Project proposal

1. **Motivation: What problem are you tackling? What interests you?**

This project aims to predict prices of different types of properties. I would be doing so by drawing analysis on different attributes of the dataset and finding correlation between those and the target variable, price.

My key area of interest is to determine what factors essentially influence prices of a housing property and drawing a model that can automate price prediction by inputting values of those variables. I am also interested in finding out whether the attributes have any meaning/interpretation on their own in this context.

2. **Dataset: Present pointers to the relevant dataset.**

Original dataset (CSV) contains 1459 rows * 81 columns.

The dataset (from Kaggle's advanced regression techniques) contains a variety of factors that play an influential role for a group of people buying a house. I am interested to find out which of these would be actually benefiting to drive prices and which are redundant. There are a few important features that are fairly obvious, such as: Lot area, Neighbourhood, Air conditioning, Year built in, Number of bedrooms etc.

3. Method: What machine learning techniques are you planning to apply or improve upon?

I am planning to apply regression model, classification decision tree, bootstrap and random forest on this dataset in order to help us predict our target of prices for properties. I will also explore additional techniques beyond what was covered in the course, depending on how the aforementioned models perform (i.e. ensembling, boosting) to find the highest performing model (in terms of accuracy on validation set). I am also planning to create visualisations to gain clarity over the attributes.

4. Intended experiments: What experiments are you planning to run? How do you plan to evaluate your machine learning algorithm?

I am going to evaluate the machine learning algorithm by looking at the accuracy of the model compared to the test dataset (which will be partitioned), and looking at the Root Mean Square Error (RMSE), etc.

Overview: We aim to predict prices of different types of properties. We would be doing so by drawing analysis on different attributes of the dataset and finding correlation between those and the target variable. Our key area of interest is to determine what factors significantly influence the price of a commercial property and creating models that can automate price prediction by inputting values for those attributes; while finding out what model yields the highest predictive accuracy. We are also interested in finding out whether the attributes have any meaning/interpretation on their own in this context.

Concepts covered:

- Decision Tree Regression, Random Forest Regression, k-NN classification, KMeans classification, Exploratory analysis, Outlier analysis, Visualisations, Boosting Models (Adaboost, Gradient Boost, XGBoost, Light Boost, Lasso, Elastic Net, Kernel Ridge), Data Cleaning (One-Hot Encoding, skew normalization, Labelencoder), Multiple Linear Regression Model, Heat Maps

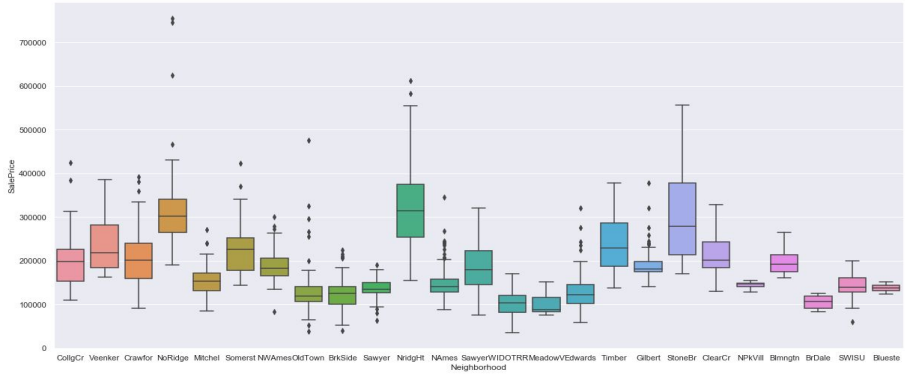
Data cleaning:
We start off with 81 columns in the initial dataset that was procured from Kaggle competition, House Prices: Advanced Regression. The first round of data cleaning was performed logically, eliminating those which were:

a) Directly related to price or overly repetitive
b) Disparate in description but are more meaningful when combined (i.e. 'bathroom count' was in separate rows in the original dataset, such as number of bathrooms in the basement, in the 1st floor, 2nd floor, etc. These were manually combined into 2 rows, one for 'total full bathrooms' in the house (comprising of all toilet amenities) and another for 'total half bath' (comprising of limited amenities of a bathroom) for a more appropriate grain of detail.)

