Project Specification

Capstone Proposal

Domain Background:

This project is based on data collected from homes in suburbs of Boston ,Massachusetts . A model has been created with training and testing of data set to evaluate the performance and predictive of the model based on this data that is seen as a good fit then it could make some predictions about a home –in particular , its monetary value .This model will work like a brokerage agent of real estate , which can be used the information on the daily basis .

Date set & input:

The dataset for this project originates from the UCI Machine Learning Repository. The Boston housing data was collected in 1978 and each of the 506 entries represent aggregated data about 14 features for homes from various suburbs in Boston, Massachusetts. For the purposes of this project, the following preprocessing steps have been made to the dataset:

- 16 data points have an 'MEDV' value of 50.0. These data points likely contain **missing or censored values** and have been removed.
- 1 data point has an 'RM' value of 8.78. This data point can be considered an **outlier** and has been removed.
- The features 'RM', 'LSTAT', 'PTRATIO', and 'MEDV' are essential. The remaining **non-relevant features** have been excluded.
- The feature 'MEDV' has been **multiplicatively scaled** to account for 35 years of market inflation.

Solutions statements:

In this first section of this project, I will make a cursory investigation about the Boston housing data and provide my observations.

Since the main goal of this project is to construct a working model which has the capability of predicting the value of houses, we will need to separate the dataset into **features** and the **target variable**. The **features**, 'RM', 'LSTAT', and 'PTRATIO', give us quantitative information about each data point. The **target variable**, 'MEDV', will be the variable we seek to predict. These are stored in features and prices, respectively.

Benchmark model:

In this second section of the project, I will develop the tools and techniques necessary for a model to make a prediction. Being able to make accurate evaluations of each model's performance through the use of these tools and techniques helps to greatly reinforce the confidence in your predictions.

Implementation: Define a Performance Metric

It is difficult to measure the quality of a given model without quantifying its performance over training and testing. This is typically done using some type of performance metric, whether it is through calculating some type of error, the goodness of fit, or some other useful measurement. For this project, I will be calculating the coefficient of determination, R2, to quantify the model's performance. The coefficient of determination for a model is a useful statistic in regression analysis, as it often describes how "good" that model is at making predictions.

The values for R2 range from 0 to 1, which captures the percentage of squared correlation between the predicted and actual values of the **target variable**. A model with an R2 of 0 is no better than a model that always predicts the mean of the target variable, whereas a model with an R2 of 1 perfectly predicts the target variable. Any value between 0 and 1 indicates what percentage of the target variable, using this model, can be explained by the **features**. A model can be given a negative R2 as well, which indicates that the model is **arbitrarily worse** than one that always predicts the mean of the target variable.

Evaluation Matrices:

In this third section of the project, we'll take a look at several models' learning and testing performances on various subsets of training data. Additionally, we'll investigate one particular algorithm with an increasing 'max_depth' parameter on the full training set to observe how model complexity affects performance. Graphing the model's performance based on varying criteria can be beneficial in the analysis process, such as visualizing behavior that may not have been apparent from the results alone.

Cell produces four graphs for a decision tree model with different maximum depths. Each graph visualizes the learning curves of the model for both training and testing as the size of the training set is increased. Note that the shaded region of a learning curve denotes the uncertainty of that curve (measured as the standard deviation). The model is scored on both the training and testing sets using R^2, the coefficient of determination.

Project Design: Following is the workflow

1. **Getting started**	In this project, there will be evaluated the performance and predictive power of a model that has been trained and tested on data collected from homes in suburbs of Boston.
2. **Data Exploration**	In this first section of this project, I will make a cursory investigation about the Boston housing data and provide my observations.
3. **Feature Observation**	As a reminder, we are using three features from the Boston housing dataset: 'RM', 'LSTAT', and 'PTRATIO'. For each data point (neighborhood): • 'RM' is the average number of rooms among homes in the neighborhood. • 'LSTAT' is the percentage of homeowners in the neighborhood considered "lower class" (working poor). • 'PTRATIO' is the ratio of students to teachers in primary and secondary schools in the neighborhood.
4. **Developing a Model**	In this second section of the project, I will develop the tools and techniques necessary for a model to make a prediction. Being able to make accurate evaluations of each model's performance through the use of these tools and techniques helps to greatly reinforce the confidence in your predictions.
5. **Training and Testing**	What is the benefit to splitting a dataset into some ratio of training and testing subsets for a learning algorithm? It is useful to evaluate our model once it is trained. We want to know if it has learned properly from a training split of the data. There can be 3 different situations: 1) The model didn´t learn well on the data, and can't predict even the outcomes of the training set, this is called underfitting and it is caused because a high bias. 2) The model learn too well the training data, up to the point that it memorized it and is not able to generalize on new data, this is called overfitting, it is caused because high variance. 3) The model just had the right balance between bias and variance, it learned well and is able predict correctly the outcomes on new data.

6. Fitting the model	The final implementation requires that we bring everything together and train a model using the **decision tree algorithm**. To ensure that we are producing an optimized model, we will train the model using the grid search technique to optimize the 'max_depth' parameter for the decision tree. The 'max_depth' parameter can be thought of as how many questions the decision tree algorithm is allowed to ask about the data before making a prediction. Decision trees are part of a class of algorithms called *supervised learning algorithms*.
7. Predicting Selling Prices	With this model we can predict the price of homes owned by clients that they wish to sell
8. Applicability	we use these results to discuss whether the constructed model should or should not be used in a real-world setting.

Presentation: In this proposal there is data collected from homes in suburbs of Boston ,Massachusetts . Model has been developed to predict the price of the houses in that area , this model will work like a real state broker who can help to predict the price of the houses in that area, As the data has been trained and tested for the developing the model to find the result to make the required prediction.