I HOPE IT DOESN’T RAIN TOMORROW

I HATE IT WHEN THE KIDS PLAY INSIDE

Elena Leonelli, on the ‘weather’ dataset

OUTLINE

Dataset visualization

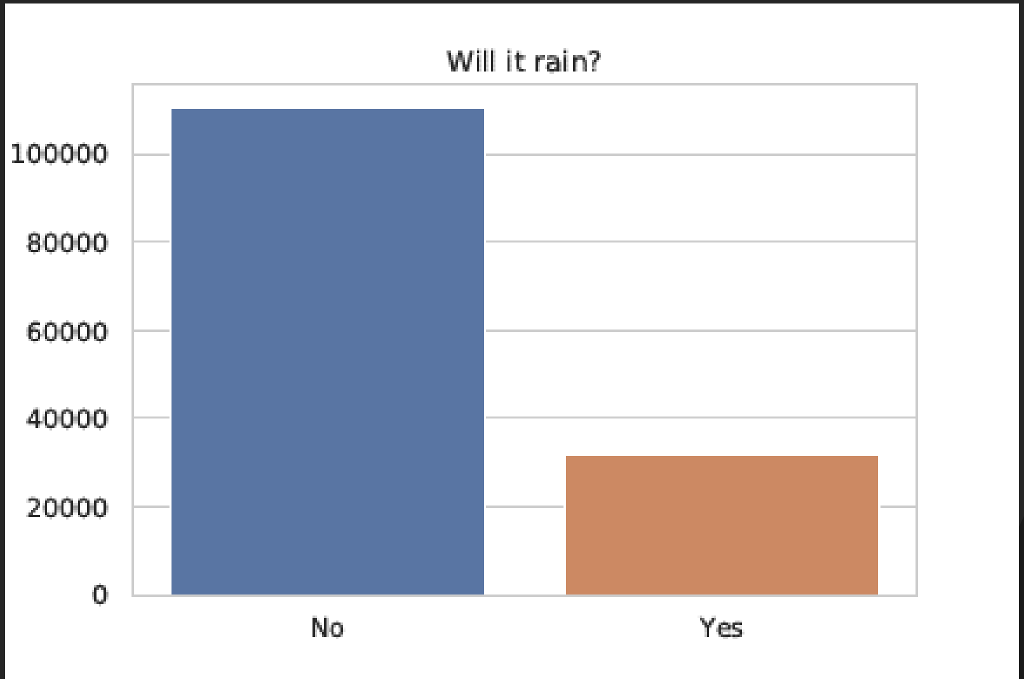
