University of Glasgow

Adam Smith Business School

Coursework Briefing 2022/23 (group, written)

Coursework information

Course Co-ordinator to complete all details below:

Course Code	*ECON5130*
Course Title	Machine Learning in Finance with Python
Couse Coordinator	Ankush Agarwal, Richard Foltyn
Second Internal Checker (from WLM)	Ankush Agarwal, Richard Foltyn
Coursework format	*Group programming-based project*
Weighting	25%
Word limit	*NA*
Action to be taken if word limit is exceeded	*NA*
Submission date	*December 2, 12:00*
Number of Student per Group (max)	*3-4*
Date Question to be posted on Moodle	*November 18*

Question

See next page.

Group project

Machine Learning in Finance with Python (ECON5130)

Richard Foltyn
University of Glasgow

Deadline: December 2, 12:00

Predicting house prices with linear models

In this project, you will work with the Ames house data set which we already encountered in the lectures. Your task is to evaluate the following four linear models in terms of their performance when predicting house prices:

1. Linear regression
2. Ridge regression
3. Lasso
4. Elastic net

General hints:

1. Clearly label all graphs (axes, title, legend if required).
2. When asked to provide a specific answer (e.g., "Report the number of non-zero coefficients...") make sure the answer is clearly printed in the notebook.
3. Whenever a computation involves random number generation, initialise the seed to `123` to get reproducible results. Specifically, for `scikit-learn` functions this requires passing `random_state=123` where applicable.

Data description

The data is stored in `data/ames_houses.csv` in the course GitHub repository and can be downloaded using the link `https://raw.githubusercontent.com/richardfoltyn/MLFP-ECON5130/main/data/ames_houses.csv`.

To load the data, you need to specify the file path depending on your computing environment:

[1]:
```
# Use this path if the CSV file is in the same directory as the Jupyter notebook
file = 'ames_houses.csv'

# Use this path if you want to download the file directly from Github
# file = 'https://raw.githubusercontent.com/richardfoltyn/MLFP-ECON5130/main/exercises/
 ↪ames_houses.csv'
```

You can load the CSV file as a `pandas` `DataFrame` as follows:

[2]:
```
import pandas as pd

df = pd.read_csv(file, sep=',')

# Display columns in the data set
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1460 entries, 0 to 1459
Data columns (total 13 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   SalePrice         1460 non-null   float64
 1   LotArea           1460 non-null   float64
 2   Neighborhood      1460 non-null   object
 3   BuildingType      1386 non-null   object
 4   OverallQuality    1460 non-null   int64
 5   OverallCondition  1460 non-null   int64
 6   YearBuilt         1460 non-null   int64
 7   CentralAir        1460 non-null   object
 8   LivingArea        1460 non-null   float64
 9   Bathrooms         1460 non-null   int64
 10  Bedrooms          1460 non-null   int64
 11  Fireplaces        1460 non-null   int64
 12  HasGarage         1460 non-null   int64
dtypes: float64(3), int64(7), object(3)
memory usage: 148.4+ KB
```

The included variables are a simplified subset of the data available at openml.org:

- `SalePrice`: House price in US dollars (float)
- `LotArea`: Size of the lot in m² (float)
- `Neighborhood`: Name of the neighborhood (string)
- `BuildingType`: Type of building (categorical stored as string)
- `OverallQuality`: Rates the overall condition of the house from (1) "very poor" to (10) "excellent" (integer)
- `OverallCondition`: Rates the overall material and finish of the house from (1) "very poor" to (10) "excellent" (integer)
- `YearBuilt`: Original construction date (integer)
- `CentralAir`: Central air conditioning: Yes/No (categorical string)
- `LivingArea`: Above-ground living area in m² (float)
- `Bathrooms`: Number of bathrooms (integer)
- `Bedrooms`: Number of bedrooms (integer)
- `Fireplaces`: Number of fireplaces (integer)
- `HasGarage`: Indicator whether house has a garage (integer)

1 Data preprocessing

Apply the following steps to preprocess the data before estimation:

1. Drop all rows which contain any missing values (`NaN`)

   Hint: Use `dropna()` to remove rows with missing observations.