Following this, we restructure the dataset by filling in NAs (i.e. lot frontage with mean values, categorical columns with 'None'). Then, we convert misinterpreted features. For example, the column MSSubClass is numeric (20-190) in the dataset, but this does not make sense, because the numbers are codes identifying different types of buildings, and style '20' is not 6 times "worse" than style '120'. These codes need to be converted into categorical (strings) to avoid confusion, for further processing later. The same logic applies to 'YrSold' because a house sold in 2006 does not mean it is 2000 times better than a house from 2000 years ago (not properly scaled)..

Now, for categorical attributes which do follow a sequential order (which are relative and can be scaled, like "heating quality"), it is appropriate to just replace them as numeric (with poor the lowest, excellent the highest) using Labelencoder to transform them from 1 to N. After this process, we are left with 8 attributes (categorical columns) which are not relative, such as Neighborhoods and SaleCondition (normal, bankrupt, family sale). We apply one-hot encoding for these 8 columns by creating additional columns to encode the information correctly. At the end of our cleaning, we are left with 35 predictors, and the target, 'SalePrice'.

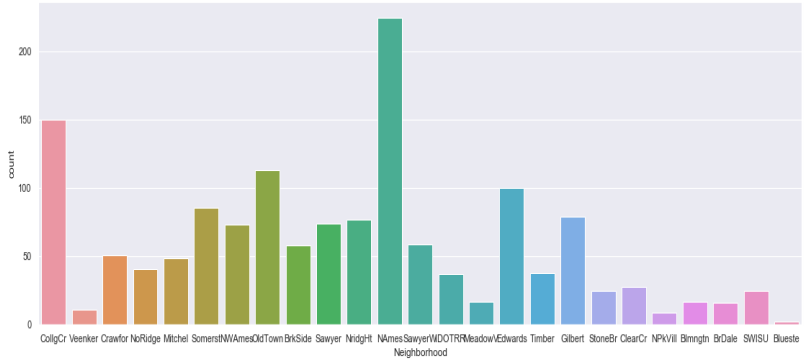
Exploratory analysis:
We move forward by identifying a few KPIs and drawing visualisations on those to gain perspective on their relevance to our goal. We are able to draw insights from the following:

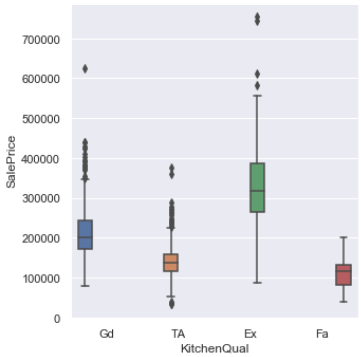


1. *Neighbourhood versus price*:

As illustrated above, we can see that Nridge Ht is the most expensive neighbourhood, followed by NoRidge and Stone Br. There are other neighbourhoods which have a lower average of housing prices such as Mitchel, NWAmes, BrkSide, Sawyer, IDOTRR, MeadowV, NPkVill, and Blueste. As an aside, this dataset shows shows how correlated location is with sales price! The value of these properties in Ames, Iowa is much more affordable than Seattle or SF!