Preprocessing

Feature selection

Classification algorithms

Metrics

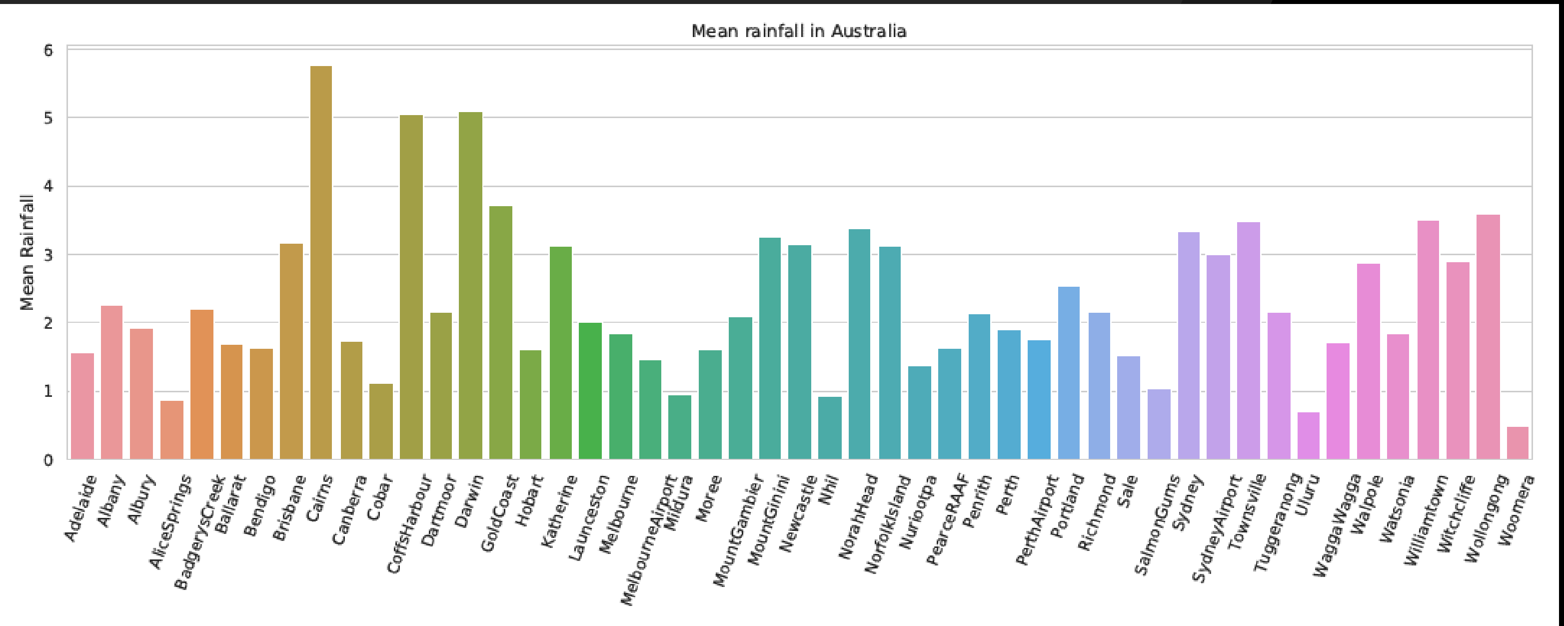
Conclusions



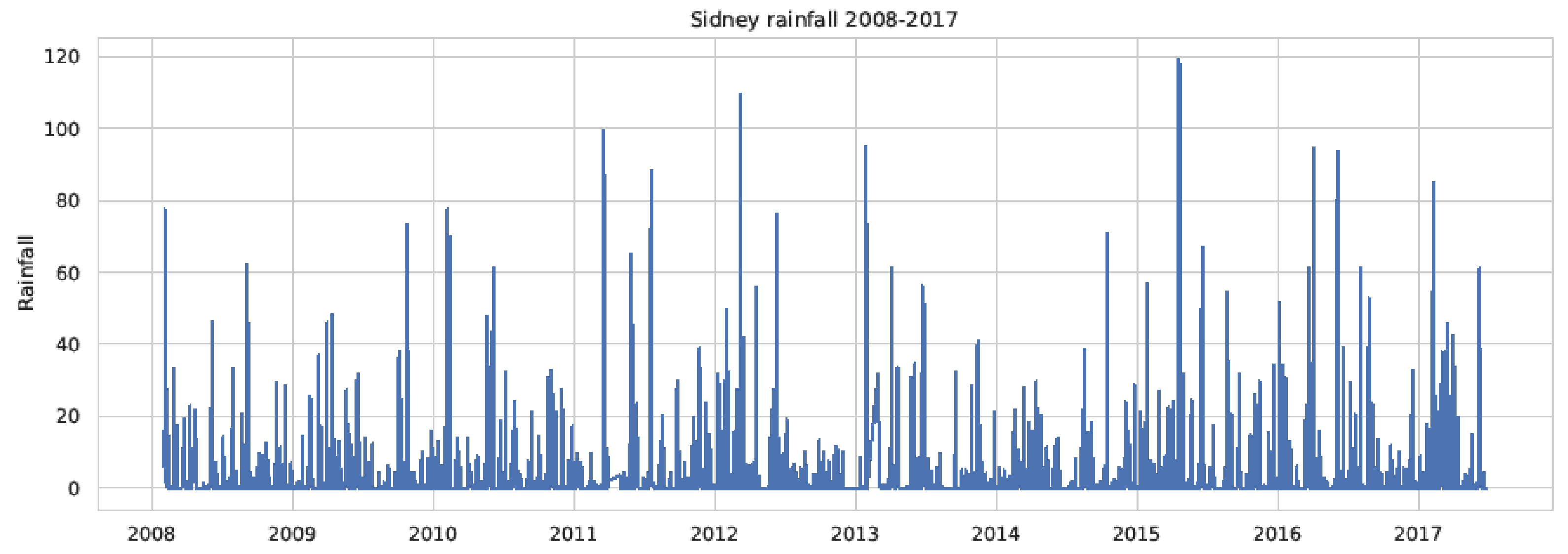