2. Trim outliers:

   1. Compute the 1st and 99th percentiles of the variables `SalePrice`, `LivingArea` and `LotArea`
   2. Drop all observations in which any of these variables is below its 1st percentile or above its 99th percentile.

   Hint: Use `quantile()` to compute the percentiles for the relevant variables. To convert percentiles to quantiles, you need to divide by 100.

3. Recode the string values in column `CentralAir` into numbers such that `'N'` is mapped to 0 and `'Y'` is mapped to 1. Store this numerical variable using the column name `HasCentralAir`.

4. Recode the values in column `Fireplaces` and create the new variable `HasFireplace` so that `HasFireplace = 1` whenever at least one fireplace is present and `HasFireplace = 0` otherwise.

5. Recode the string values in column `BuildingType` and create the new variable `IsSingleFamily` which takes on the value 1 whenever a house is a single-family home and 0 otherwise.
6. Plot the kernel densities (or histograms) for the variables `SalePrice`, `LivingArea` and `LotArea`. You will notice that all three variables have a right-skewed distribution.

   Hint: You can plot kernel densities using `DataFrame.plot.kde()` and histograms with `DataFrame.plot.hist()`
7. Convert the variables `SalePrice`, `LivingArea` and `LotArea` to (natural) logs and re-create the kernel density or histogram plots for the logged variables. Name the transformed columns `logSalePrice`, `logLivingArea` and `logLotArea`.

2 Estimation

2.1 Model specification

You are now asked to estimate the following model of house prices as a function of house characteristics:

$$\begin{aligned}\log(SalePrice_i) = \alpha + f\Big(&\log(LivingArea_i),\ \log(LotArea_i),\ OverallCondition_i,\\ &OverallQuality_i,\ Bathrooms_i,\ Bedrooms_i\Big)\\ &+ \gamma_0 YearBuilt_i + \gamma_1 HasCentralAir_i + \gamma_2 HasFireplace_i + \gamma_3 IsSingleFamily_i + \epsilon_i\end{aligned}$$

where i indexes observations and ϵ is an additive error term. The function $f(\bullet)$ is a *polynomial of degree 3* in its arguments, i.e., it includes all terms and interactions of the given variables where the exponents sum to 3 or less:

$$\begin{aligned}f(\log(LivingArea_i), \log(LotArea_i), \dots) = &\beta_0 \log(LivingArea_i) + \beta_1 \log(LivingArea_i)^2\\ &+ \beta_2 \log(LivingArea_i)^3 + \beta_3 \log(LotArea_i)\\ &+ \beta_4 \log(LotArea_i)^2 + \beta_5 \log(LotArea_i)^3\\ &+ \beta_6 \log(LivingArea_i) \log(LotArea_i)\\ &+ \beta_7 \log(LivingArea_i)^2 \log(LotArea_i)\\ &+ \beta_8 \log(LivingArea_i) \log(LotArea_i)^2\\ &+ \cdots\end{aligned}$$

Create a feature matrix `X` which contains all polynomial interactions as well as the remaining non-interacted variables.

Hints:

- Use the `PolynomialFeatures` transformation to create the polynomial terms and interactions from the columns `logLivingArea`, `logLotArea`, `OverallCondition`, `OverallQuality`, `Bathrooms` and `Bedrooms`.
- Make sure that the generated polynomial does *not* contain a constant ("bias"). You should include the intercept when estimating a model instead.
- You can use `np.hstack()` to concatenate two matrices (the polynomials and the remaining covariates) along the column dimension.
- The complete feature matrix `X` should contain a total of 87 columns.

2.2 Train-test sample split

Split the data into a training and a test subset such that the training sample contains 70% of observations.

Hint:

- Use the function `train_test_split()` to split the sample. Pass the argument `random_state=123` to get reproducible results.
- Make sure to define the training and test samples only *once* so that they are identical for all estimators used below.

2.3 Linear regression

Perform the following tasks:

1. Comment on whether you need to standardise features before estimating a linear regression model. Does the linear regression model have any hyperparameters?
2. Estimate the above model specification using a linear regression model on the training sub-set.
3. Compute and report the mean squared error (MSE) on the test sample.
4. Plot the prediction errors (on the y-axis) against the outcome variable on the test sample.

