Machine Learning Engineer Nanodegree

Capstone Proposal

Domain Background:

The task here is to build a model of housing prices in Ames, Iowa using the Ames Housing Dataset.

The author of this data, Dean De Cock grew unsatisfied with the Boston Housing Dataset which was used as an end semester regression project for his students. The Boston Housing dataset is from the 70's and the housing prices have become unrealistic for today's market. Even inflating the prices does not seem right as it'll make the data from "real to realistic", as said by him.

The `"SalePrice"` prediction can be used for multiple purposes by a real estate agent or even by a firm for investment analysis.
Without machine learning, the house prices are usually predicted by experts using complex rules. This is neither cost, nor time - effective. Furthermore, experts in the field cannot judge every aspect of the house. This is where Machine Learning comes in.

This is a *supervised multivariate regression* problem. *Supervised*, meaning we are given a set of *labeled* training examples. *Multivariate*, meaning we are given multiple features about the house. *Regression* refers to predicting a *target* numeric value, such as the price of the house, given a set of *features*, such as the age, area, etc. of the house.

Problem Statement:

Given are 79 explanatory describing the quality and quantity of many physical attributes of residential homes in Ames, Iowa. Our aim to train a supervised multivariate regression model to predict accurately, the sale prices of new houses not seen before by the model.
Concretely, given the features, we want to predict a selling price for the house, such that the deviation between the predicted price ($\hat{y}$) and the actual price(y) is minimal, i.e. $min(y - \hat{y})$

Datasets and Inputs:

The dataset was retrieved from the Ames City Assessor's Office and further edited and curated by Dead De Cock.
It is now available as a `.xls` file via a direct link (http://www.amstat.org/publications/jse/v19n3/decock/AmesHousing.xls) or through online platforms like Kaggle (https://www.kaggle.com/c/house-prices-advanced-regression-techniques/data).
The `train` file has 1460 observations and the `test` file has 1459 observations.

Datset: (http://www.amstat.org/publications/jse/v19n3/decock/AmesHousing.xls):
The original dataset contained 112 explanatory variables describing 3970 property sales that had occurred in Ames, Iowa between 2006 and 2010.
The final dataset has 79 explanatory variables describing 2930 residential property sales.
Any explanatory variables requiring special knowledge were deleted along with records of property sales that were not residential sales, such as sale records of stand-alone garages, condos, and storage areas.

Inputs: (https://ww2.amstat.org/publications/jse/v19n3/decock/DataDocumentation.txt)
The 79 explanatory variables include:

- 20 continuous variables relating to various area dimensions for each observation. Such as the total dwelling square footage, lot size, etc. Area measurements of living area, basement, porches are broken down into individual categories based on quality and type.
- 14 discrete variables which quantify the number of items occurring within the house. Such as the number of bathrooms(full and half), bedrooms, kitchens, cars that can fit in the garage, etc. Date of construction and remodeling are also recorded.
- 23 ordinal variables which rate various items within the property such as the quality and condition of the heating/garage/exteriors/fireplaces, etc.
- 22 nominal variables which describe the type of various physical aspects of the house along with the neighborhood, type of sale, and much more.
- There is a unique "Id" associated with each house.

The explanatory variables describe most, if not everything that a typical home buyer would want to know about a potential property. These 79 explanatory variables will be used to predict the price of a house.

Solution Statement:

This project will use a supervised multivariate regression model to predict the sale price of a house in Ames, Iowa, given 79 explanatory variables about the house.
The supervised machine learning algorithm can be:

- `LinearRegression`
- `SVMRegressor`
- `DecisionTreeRegressor`
- `AdaBoostRegressor`
- `RandomForestRegressor`
- `XGBRegressor`

The aim would be to minimise the difference between the predicted price and the actual price of the house.

Benchmark Model:

Dean De Cock splits this project into 3 parts for his students. The first model requires students to submit a simplistic model (`Model1`) and should have a minimum $R2\ Score$ of 0.73. He further provides a benchmark model to have an R2 score of 0.80 as 80% of the variation in the residential sale prices can be explained by simply using the total square footage (`TotalBsmtSF + GrLivArea`) and the `Neighbourhood` feature.
Kaggle (https://www.kaggle.com/c/house-prices-advanced-regression-techniques/leaderboard) provides a benchmark model too at 0.408090, which is evaluated as the $Root - Mean - Squared - Error$ between the logarithm of the predicted value and the logarithm of the observed sale price.
I will use the benchmark model provided by Kaggle.

Evaluation Metrics:

The evaluation metrics that *can* be used for this problem are:

- MAE: Mean Absolute Error or the Average Absolute Deviation
- MSE: Mean Squared Error
- $RMSE$: Root of Mean Squared Error
- $R2Score$

Logarithmic RMSE : The main metric used for this project (as given by kaggle (https://www.kaggle.com/c/house-prices-advanced-regression-techniques#evaluation)) will be $RMSE$(Root Mean Squared Error) between the logarithm of the predicted value and logarithm of the observed sale price.

$$\sqrt{\frac{1}{m}\sum_{i=1}^{m}(\ln y^i - \ln(h_\theta(x^i))^2}$$

Here:

- m is the number of instances in the dataset you are measureing the $RMSE$ on.
- i refers to the index into the instances.
- y^i is the target label (the desired output value for that instance).
- h is known as the prediction function, also known as the $hypothesis$.
- θ refer to the parameters/weights of the prediction function, which decide how much weight to give each feature.
- x^i is the vector of all the feature values of the i^{th} instance instance in the dataset.
- $h_\theta(x^i) = \hat{y}$ is the predicted value.