The dataset

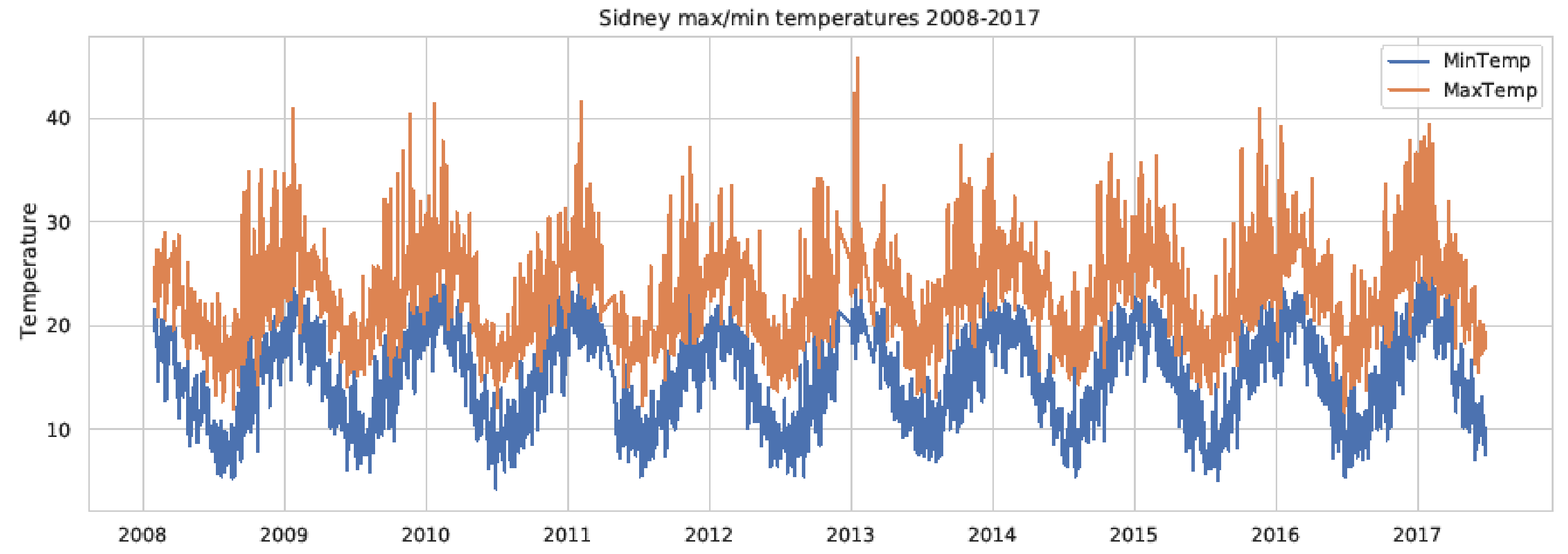
- Timeseries 2008-2017
- 49 different locations
- Numerical and non-numerical variables
- Imbalanced classes