Hints:

- Use the `LinearRegression` class to estimate the model.
- The mean squared error can be computed with `mean_squared_error()`.

2.4 Ridge regression

Perform the following tasks:

1. Does Ridge regression require feature standardisation? If so, don't forget to apply it before fitting the model.
2. Use `RidgeCV` to determine the best regularisation strength α on the training sub-sample. You can use the MSE metric (the default) to find the optimal α. Report the optimal α and the corresponding MSE.
3. Plot the MSE (averaged over folds on the training sub-sample) against the regularisation strength α on the x-axis (use a log scale for the x-axis).
4. Compute and report the MSE on the test sample.
5. Plot the prediction errors (on the y-axis) against the outcome variable on the test sample.

Hints:

- Determine a suitable range for the grid of candidate α and space them uniformly in logs.
- Recall that the (negative!) best MSE is stored in the attribute `best_score_` after cross-validation is complete.

2.5 Lasso

Perform the following tasks:

1. Does Lasso require feature standardisation? If so, don't forget to apply it before fitting the model.
2. Use `LassoCV` to determine the best regularisation strength α on the training sub-sample using cross-validation with 5 folds. You can use the MSE metric (the default) to find the optimal α. Report the optimal α and the corresponding MSE.
3. Plot the MSE (averaged over folds on the training sub-sample) against the regularisation strength α on the x-axis (use a log scale for the x-axis).
4. Compute and report the MSE on the test sample for the model using the optimal α.

5. Report the number of non-zero coefficients for the model using the optimal α.
6. Plot the prediction errors (on the y-axis) against the outcome variable on the test sample.

Hints:

- Getting Lasso to converge may require some experimentation. The following settings should help: increase the max. number of iterations to `max_iter=100000` and use `selection='random'`. Set `random_state=123` to get reproducible results:

  ```
  LassoCV(..., max_iter=100000, selection='random', random_state=123)
  ```

- After cross-validation is complete, the MSE for each value of α and each fold are stored in the attribute `mse_path_` which is an array with shape `(N_ALPHA, N_FOLDS)`.

2.6 Elastic net

The elastic net is a linear model that applies both L1 and L2 regularisation, i.e., it's a generalisation of Ridge regression and Lasso. Its loss function is given by

$$L(\mu, \boldsymbol{\beta}) = \frac{1}{2N} \underbrace{\sum_{i=1}^{N} \left(y_i - \mu - \mathbf{x}_i' \boldsymbol{\beta} \right)^2}_{\text{Sum of squared errors}} + \underbrace{\alpha\rho \sum_{k=1}^{K} |\beta_k|}_{\text{L1 penalty}} + \underbrace{\alpha(1-\rho) \sum_{k=1}^{K} \beta_k^2}_{\text{L2 penalty}}$$

The additional parameter ρ is called the L1 ratio and determines the relative weight of the L1 vs L2 penalty terms (see also the `scikit-learn` user guide). Compared to Ridge regression and Lasso, this model therefore includes two hyperparameters, α and ρ, both of which should be determined using cross-validation. It is easy to see that for the corner case of $\rho = 1$, the elastic net corresponds to the Lasso, while for $\rho = 0$ is corresponds to Ridge regression.

Perform the following tasks:

1. Does the elastic net require feature standardisation?
2. Use `ElasticNetCV` to determine the best regularisation strength α and L1 ratio ρ on the training sub-sample using cross-validation with 5 folds. You can use the MSE metric (the default) to find the optimal hyperparameter values. Report the optimal α and ρ and the corresponding MSE.
3. Compute and report the MSE on the test sample for the model with optimal hyperparameters.
4. Report the number of non-zero coefficients for the model with optimal hyperparameters.
5. Plot the prediction errors (on the y-axis) against the outcome variable on the test sample.

