Linear Models Extended Project Feedback

Well done

- I was generally pretty impressed by the work that has been done
- Cleary a lot of time was put in and it appeared as though students learnt a lot
- The average grade was 85/100
- I will return individual feedback this afternoon

Main areas for Grading

1. Data Preparation
    a. Exploratory data analysis
    b. Missing data
    c. Feature Transformations
    d. Dummy Variables
    e. Standardisation
    f. Training/Testing matching

2. Modelling
    a. LinearRegression()
    b. LASSO()
    c. Hyperparameter selection

3. ‘Topics not seen in class’

Data Preparation: Exploratory data analysis

- Quick look at the data
- Do you have missing data?
- What types of columns do you have? Will you need dummy variables for categorical columns?
- Some simple plots
    - Marginal summaries
    - Correlations
    - Marginal distributions
    - Interactions with the response

Note: In machine learning we often have a large number of columns and therefore inspecting each one, and checking every model assumption is satisfied, is often not feasible.

Data Preparation: Missing data I

- The instructions said to remove columns with a high proportion of missing data, then remove rows
- Importantly you can't remove test set observations (not a problem here but would need to impute)

Data Preparation: Missing data II

- Some people played around with removing fewer/more columns and more/fewer observations.
- Others imputed data using summaries - means/median for continuous variables and mode for continuous variables - **Important: you should estimate the summaries on the training data to impute the testing data (just like standardisation)**
- Many noticed what we call ‘structural missingness‘ - ‘poolnum’ and ‘garagenum’ were either 1 or NaN so it was reasonable to impute NaNs as 0’s
- Lastly, people tried to use other variables to impute the neighbourhood code using some type of nearest neighbour algorithm - there were some lovely plots here

Data Preparation: Feature transformations

- Many noticed the y's were skewed, so took either log or sqrt transform (remember to transform back when making predictions)
- Often the exploratory data analysis uncovered non-linear relationships between features and response, so polynomial features were added - these not only raise the variables to a power they also consider interactions
- Power functions only relevant for continuous variables, but for interactions you may want to also consider the categorical variable (after transforming the dummies)
- Again rather than bespoke transforming each variable, this is slightly more about trying out as many things as possible in a Machine Learning perspective

Data Preparation: Dummy variables

- Most students correctly worked out which variables were ordinal (and could be left, i.e. ‘numrooms’, and which variables needed to be turned into dummies
- Working with the ‘regioncode’ was difficult as you had many variables
- The more variables you have the slower things will take to run - if you also do polynomial feature engineering you could have thousands of features
- Can remove dummies by simply grouping any category with below a certain threshold of observations into an ‘other’ category - therefore you preserve the non-ordinal structure, but don’t have too many categories
- Having an ‘other’ category can also be helpful when matching training and testing columns
- Others tried to group categories together - I would urge caution here. Grouping close categories together (i.e. ‘regioncode’) is assuming locally an ordinal structure. Better to group in terms of the response y.

Data Preparation: Standardisation

- Most people correctly used StandardScalar(), '.fit' to the training data and then '.transform' to the training and testing data.
- You use the training means and variances to 'scale' the test set
- Be careful using the scl() function

- Important for LASSO, not essential but useful for LinearRegression - **my advice is always scale your X's (for a linear regression it changes nothing in the modelling but makes computation and interpretation easier)**
- Scaling of the y's is slightly less standard - but can be helpful in setting the range for the beta's and lambda's (remember if you scale y's you need to undo the scaling to predict - using the training values).
- If you scale y to have mean 0 then you don't need an intercept beta0, if you don't then you either need fit_intercept=True or a column of 1's in the features (given by poly for example)

Data Preparation: Training/Testing matching

- Everybody was able to make sure their testing and training sets had the same columns.
- How I would think about this is as follows
  - Everything I do to my training set - I also do to my test set in the same way
  - Rather than independently processing both testing and training sets then matching
- Having an other category for dummies can really help here - leads to less information being lost

Modelling: LinearRegression() and LASSO()

- Generally this was understood well
- After creating dummies and polynomially transforming things there could be hundreds of variables
- Fitting a LinearRegression() did okay in-sample
- But evaluating out-of-sample - either in Kaggle or Cross-validating we saw the performance drop
- LASSO() helps to fit the model in a way that guarantees more ‘stable’ out-of-sample performance

Modelling: Hyperparameter selection I

- In class I showed you how to do this with GridSearchCV - applicable for any model.
- Specify a sensible grid for alpha = lambda - easier if everything is standardised
- This was time consuming if you had many columns
- LassoCV does the same thing but is faster - bespoke for LASSO, warm start optimisation

- Others uses LassoLarsCV - automatically chooses the grid over alpha - https://en.wikipedia.org/wiki/Least-angle_regression
- HalvingGridSearchCV - general and faster

Modelling: Hyperparameter selection II

- Some students considered AIC and/or BIC selection.
- These are ‘penalised likelihood’ methods that are alternatives to cross-validation
- Only fit the model once (for each hyperparameter) rather than for each cv-fold
- Looks at MSE but applied a bigger penalty for more parameters in the model

Topics not seen in class

- As discussed already - different ways to select alpha/lambda in LASSO
- I gave marks for any 'bespoke' imputing of missing data
- Some students tried dimension reduction techniques before doing the LASSO/GridSearchCV e.g SVD, variance thresholding
- Encoding - different types other than dummies.
- Outlier removal

Any Questions?