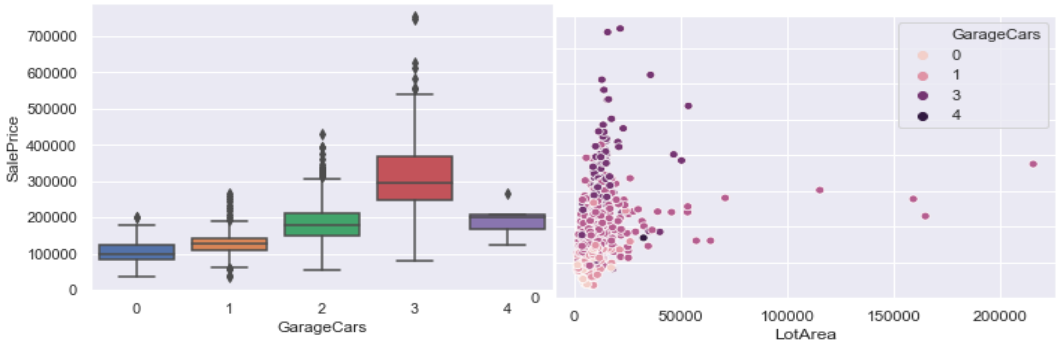
The below figure illustrates the spread of properties across different neighbourhoods. Clearly, NAmes, CollgCr, OldTown and Edwards are quite densely populated areas. These also happen to be amongst the low price range areas. It is possible that these are quite popular and relatively old habitants and therefore the overall quality and prices are quite low. (NAmes :5.36 CollgCr : 6.64 OldTown : 5.38 Edwards : 5.08)





2. *Kitchen quality versus price*:
There aren't many houses with "Excellent" kitchen conditions and those which are, typically have higher prices. Kitchens of "Fair" quality, have the lowest prices. The unique separation between prices of Good and TA quality, could help us in predicting price later (kitchen quality could be a good predictor).

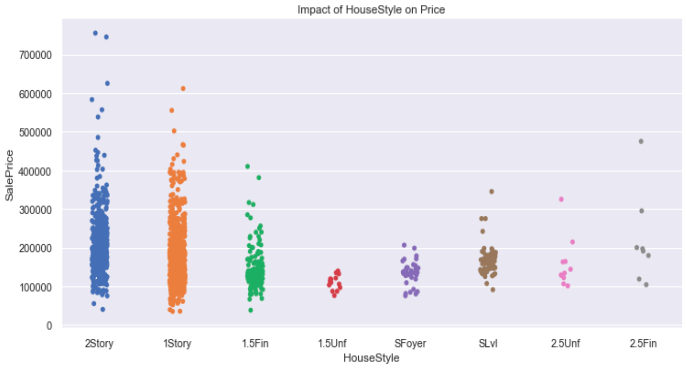
3.*Garage Cars*
Based on the scatter plot, most house has lot area smaller than 50k square feet, and "LotArea" is not an important factor in determine the price. However, the size of the garage is positive correlate to the sale price. House worth more than 500K typically can park 3 cars. According to the box plot, we can easily conclude that most of the houses with a 3-car garage have a higher price compare to any other houses with different size of the garage. The house with 4 cars are below $300,000 with the highest sales at $265,979 which is lower than the 75% of the house with a capacity of parking 3 cars. Besides that, the big trend is just as we predicted that with the bigger size of the garage, the higher the price.



4. *LotArea*: We analyse the contributors of influence against Sale Price, of which an interesting graph came through from Sale Price versus Lot Area. We see a few significant outliers and decide to probe them further. Our analysis shows that although these are superficially abnormal, their Overall Quality score indicates otherwise. The properties that are anomalous are actually bigger in size, but having a poor quality dips their sale value (since quality is a significant influencer to price). Thus, these aren't actually outliers, but an important set of records which will shape our models more holistically and are thus not removed.

5.*House style versus price:*

We analyze the relationship between the house style and the sale price. Based on the plot, “2story” and “1story” are the most common house style. Other type of house style (exclude 1.5 Unf), has a sale price range from 50K to 300K. The “1.5Unf” listed around 100K. The least common style on on file is the “2.5Unf” and “2.5Fin” which are two and one-half story with or without second floor furnished.