Hints:

- Getting elastic net to converge may require some experimentation. The following settings should help: increase the max. number of iterations to `max_iter=100000` and use `selection='random'`. Set `random_state=123` to get reproducible results.

  ```
  ElasticNetCV(..., max_iter=100000, selection='random', random_state=123)
  ```

- The grid for α is determined in the same way as for `LassoCV`. For ρ, use the argument `l1_ratio` to pass a grid of candidate L1 rations given by `[0.1, 0.5, 0.7, 0.9, 0.95, 0.99]`:

  ```
  ElasticNetCV(..., l1_ratios=[0.1, 0.5, 0.7, 0.9, 0.95, 0.99], ...)
  ```

- Use `ElasticNet` to estimate the elastic net once you identified the optimal hyperparameters. Make sure to pass the same values for `max_iter`, `selection` and `random_state` as you did earlier.

2.7 Compare estimation results

Create a table which contains the MSE computed on the test sample for all four models (using their optimal hyperparameters). Which model yields the lowest MSE? Comment on why you think this is the case.

Arrangements for forming groups and allocating roles

Groups will be created by Programme Administrators, based on tutorial groups where possible.

Groups must discuss the **Business School's Assessed Groupwork Policy** and agree how they will apply it at their first meeting. The policy can be found on the Student Information Point Moodle. Note that the School's Coursework Group Policy is being revised as of August 2022 and may change slightly. We will update you when the policy is finalised.

Moodle forum

Group members must use the group forum provided to interact to ensure that no student is disadvantaged by not having a particular social media account.

Coursework Rubric

A holistic rubric provides a list of assessment criteria together with broad description of the characteristics that would be expected for each level of performance.

Criteria	Excellent	Very Good	Good	Satisfactory	Weak
Code functionality	The codes are written as functions which can take different parameter values as inputs.	The codes are written as functions but are not able to take all the parameter values as inputs	The codes are not written as functions but all the parameter values can be changed in one place	The codes are not written as functions and not all the parameter values can be changed in one place	The codes are not written as functions and the parameter values cannot be changed
Code readability	The codes are written as functions with clearly separated sections and informative comments	The codes are written as functions but without clearly separated sections. The comments are informative.	The codes are written as functions with clearly separated sections. The comments are not informative.	The codes are written as functions without clearly separated sections. The comments are not informative.	The codes are written as functions without clearly separated sections. No comments are provided.
Scalability	The codes are able to execute successfully for any parameter values in addition to	The codes are able to execute successfully for some parameter values in addition to	The codes are able to execute successfully for all the parameter values specified in the	The codes are able to execute successfully only for some of the parameter values	The codes are able to execute successfully only for a single combination of

	the ones specified in the assignment.	the ones specified in the assignment.	assignment.	specified in the assignment.	parameter values specified in the assignment.
Accuracy	The codes are able to execute accurately for any parameter values in addition to the ones specified in the assignment.	The codes are able to execute accurately for some parameter values in addition to the ones specified in the assignment.	The codes are able to execute accurately for all the parameter values specified in the assignment.	The codes are able to execute accurately only for some of the parameter values specified in the assignment.	The codes are able to execute accurately only for a single combination of parameter values specified in the assignment.
Speed	The codes are able to execute successfully and accurately very close to the best-case run-time.	The codes are able to execute successfully and accurately reasonably close to the best-case run-time.	The codes are able to execute successfully and accurately in somewhat higher than the best-case run-time.	The codes are able to execute successfully and accurately in very high time values compared to the best-case run-time.	The codes are not able to execute successfully and accurately in time.
Correctness	The codes address the question asked in the best possible manner with an excellent overall study of the problem asked.	The codes address the question asked in a very good manner with a very good overall study of the problem asked.	The codes address the question asked in a good manner with a good overall study of the problem asked.	The codes answer only a few aspects (50%) of the question asked.	The codes do not address the question asked appropriately.

Feedback method

[edit only if necessary]
Feedback on your assignment will normally be provided via Moodle. Generic (class-level) feedback and grade profiles will normally be posted on Moodle.
Students can use academic staff office hours for additional feedback on your work.