Some points to note:

- $RMSE$ is negatively oriented. This means that low RMSE is better.
- $RMSE$ is indifferent to the direction of the error. Penalises both types of errors equally.

Another metrics which is sometimes helpful is the $\mathbf{R2\ Score}$, also known as the *coefficient of determination*. It is measured as the proportion of the total variation in y(target label or price of the house, here) in the sample that can be attributed to the linear relationship with X (input variables or features).
In other words, the proportion of variation in y, that can be explained by X is known as r^2.

$$r^2 = \frac{Variance\ of\ \hat{y}}{Variance\ of\ y}$$

where y are the target labels, and $\hat{y}$ are the predicted labels.

Project Design:

Programming Language and Libraries:

- `Python 3.5`
- `Scikit-Learn`
- `Numpy`
- `Pandas`
- `Matplotlib`
- `Seaborn`

I have broken down the series of steps I will go through during this project.
Note: This is not the final theoretical workflow and will be altered as and when needed.

1. Get the data:

- Download the data, read the train and test files into a pandas data frame.
- Append the test set to the train set as they will need same transformations.
- Get information about the data, such as the number of instances and features, the datatype of each attribute and the number of non-null values.

2. Visualising and gaining insights:

- Creating a copy of the data for exploration purposes.
- Creating lists of continuous, discrete, ordinal column names by referencing Dean De Cock's paper.
- Make a list of numerical columns combining the continuous, discrete, and ordinal variables.
- Make a separate list of nominal variables, those which are not included in the numerical variables.
- Use the `describe()` function and `hist()` function to see the distributions of the features. Note down observations.
- Check if there are any tail-heavy distribution which needs log transformations.
- Check how many values are missing from each column.
- Create a separate series of the target label for convenience.
- Study correlation of features with the target label, as well as with each other. This will identify important features as well as repetitive features.

4. Preparing the data:

- Creating new features after gaining insights in the previous stage by trying out various attribute combinations.
- See if the new features have a better correlation with the target label than the previous variables combined to form the new feature.
- Fix or removal of outliers:
  - The 5 instances recommended by the author of the dataset, i.e houses that have more than 4000 square feet of `GrLivArea` or ground living area.
  - Univariate and Bivariate Analysis will be conducted to identify outliers and gain insights about the data.
- Create pipelines for all transformations needed to be applied to the dataset separately, for numerical and categorical variables.
- For numerical variables, this may include filling in missing values, log transformations, feature scaling, etc.
- For categorical variables, this will include creating dummy variables for each category.
- Combine both the pipelines to get a dataset ready for machine learning.
- Split the prepared dataset into train and test sets. The train set will be the first 1460 rows in the prepared data.

5. Exploring different models and picking the best 3:

- I will start of my trying various quick and dirty models (default hyperparameters):
  - `LinearRegression`
  - `SVMRegressor`
  - `DecisionTreeRegressor`
  - `AdaBoostRegressor`
  - `RandomForestRegressor`
  - `XGBRegressor`
- They can be compared on the basis of:
  - $RMSE$
  - Training Time
- For each simple dirty model created, I will create a `for loop`, which does an `N`- fold cross-validation and computes the mean and standard deviation of the evaluation metric on `N` folds of the training data.
- Understand which features are more important than others using `.feature_importances`.
- Do feature selection and create new features according to the importance of features. This may involve deciding the top n features to use. This will be done by creating an *elbow curve* representing how performance increases, with increasing number of features.

- With the reduced number of features, For each simple dirty model created, I will create a `for loop`, which does an `N`- fold cross-validation and computes the mean and standard deviation of the evaluation metric on `N` folds.
- Then I will pick the top 3 models which perform the best, within a short amount of time.

6. Fine-tuning the top 3 models:

- This step will involve using `GridSearchV` or `RandomSearchCV` to fine tune the models' hyperparameters.
- Transformations will also be treated as hyperparameters such as replacing the missing values with 0 or median values.

7. Predict on the test set and submit:

- The final step would be to predict on the `test` set using the best model I have.
- Next, I will submit it to Kaggle to get to know my score on the `test` set. I will post the same in my report.

References:

1. https://ww2.amstat.org/publications/jse/v19n3/decock.pz (https://ww2.amstat.org/publications/jse/v19n3/decock.pz)
2. https://ww2.amstat.org/publications/jse/v19n3/decock/DataDocumentation.txt (https://ww2.amstat.org/publications/jse/v19n3/decock/DataDocumentation.txt)
3. https://www.kaggle.com/c/house-prices-advanced-regression-techniques#description (https://www.kaggle.com/c/house-prices-advanced-regression-techniques#description)