6. *Correlation study*:

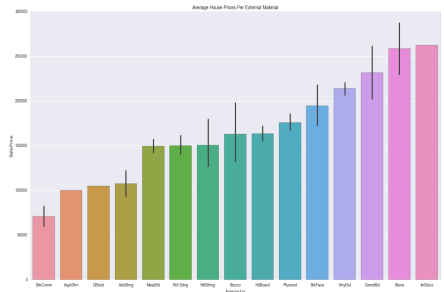
We study the attributes of the dataset iteratively to select variables that have the most significant contribution to the sale price. The matrix below is an illustration of the same and we have used these to build our model upon.

7. *House’s price vs type of exterior material, basement quality, and year sold*:

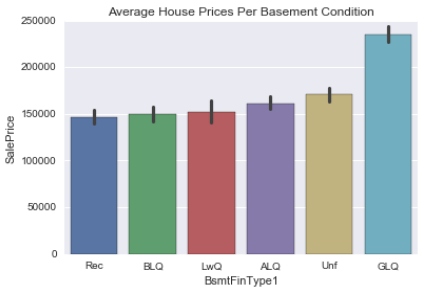
Starting with material, barplot 1 shows that the Brick Common material has the lowest average housing price. Imitation Stucco actually has the highest average price, followed by Stone, and then Cement Board. These are intuitively more 'premium' materials seen on luxury homes, and demonstrate good correlation with a house’s value. Unfortunately, the same cannot be said about basement condition and housing price (barplot 2), as unless the basement is at the level of GLQ - 'Good Living Quarters' (highest standard) - it does not have a high correlation with price; there is not much delineation between all of the other basement condition classes.

For this particular dataset we cannot see a high correlation between year sold, and the house price (barplot 3 - averages are close across 2006-2010). Part of the reason is because this dataset does not provide a larger range of records (spanning more years). Historically, 2006-2009 represents the housing bubble recession within the United States.

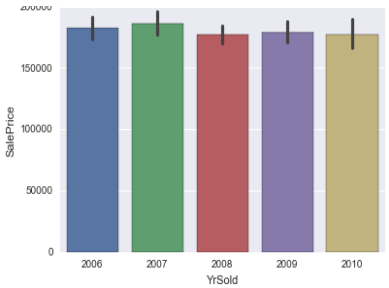
Barplot 1: External Material, Avg Housing Price



Barplot 2: Basement Condition



Barplot 3: Year Sold, Avg Housing Price



ML Modelling:
For all models: we partition the cleaned dataset into 80% training set, 20% test set. StandardScaler was used to normalize the data for all 35 predictor columns. To normalize skew for the target variable (saleprice), we should apply log (log1p).

1. Price Prediction:

Multiple Linear Regression
By using the sklearn package, we ran regression model after converting 8 columns separately into numbers adopting one-hot encoding to distinguish different values of the attributes. Using R square value to exam the fitness of our linear regression model. By running the test set data on our model, the R^2 is around 0.85, which means the model can explain 85% of the variability of the test dataset around its mean. The RMSE and MAE looks reasonable, thus this model can be used to predict the house price to a very good extent.

Decision Tree Regressor & Random Forest Regressor:
We used Decision Tree Regression to predict prices with our aforementioned retained attributes. Using this, we got a best score value of 0.75, with max depth as 10 and minimum number of splits for the tree as 0.05. Next, we considered using a Random Forest Regressor. Here, we found best score value to have improved, only slightly, as 0.77, with max depth at 4 and same number of minimum splits. The RMSE value obtained here was 38653.8, which was lesser than the earlier model. Interpreting this model's performance, it can be inferred that they can probably be more beneficial, but a better result can surprisingly be obtained from a Multiple Linear Regression model for this particular dataset.

