11/17/2020 6:14 AM

11/10/2020 11:56 AM

11/2/2020 11:36 AM

Please use this template. Retain the gray text. Your new materials—in black 12-point Times New Roman—should not exceed 5 additional pages excluding references and figures. Note the evaluation criteria, and leave plenty of time for editing. Your term project should have functionality and it should be an opportunity to learn. Please keep the examples and hints in mind from Assignment 1.

This assignment should respond to all of your facilitator’s suggestions in the previous assignment because we want to see that you are learning.

11/10/2020 1:36 PM

Bid Estimator (BE)

by Saurav Banerjee

# ASSIGNMENT 1: PROJECT PROPOSAL DRAFT

Please use this template in creating your response. Retain the gray text and MS Word headings, and supply your responses where indicated. Your materials, in black 12-point Times New Roman, should not exceed 5 pages excluding this gray text, the references and your figures. You may add appendices, which should be referred to in the body of the paper, and which will be read on an as-needed basis. Note the evaluation criteria below, and leave plenty of time for editing so that your paper responds to them and you obtain the most favorable grade. You may alter your project plans as the term progresses: that is to be expected. Your changes will fit with this growing document. Keep notes of what you read so that you can provide references. Responses considered “good” should go beyond the minimum of what’s requested.

## 1.1 SUMMARY DESCRIPTION

One- or (if necessary) two-paragraph overall purpose of a proposed term project. You will be free to change this in future but we want you to think through your most promising project idea as early as feasible because implementation can be time-consuming. The *clarity* of your summary is especially relevant, so avoid details in this summary. An example: *RealEval* will estimate the value of a house in the ABC neighborhood of Tucson based on recent sales of comparable properties.

Buying a house is one of the most expensive investments that an individual makes in a lifetime. During the buying process, the buyer makes an offer based on the quoted Sales Price. The seller, on the other hand, gets to decide on the final price at which they would sell the property, out of multiple offers from interested buyers.

As a buyer, it is often challenging, to come up with a reasonable offer which will be accepted by the seller. Bid Estimator (BE) helps the buyer to estimate the offer amount, based on the recent sales of comparable properties in Ames, Iowa.

## 1.2 I/O EXAMPLES

At least two concrete examples of projected output for designated input. You will not be held to this output exactly—it is just illustrative at this stage. The *specificity* of your examples is especially relevant: avoid generality. An example: Input 123 Main Street, 4 bedrooms, 2 bathrooms, …. Output $650,000 with confidence 80%.

Below are sample input/output examples generated by BE –

1. Sample Input/Output 1 –

Building Type – Single Family, Year Built – 1990, Sq. Footage – 1200 sq. ft, Bedrooms – 4, Bathrooms – 3

Output A - $ 700,000 with confidence 80%

1. Sample Input 2 –

Building Type – Town House, Year Built – 1950, Sq. Footage – 1200 sq. ft, Bedrooms – 3, Bathrooms – 2

Output - $ 659,000 with confidence 82%

## 1.3 REQUIREMENTS

Provide high-level requirements. Separate your requirements into three approximately even categories using triage (place requirements the two extreme categories first—*definite* and *nice-to-do*—and then place the remaining requirements in the middle category). Be conservative with your “definite” requirements: make them the bare necessity to have an actual project but no more. State requirements in declarative language such as “NUMRECO shall recognize numbers 0-9 from a 12 by 35 array of black-or-white pixels” (not “First I will build a neural net” because this is a procedure and a design element rather than a requirement). The *clarity* and properly declarative form of your requirements are especially relevant.

### 1.3.1 Definite Requirements

Below are the definite requirements for Bid Estimator –

1. Bid estimator accepts the building type, the year the property was built, the area of the property (in sq. foot), number of bedrooms and bathrooms.
2. Output of BE is best price ($ value), to offer for the input property.
3. Accepted building types are Townhouse and Single Family.

### 1.3.2 Requirements Not Classified Yet

At this time, below are the requirements which are unclassified –

1. Include median sale price for houses of similar configuration between 2005 – 2010
2. Include median price change for houses of similar configuration between 2005 - 2010

### 1.3.3 Nice-to-do Requirements

Below are the nice to have requirements for Bid Estimator –

1. Predict the best price of the property 5 years down the line from current year. For e.g. given the price predicted in 2020 is $ 700,000, the value of this property in 2025 will be $ 780,000
2. Accept building types – Condo, Multi Family.
3. Suggest neighborhoods where houses of this configurations were sold between 2005-2010.

## 1.4 HOW SUCCESS WILL BE ASSESSED

Explain, as specifically as possible how success of the project will be assessed. Quantification is ideal. We realize that you can’t know at this stage what realistic goals are, nor will we evaluate you project on this a lot but we want you to think this through. An example is “90% successful recognition of a cat in 1000 random images containing animals taken from the Web.” The *clarity*, especially this specificity, of your assessment is especially relevant.

Ideally, prediction of the offer price will be considered a success, if the offer price matches the sales price of the property. For this project an accepted accuracy of the predicted value will be at least 80% of the sales price.

## 1.5 TECHNOLOGY EXPLANATION

Explain what two machine learning technologies you are seriously considering--and why you feel they apply. One technology may be emphasized as the implementation and the other as an alternative for discussion. The *technical correctness* in this part is especially relevant, including your explanation of “why.”

Neural Network ML technology will be the main technology for the Bid price estimation.

