Machine Learning Engineer Nanodegree

Capstone Proposal

János Tamási

October 2, 2018

1. Proposal

Predicting Ames housing prices with the help of supervised learning algorithms and comparing the results of different approaches in data preparation, data selection and model architecture

https://www.kaggle.com/c/home-data-for-ml-course#description

1.1 Domain Background

Predicting housing prices based on the individual house's attributes is one of the classic examples of supervised learning tasks. It is a relevant task to solve because it has great practical and financial importance to be able to determine the real value of a house as precisly as possible based on its characteristics. The target variable is the price of the house in dollars, which we want to predict based on the predictors, which can be any characteristic of the property, both numerical and categorical. The task and the dataset is part of an ongoing Kaggle competition (link in the proposal section) which has already more than 700 competitor. My goal is to try out as many possibilities in the data preparation and the model building process as possible to finally reach a model with the best possible predictive capability to rank as high on the competition's leaderboard as possible.

In the history of supervised learning it all started with linear regression and logistic regression, then decision tree algorithms became widespread. A more efficient version of the decision tree algorithm is the random forest model which is an ensemble type of model which means it incorporates many different decision trees in one single model to find a more efficient one that performs better than all of the elements. Lately an even more efficient version of the random forest algorithm was developed which is called gradient boosted decision trees and it iteratively adds new decision trees to the ensemble model to minimize the prediction errors. There are many other algorithms to use in supervised learning like support vector machines, but I plan to investigate and compare to each other the decision tree related algorithms only. Finally here is a link to an academic paper where the authors applied machine learning algorithms for housing price prediction:
https://www.sciencedirect.com/science/article/pii/S0957417414007325?via%3Dihub

1.2 Problem Statement

The problem that I chose to solve is to determine the value of a house in Ames, Iowa in a certain historical time based on 79 explanatory variables of the houses and the data of the

previously sold houses and their characteristics in the area. The problem is not 100% objective because the value of a house can vary among individuals based on their subjective preferences, but at least can be approximated via actual transactions that took place in the past in the real estate market in the area. I plan to use supervised machine learning algorithms, namely decision tree related algorithms after proper data preparation. The predictive capability of the built model can be measured via mean absolue error values on a separated test data set and our predictions on those instances. To learn how good is our solution in the field we can compare our results with the results of the other competitors in the Kaggle competition.

1.3 Datasets and Inputs

The Ames Housing dataset was compiled by Dean De Cock for use in data science education. I acquired the dataset free of charge in the context of the kaggle competition that I specified previously. The dataset is divided into training and testing datasets. The training dataset contains 1460 datapoints with 79 predictor variables (from which 36 is numerical and 43 is categorical), an id column and a target variable (sale price). The testing dataset contains 1459 datapoints with the same predictor variables but without target variable data - this is used in the competition to check predictive performance of the competitors' models.

During my analysis I will examine which variables to use, how to translate categorical data to numerical ones and how to deal with missing data by imputation or omission of the incomplete datapoints.

1.4 Solution Statement

In my final solution I will use all of the numerical predictors and the best 20 one-hot encoded categorical variables (which I will determine by checking the effects of every categorical variable one by one by adding them to the numerical ones and comparing the results with and without them using cross validation - cross validation is necessary given the small size of the training data) in an XGBRegressor model. For this model I will find the optimal number of estimators and then I will fit it on the training data that is imputed with mean values at nans.

1.5 Benchmark Model

A benchmark model could be a really simple one, that many people use in everyday practice when considering to buy a home, namely that how big is the home in square feets - this is basically a linear regression problem with only one predictor. But to have benchmark model that is more difficult to supersede, I will use a linear regression model with the same predictors that I will use for the final xgboost model. A linear regression model can be implemented easily with the help of sklearn and I will compare my final results to this baseline model. The metric that I will use for comparison is the mean absolute error of the models.

1.6 Evaluation Metrics

The mean absolute error is calculated via taking the mean of the absolute values of the differences between the predictions and the actual home prices of the certain dwelling. By minimizing this value we can assure that our predictions are more and more precise and closer to the actual values of the buildings.

1.7 Project Design

After basic data preparation I will examine the predictive performance of the basic decision tree algorithm without parameter tuning, then try to find the optimal value of max leaf nodes and examine predictive performance as well, using only numerical data in both of the cases. Then I will find out how much better is a random forrest model without any parameter tuning. Then I will analize which categorical predictors to include in my model via iterative inclusion of one categorical variable at a time and examining how much the predictive performance improved with the inclusion. I will use random forest regressors during this comparative process. I will use the best 20 categorical variables from the total of 43. Then I will find the best number of estimators for an xgboost regressor by early stopping method to avoid under and overfitting, then use this regressor for my final predictions. During the competition I tested my results on the real test dataset by submitting my predictions, but for the sake of this project I will separate a part of the training data to test my models' performance in a reproducible way. Finally I will do some partial dependence plotting to examine the effects of certain predictor variables and create some pipelining tools to be able to use my final model architecture in an elegant and easy to use way.

An example of a partial dependence plot: how plot area affected home prices