Boosting:
Exploring boosting models, we will cover 7 techniques: Adaboost, Gradient Boost, XGBoost, Light Boost, Lasso, Elastic Net, and Kernel Ridge. Starting with Adaboost, as we learned in class, this is a type of "Ensemble Learning" where multiple learners are employed to build a stronger learning algorithm, by iteratively improving and accounting for incorrectly classified examples in a training set. Applying Adaboost regressor as base learners using randomized search with cross-validation, our best score was 0.831, with the best hyper parameter setting {'n_estimators': 80, 'learning_rate': 1}. The second boosting method we covered in lecture was Gradient Boosting. It also uses ensemble learning, but generates new learners during the training process to predict/calculate loss, compared to Adaboost requiring users specify a set of weak learners and adjusting their weights during the training process. With gradient boosting regressor, we yield a Root Mean Square Error (on test set) of 0.133. Performing cross-validation with randomized search CV, the best score is higher than Adaboost at 0.885. The best hyper parameter setting is {'n_estimators': 60, 'learning_rate': 0.1}

For models going beyond the scope of lecture content, let's start with XGBoost. XGBoost is short for eXtreme gradient boosting. The model uses Gradient Boosting, in addition to Stochastic Gradient Boosting with sub-sampling at the row, column and column per split level, and Regularized Gradient Boosting with L1 (LASSO) and L2 (Ridge) regularization. Calling XGBRegressor to fit the model, we yield a variance score of 0.8841 ('explained variance regression score' function is a metric for accuracy, like R^2, used because 'accuracy score' cannot be performed for continuous value predictions). The RMSE on Test data is 0.1317. These results are comparable to our (fine-tuned) regular, gradient boost model.

LightGBM is a gradient boosting framework that uses tree based learning; it splits the tree, leaf-wise, with the best-fit, whereas most boosting algorithms split the tree depth-wise. It produces much more complex trees and achieves higher accuracy, but can sometimes lead to overfitting. Scores are comparable to XGBoost, but the training is often (surprisingly) faster. Calling LGBMRegressor, we yield a variance score of 0.901, with RMSE of 0.127.

Lasso Regression is a type of linear regression that uses shrinkage. Shrinkage is where data values are shrunk towards a central point, and encourages simple models, well-suited for models showing high levels of muticollinearity (like this dataset we are using for housing attributes). Our model variance score is 0.8857, with a RMSE of 0.1293.

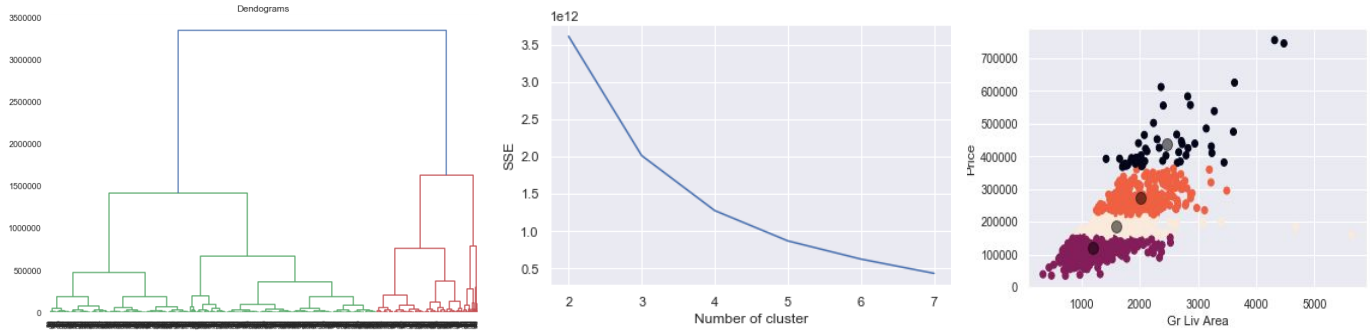
Finally, Elastic Net Regularization and Kernel Ridge touch on the features aforementioned from XGBoost, that is, they solve the regularization problem specifically (preventing overfitting by adding noise). Elastic net regularization method includes both LASSO (L1) and Ridge (L2) regularization methods. It's no surprise then, that the variance is similar to what we've seen already, 0.8856, but with a RMSE of 0.1292 (slightly lower). And Kernel ridge focuses on L2, plus a 'kernel

trick', defined as "mapping to get linear learning algorithms to learn a more complicated, nonlinear decision boundary." For this, our model variance score is 0.8889, with an RMSE on test set of 0.1295.

