You may take a look at the website if you're interested in details about features and the background. For this problem, please directly use the npz file provided in the course material.

Three sets are included in the npz file: the training set, the validation set and the testing set, which contains 27435, 13514 and 10044 samples, respectively. The feature dimension of each sample is 53, and all features are numerical values. The target value to regress is a non-negative integer. In this problem, you don't need to round your float estimation to integers, and any evaluation such as MSE can directly be calculated using your float estimation and the ground truth.

You'll try the following regression models on the real-world data and conduct some analysis: linear regression, linear regression with l_1/l_2 regularization, CART, random forest and Adaboost.

For all model selection process, please use the training set to train models and use the validation set to select the final one. After selection, just test with the selected model and no need to train on (train+val). Don't standardize the dataset unless specified in questions.

Programming questions (for parts (b)-(f), see the "Hints on functions to use" box below):

(a) Come up with a baseline regressor which doesn't use much learning for later comparison. Explain your idea and why it would be a reasonable baseline. Implement your baseline regressor and report the testing MSE.

Note: your baseline should still use (or learn from) the training data but not much learning. This means that a baseline system that randomly generates an arbitrary number as output, without taking consideration of the dataset, would not qualify. The baseline cannot be one of the regression models mentioned in the problem description (such as linear regression), or a model of similar complexity.

(b) Try linear regression model with three regularization settings: no regularization, l_1 regularization and l_2 regularization. As for the regularization coefficient λ, search from $\log_2 \lambda = -10$ to $\log_2 \lambda = 10$ at step size of $\Delta \log_2 \lambda = 1$. Choose the parameter with the best val_MSE. Report the val_MSE and test_MSE of your best models.

Table you can use:

	Best λ	val_MSE	test_MSE
No regularization	-		
l_1 regularization			
l_2 regularization			

(c) Standardize all data in the following way: calculate the mean and std of all feature dimensions on the training set, and then subtract the mean and divide by the std to standardize (train, val, test). Repeat (b). Present the table and answer:

i. Do you observe any difference on the learned coefficients among three models? Explain.

ii. Compare test_MSE with those in (b). Which methods have obvious changes and which not? Explain.

Note: there might be some features which have the same value for all samples. Please handle their std values properly so there won't be divide-by-zero errors. Standardization only applies for this question, and don't do this in other questions.

(d) Try CART. Search from 1 to 10 on max_depth. As for other parameter setting, please use (criterion='mse', max_features=None, random_state=0) and default for others. Report the val_MSE and test_MSE of your best model.

(e) Try random forest. Set max_depth as the best one you find in (d). Search from 2 to 30 on n_estimators. Report the val_MSE and test_MSE of your best model.

(f) Try Adaboost. Set max_depth and n_estimators as your best values in previous questions. Search for a good value for learning_rate. Report the val_MSE and test_MSE of your best model.

(g) Do those regressors learn from the data compared with the baseline? Compare and comment on (explain) their performances relative to the baseline, and relative to each other.

Written questions (no computer simulations are necessary in (h), (i) below):

(h) Some of the features only have 0/1 values because they are converted from categorical features. Now for some of those categorical features, you modify the converted numerical values to 0/12345 (i.e., that attribute is either 0 or 12345) for all data including training, val and test and then do the regression problem again with the same random seed. Will your regressor give different estimations compared to your previous results? Give Yes/No answers for linear regression, CART, random forest and Adaboost, and explain why.

Note: Assume that the change of value doesn't affect any random process such as the feature sampling in random forest.

(i) What's the underlying assumption for the prior distribution of the model parameter in linear regression with l_1 regularization? Now given that the prior distribution is Exponential, derive the regularization term you should use. You may assume the components of $\underline{w}$ are independent for the purpose of the regularizer term.

Also answer: how could you ensure the w_j will all be positive in this problem?

Note: if a random variable x is Exponential(θ), then its probability density function is $p(x) = \frac{1}{\theta}\exp\left(-\frac{x}{\theta}\right), \theta > 0, x \geq 0.$

Hints on functions to use:

sklearn.linear_model. LinearRegression
sklearn.linear_model. Lasso
sklearn.linear_model. Ridge
sklearn.tree.DecisionTreeRegressor
sklearn.ensemble.RandomForestRegressor
sklearn.ensemble.AdaBoostRegressor