Preparing your coursework

Document creation

Coursework

1. Please use this file naming convention: **GroupNo_CourseCode, e.g. Group3_ECON5130**.
2. The file type must be .ipynb, .py and .pdf.
3. Include your group number in your document, ideally in the header on each page with the course code and title, e.g. Group3_ECON5130_Finance1.
4. The maximum file size limit on Moodle is 230MB

Formatting

You won't be penalised if you don't follow this good practice on formatting, but it will help your markers:

- Use a Sans Serif font in black, e.g. Arial, Avant Garde, Calibri, Helvetica and Geneva.
- Use font size 12.
- Use 1.5 or double line spacing.
- Align your text to the left margin.
- Add page numbers.

Referencing and bibliography

You should reference your sources appropriately and list these in a bibliography. The bibliography is excluded from your word limit. You should use the 'Harvard' referencing system, as detailed below for written coursework.
In the text, use the following referencing conventions:

- Smith (1999) argues that…. *or*
- It has been argued that……. (Smith, 1999).
- If you use a direct quote, use quotation marks and cite the page number as well as the author and date, i.e. (Smith, 1999, p. 4).
- If you have two items by the same author in the same year, refer to one as 'a' and the other as 'b', i.e. Smith (1999a) and Smith (1999b).

For more information, please refer to the University Library webpage.

Student conduct

Plagiarism

You must adhere to the University's rules regarding plagiarism which are based on the premise that 'all work submitted by students for assessment is accepted on the understanding that it is the student's own effort', in this case, the efforts of group members. More specifically, you must avoid plagiarism in the following forms:

- Copying from sources without 'formal and proper acknowledgement'
- Inappropriate collaboration – working with non-members to produce your group's coursework or copying work produced by another student/group

- Submitting work which you have obtained from another source, e.g. an essay mill
- Self-plagiarism – basing coursework on work that has already been submitted for assessment purposes.

For advice and more information, please consult:

- LEADS web pages
- University Plagiarism Statement

Turnitin

Note that your coursework will be processed through Turnitin for similarity checking. You can submit a draft of your coursework to Turnitin before submitting your final copy. You will find information about using Turnitin on the Student Information Point Moodle.

Submitting your coursework

A designated member of the group must submit in accordance with the stated time and date on page 1. See below for information if you are unable to do so.

Finalising your document

Please follow the steps listed below:

1. Check spelling and grammar using the inbuilt tool on your device. You will not be penalised for grammatical and spelling errors but we recommend that you take the opportunity to correct them.
2. Check your file name (see above).
3. Check that you have used an accepted file type (see above).
4. Do not include any names in the file name or the document to support anonymous marking.

Uploading your document to Moodle

1. One group member will upload your document to the designated section of the Moodle course, which will be clearly signposted.
2. Try to upload your document at least 30 minutes before the deadline (page 1) in case you encounter any technical issues. You will be able to resubmit the document as often as you like until the submission deadline.
3. Complete the Declaration of Originality (see below) on behalf of the group.

Declaration of Originality

When you upload your coursework on Moodle, you will be required to select a checkbox to confirm that you agree with the University's Declaration of Originality which applies to all academic work, as follows:

I confirm that this assignment is my own work and I have:

- Read and understood the guidance on plagiarism provided on the Student Information Point Moodle course including the University of Glasgow Statement on Plagiarism.
- Clearly referenced, in both the text and the bibliography or references, all sources used in the work.
- Fully referenced (including page numbers) and used inverted commas for all text quoted from books, journals, web etc.
- Provided the sources for all tables, figures, data etc. that are not my own work.
- Not made use of the work of any other student(s) past or present without acknowledgement. This includes any of my own work, that has previously, or concurrently, been submitted for assessment, either at this or any other institution, including school.

- Not sought or used the services of any professional agencies to produce this work.
- In addition, I understand that any false claim in respect of this work will result in disciplinary action in accordance with University regulations.

Extensions and non-submission with good cause

Please refer to the Student Information Point Moodle for relevant information.

There is no reassessment opportunity for this assignment.

Late submission penalties

In the absence of good cause, late submission penalties will be applied as explained in the Student Information Point Moodle.

Questions

If you have any questions about this coursework briefing, please read it carefully again to ensure you fully understand it. If you still have questions, please post these on the course Moodle Discussion Forum.

Personal questions only can be sent to [delete as necessary]:
business-accounting-finance@glasgow.ac.uk
business-economics@glasgow.ac.uk
business-management@glasgow.ac.uk