2. Building classification (secondary exploration):
Playing around with data further, we can use classification technique to use attributes such as Land area, price, number of bedrooms, basement area, number of floors etc to determine a building type. This predictive model can be used in a setting where a potential buyer can share his inputs (the attributes mentioned above) and can see what building type would cater to his needs in this city. Using k-NN classifier for this task, the model can achieve 87% weighted accuracy on test set.

3. Resident economic background profiling (secondary exploration):
Upon looking at various scatter plots, we searched for other potential questions that can be solved using unsupervised learning. One such question was profiling economic backgrounds of residents based on their house area and house price. We follow Agglomerative and KMeans clustering for this task. 4 clusters was determined to be the optimum elbow point from both methods, signifying that there are likely 4 kinds of economic backgrounds (of the homeowners) in the dataset.

Cluster 1: Living area ranges from 100~2500 units while the prices are close to about $180000. These would probably be single occupants, students living on a budget and low income brackets.
Cluster 2: Living area ranges from 1000~3000 units while the prices are close to about $180000~$210000. These would probably be small size families.
Cluster 3: Living area ranges from 1200~3500 units while the prices are close to about $220000~$380000. These would probably be medium size families who are moderately upper class in income holdings.
Cluster 4: Living area ranges from 1200~4200 units while the prices are close to about $380000~$620000. These would probably be small to big size families, but typically elite crowd of the city.



Performance measurement: "Silhouette Coefficient" measures mean distance between:
1. Sample and all other points in the same class, and
2. Sample and all other points in the next nearest cluster. A higher score is reflects a good good model performance. In our case, we got 0.53 for Agglomerative and 0.54 for KMeans clustering models. Both seem to be equal in performance and not entirely poor.

Analysis:
Pricing prediction objective wise, we were able to derive high accuracy from the Multiple Linear Regression model (at 84.6%). Its contemporary regression models like Decision Tree and Random Forest can be concluded as not as expectably ideal for this particular dataset (surprisingly). From the 7 further boosting models, we can see that LightGBM performed the best for our dataset (with 90% accuracy-variance score), followed by Kernel Ridge (88.9%). Adaboost - even with applied random and grid search - had the lowest score of 83%. The other models, despite introducing sound, advanced regularization techniques, don't have a dramatic improvement in score compared to a well-tuned, regular gradient boost model, which sits at 88%. However, the other models (Elastic Net, Lasso, XGB) do have slightly lower error scores. From this exercise, we can see that indeed, boosting can be a great way of creating more accurate, better performing models!

For secondary objectives, on the classification front, we explored how the attributes can be used to predict the type of building/house (discreet classes). Perhaps a deeper real estate perspective could give us more KPIs to research on and build/fine tune additional models in future explorations of classification. For clustering, we applied unsupervised learning to gain insight on the types of residents likely living in different clusters. We applied profiling analysis to deduce the types of demographics/economic backgrounds from the clusters we observed. In the future, we could develop more complex models to allow for more attributes to be introduced. This can improve our clustering model's performance and be more helpful in a realistic demographic profiling.

Conclusion:
This project was an insightful, hands-on opportunity to tackle a real-world data science problem, going through the process of handling (tricky) variables to process/clean, and then comparing + boosting the performance of various ML models. We investigated a variety of factors that play an influential role for a group of people buying a house through visualizations, then successfully applied course concepts like Random Forest, Linear Regression, Classification, Clustering, and Boosting Models (Adaboost, Gradient Boost). In addition to attempting new techniques beyond the scope of what we've learned in lecture (XGBoost, Light Boost, Lasso, Elastic Net, Kernel Ridge). This ultimately provided us with a house price prediction model with great performance (LightGBM, model can explain 90%-92% of the variability of the test dataset around its mean), and left us with a deeper understanding of how to tackle a complicated DS problem. Yay!