$$\frac{no\ rain\ predicted}{rain\ predicted} \approx 3.5$$

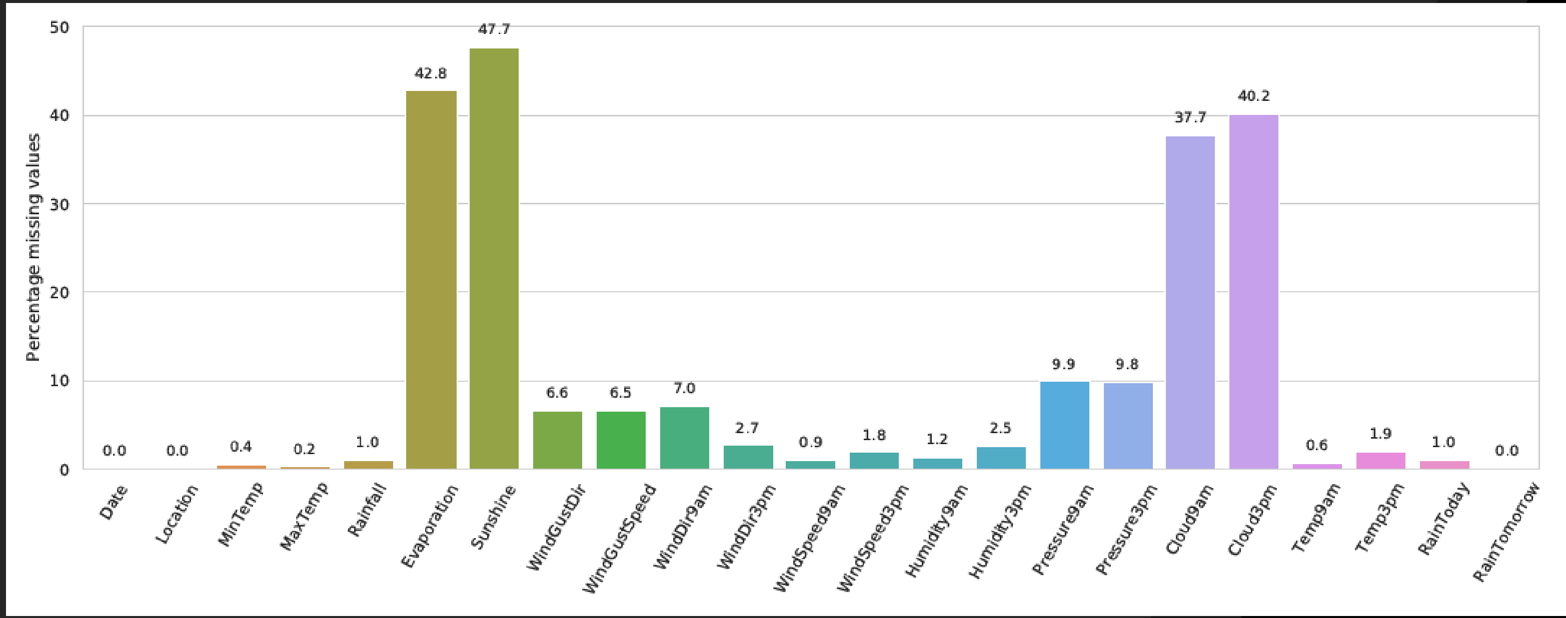


Rainfall and temperatures in Sydney





Preprocessing: missing values and dummies



Raw dataset shape:
(142193, 23)

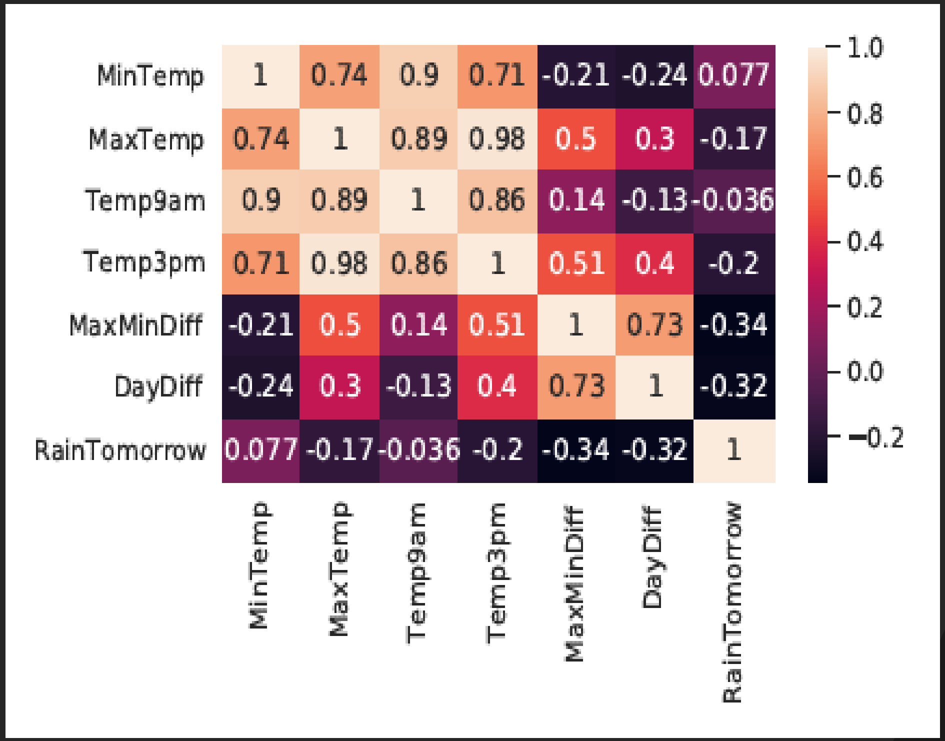
Preprocessed dataset shape:
(123710, 61)

87% of the initial rows

- Remove the features with more than 35% of missing values
- Replace with the mean the missing values for the numerical features
- Remove the remaining rows which still contain a missing value
- Create dummies for *WindGustDir*, *WindDir9am*, *WindDir3pm*

Feature selection

In this section I performed a feature selection based on two steps: firstly, I evaluate the (Pearson) correlation between numerical features in order to remove redundancies; after a normalization of the values in [0,1], I select the 10 features with the highest chi2 score (from the remaining numerical + dummies).



Example of correlation matrix for temperature. I created two extra features from the differences of temperature (max-min, 3pm-9am), which ends up to be more correlated (in absolute value) to the target than the others.

