2 to 9
3. month - month of the year: "jan" to "dec"
4. day - day of the week: "mon" to "sun"
5. FFMC - FFMC index from the FWI system: 18.7 to 96.20
6. DMC - DMC index from the FWI system: 1.1 to 291.3
7. DC - DC index from the FWI system: 7.9 to 860.6
8. ISI - ISI index from the FWI system: 0.0 to 56.10
9. temp - temperature in Celsius degrees: 2.2 to 33.30
10. RH - relative humidity in %: 15.0 to 100
11. wind - wind speed in km/h: 0.40 to 9.40
12. rain - outside rain in mm/m2 : 0.0 to 6.4
13. area - the burned area of the forest (in ha): 0.00 to 1090.84

1.2.1 Specific Tasks

- Construct four model pipelines, out of combinations of the following components:
  - Preprocessors:
    - A simple processor that only scales numeric variables and recodes categorical variables.

        * A transformation preprocessor that scales numeric variables and applies a non-linear transformation.
    - Regressor:
        * A baseline regressor, which could be a K-nearest neighbours model or a linear model like Lasso or Ridge Regressors.
        * An advanced regressor of your choice (e.g., Bagging, Boosting, SVR, etc.). TIP: select a tree-based method such that it does not take too long to run SHAP further below.
- Evaluate tune and evaluate each of the four model pipelines.
    - Select a performance metric out of the following options: explained variance, max error, root mean squared error (RMSE), mean absolute error (MAE), r-squared.
    - *TIPS*:
        * Out of the suggested metrics above, some are correlation metrics, but this is a prediction problem. Choose wisely (and don't choose the incorrect options.)
- Select the best-performing model and explain its predictions.
    - Provide local explanations.
    - Obtain global explanations and recommend a variable selection strategy.
- Export your model as a pickle file.

You can work on the Jupyter notebook, as this experiment is fairly short (no need to use sacred).

2 Load the data

Place the files in the ../../05_src/data/fires/ directory and load the appropriate file.

[ ]:
```
# Load the libraries as required.
```

[ ]:
```
# Load data
columns = [
    'coord_x', 'coord_y', 'month', 'day', 'ffmc', 'dmc', 'dc', 'isi', 'temp',␣
 ↪'rh', 'wind', 'rain', 'area'
]
fires_dt = (pd.read_csv('../../05_src/data/fires/forestfires.csv', header = 0,␣
 ↪names = columns))
fires_dt.info()
```

3 Get X and Y

Create the features data frame and target data.

[ ]:
```

```

[ ]:

4 Preprocessing

Create two Column Transformers, called preproc1 and preproc2, with the following guidelines:

- Numerical variables
  - (Preproc 1 and 2) Scaling: use a scaling method of your choice (Standard, Robust, Min-Max).
  - Preproc 2 only:
    * Choose a transformation for any of your input variables (or several of them). Evaluate if this transformation is convenient.
    * The choice of scaler is up to you.
- Categorical variables:
  - (Preproc 1 and 2) Apply one-hot encoding where appropriate.
- The only difference between preproc1 and preproc2 is the non-linear transformation of the numerical variables.

4.0.1 Preproc 1

Create preproc1 below.

- Numeric: scaled variables, no other transforms.
- Categorical: one-hot encoding.

[ ]:

4.0.2 Preproc 2

Create preproc1 below.

- Numeric: scaled variables, non-linear transformation to one or more variables.
- Categorical: one-hot encoding.

[ ]:

4.1 Model Pipeline

Create a model pipeline:

- Add a step labelled `preprocessing` and assign the Column Transformer from the previous section.
- Add a step labelled `regressor` and assign a regression model to it.

4.2 Regressor

- Use a regression model to perform a prediction.
  - Choose a baseline regressor, tune it (if necessary) using grid search, and evaluate it using cross-validation.
  - Choose a more advance regressor, tune it (if necessary) using grid search, and evaluate it using cross-validation.
  - Both model choices are up to you, feel free to experiment.

[ ]:
```
# Pipeline A = preproc1 + baseline
```

[ ]:
```
# Pipeline B = preproc2 + baseline
```

[ ]:
```
# Pipeline C = preproc1 + advanced model
```

[ ]:
```
# Pipeline D = preproc2 + advanced model
```

5 Tune Hyperparams

- Perform GridSearch on each of the four pipelines.
- Tune at least one hyperparameter per pipeline.
- Experiment with at least four value combinations per pipeline.

[ ]:

[ ]:

[ ]:

[ ]:

[ ]:

6 Evaluate

- Which model has the best performance?

7 Export

- Save the best performing model to a pickle file.

[ ]:

[ ]:

8 Explain

- Use SHAP values to explain the following only for the best-performing model:
  - Select an observation in your test set and explain which are the most important features that explain that observation's specific prediction.
  - In general, across the complete training set, which features are the most and least important.
- If you were to remove features from the model, which ones would you remove? Why? How would you test that these features are actually enhancing model performance?

[ ]:

[ ]:

(Answer here.)

8.1 Criteria

The rubric contains the criteria for assessment.

8.2 Submission Information

Please review our Assignment Submission Guide for detailed instructions on how to format, branch, and submit your work. Following these guidelines is crucial for your submissions to be evaluated correctly.

8.2.1 Submission Parameters:

- Submission Due Date: `HH:MM AM/PM - DD/MM/YYYY`
- The branch name for your repo should be: `assignment-2`
- What to submit for this assignment:
  - This Jupyter Notebook (assignment_2.ipynb) should be populated and should be the only change in your pull request.
- What the pull request link should look like for this assignment: `https://github.com/<your_github_username>/production/pull/<pr_id>`
  - Open a private window in your browser. Copy and paste the link to your pull request into the address bar. Make sure you can see your pull request properly. This helps the technical facilitator and learning support staff review your submission easily.

Checklist: - [ ] Created a branch with the correct naming convention. - [ ] Ensured that the repository is public. - [ ] Reviewed the PR description guidelines and adhered to them. - [ ] Verify that the link is accessible in a private browser window.

If you encounter any difficulties or have questions, please don't hesitate to reach out to our team via our Slack at the `help` channel. Our Technical Facilitators and Learning Support staff are here to help you navigate any challenges.

9 Reference

Cortez,Paulo and Morais,Anbal. (2008). Forest Fires. UCI Machine Learning Repository. https://doi.org/10.24432/C5D88D.