SELECTED

- Rainfall
- MaxMinDiff
- WindGustSpeed
- Humidity3pm
- RainToday
- WindDir9am_ESE
- WindDir9am_N
- WindDir9am_NNW
- WindDir9am_SE
- WindDir3pm_NW

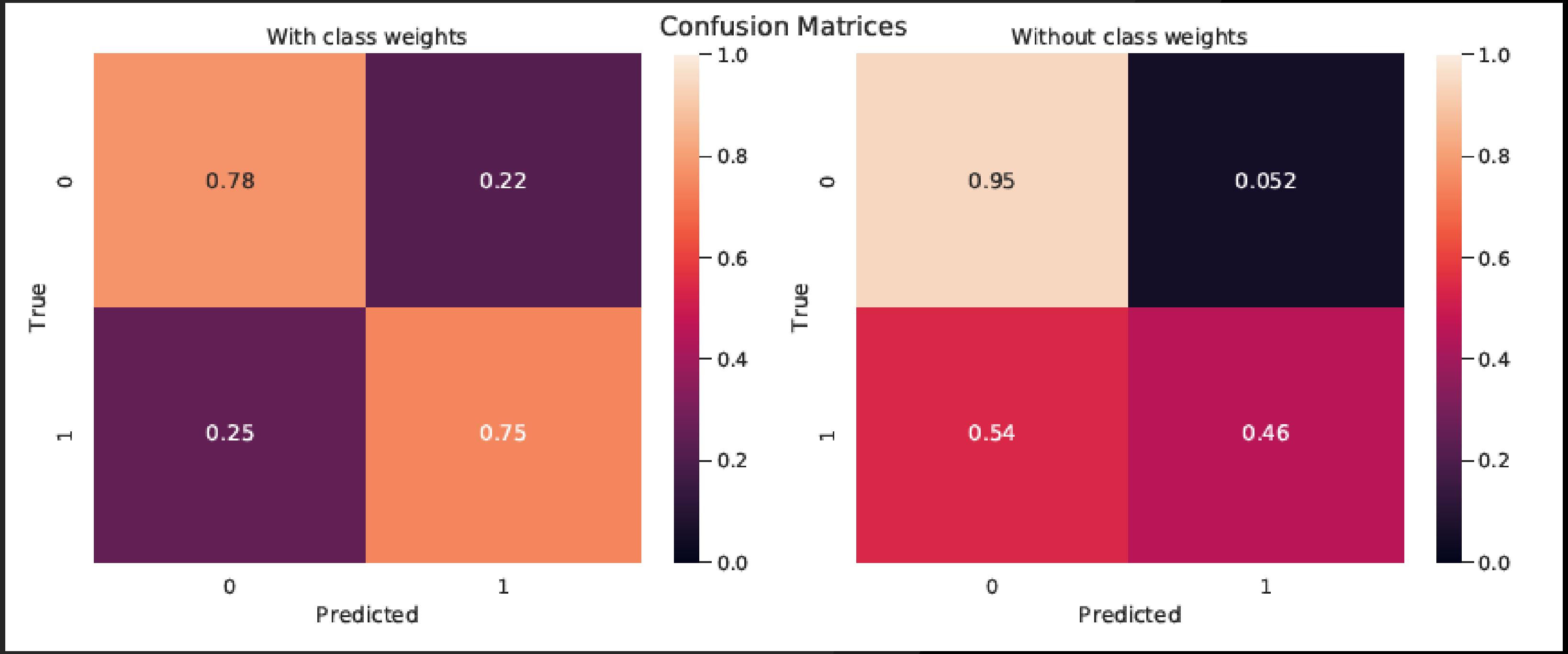
Classification algorithms

The imbalance between the two classes can cause a poor predictive performance for the 'Yes: it will rain' class (which is the minority class). To overcome this, I used two cost-sensitive algorithms by adding a 'weights' variable which contains the ratio between elements in the two classes. The main idea is that the model is penalized more in case of errors made on samples from the minority class, and less for errors made on samples from the majority class.

- Cost-sensitive Logistic Regression
- Cost-sensitive Artificial Neural Network

Metrics

Metrics plays an important role in evaluating the best model. I choose to report the confusion matrix between true and predicted labels: I want a model which is performing well in both True Positive and True Negative prediction, instead of having a high score only in True Negative predictions (*tomorrow will not rain*).



Confusion matrices for the Logistic Regression model: on the left, for the model trained with class weights; on the right, for the model trained without class weights.
The first model is performing well in detecting both True Positive and True Negative.

Conclusions

The peculiarities of this dataset that I faced were the percentage of missing values for some features, the presence of numerical and non-numerical variables, and the imbalance between classes.

The resulting best classification model is Cost-Sensitive Logistic Regression, which is able to reach over 75% of true predicted labels for both classes.