Practically, the bidding process of houses on sale, do not have a systematic process. Interested buyers place their best price based on their judgement about the best value, at which they can ensure they are the winners in the bidding process. This uncertainty in the process makes it a suitable candidate for Neural Net, as neural nets do not have a model of how the process works. Also, the both the inputs and outputs can be easily represented as vectors, which meets the requirement for applying Neural Nets [2].

As an alternative, Linear Regression and SVM’s [3] would be other obvious choices for this prediction as this is a regression problem. In both these technologies, we try to find a mathematical function or a hyper plane, which when supplied with an arbitrary input, returns an output which would the offer price for the house.

## 1.6 DATA SOURCES

Explain whether or not your project requires data. If so, describe specifically were you will obtain it. If you intend to gather the data yourself, please explain how this will be practical. The *clarity*, especially this specificity, of your account is especially relevant.

This project uses the data from Kaggle Website. Link to data set - <https://www.kaggle.com/c/house-prices-advanced-regression-techniques/overview>

This is a data set contains 79 variables describing aspects of residential homes in Ames, Iowa [1].

## 1.8 REFERENCES FOR PROPOSAL PHASE

Fill in the following, and cite each reference at least once (e.g., “[2]“) within the text. References can include specific places in the notes and textbook. Note that “Use of Resources” is an entire evaluation criterion equal in weight to the others. This section is not hard to do, so it’s easy to gain points (and the use of references always improves a paper in any case). On the other hand, if you don’t complete this section, the grade could suffer up to 20%. There should be a minimum of three meaningful references.

[1] Competition Description section at <https://www.kaggle.com/c/house-prices-advanced-regression-techniques/overview>

[2] Study Guide, Module 1, Introduction to Neural Nets

[3] Study Guide, Module 1, Introduction to Classification

## 1.7 Staff Evaluation of Assignment 1



# ASSIGNMENT 2: PROJECT PROPOSAL PLUS

## 2.0 WHAT’S CHANGED

Provide no more than a page of 12-point type explaining what has been changed or added since assignment 1. Include in this whether and how the material in module 2 influenced this, or refer to reading that you did in working on this assignment (#2).

Changes on this document –

Section 2.1, 2.2 – No changes

Section 2.3 – Updated as per comments on Assignment 1.

## 2.1 SUMMARY DESCRIPTION, VERSION 2

One- or two-paragraph overall description of your proposed term project.

Buying a house is one of the most expensive investments that an individual makes in a lifetime. During the buying process, the buyer makes an offer based on the quoted Sales Price. The seller, on the other hand, gets to decide on the final price at which they would sell the property, out of multiple offers from interested buyers.

## As a buyer, it is often challenging, to come up with a reasonable offer which will be accepted by the seller. Bid Estimator (BE) helps the buyer to estimate the offer amount, based on the recent sales of comparable properties in Ames, Iowa.

## 2.2 I/O EXAMPLES, VERSION 2

At least two specific examples of projected output for designated input. You will not be held to this—it is just explanatory at this point.

Below are sample input/output examples generated by BE –

1. Sample Input/Output 1 –

Building Type – Single Family, Year Built – 1990, Sq. Footage – 1200 sq. ft, Bedrooms – 4, Bathrooms – 3

Output A - $ 700,000 with confidence 80%

1. Sample Input 2 –

Building Type – Town House, Year Built – 1950, Sq. Footage – 1200 sq. ft, Bedrooms – 3, Bathrooms – 2

Output - $ 659,000 with confidence 82%

## 2.3 FUNCTIONAL REQUIREMENTS, VERSION 2

High-level functional requirements statement in two roughly equal numbered lists, organized by triage. Separate your requirements into two approximately even categories (select modest “definite” requirements, otherwise “nice-to-do”). This organization allows you to first attain readily do-able goals without getting bogged down, and then move on to other goals if you can. State requirements in declarative language as advised in assignment 1. Giving each requirement a label (e.g., “(Recognize 0-9):”) helps with clarity and readability.

Your response replaces this.

### 2.3.1 Definite Functional Requirements (first priority)

### 2.3.1.1 Definite Requirement #1: Sales Price Estimation

Bid Estimator accepts the building type, the year the property was built, the area of the property (in sq. foot), number of bedrooms and bathrooms.

### 2.3.1.2 Definite #2: Restrictions on input building types

Bid Estimator will allow Townhouse and Single Family as inputs for Building type.

### 2.3.2 Nice-to-do Functional Requirements (second priority)

### 2.3.2.1 Nice-to-do Requirement #1: Sales Price prediction for next 5 years

Predict the best price of the property 5 years down the line from current year. For e.g. given the price predicted in 2020 is $ 700,000, the value of this property in 2025 will be $ 780,000

### 2.3.2.2 Nice-to-do #2: Extend predictions for other Building types

The application would be extended to accept Condo and Multi Family building types, as inputs.

## 2.4 V2: HOW SUCCESS WILL BE ASSESSED

Explain, as specifically as possible (quantification is ideal) how success of the project should be assessed. Avoid an aimless project of the form “I’ll play around with X until time runs out” because this is less motivational, and because you learn less.

Ideally, prediction of the offer price will be considered a success, if the offer price matches the sales price of the property.

For this project accepted error will be $2000 above or below the sales price. So, for e.g. if the Sales Price for the house is $700,000, a predicted value of $680,000 - $720,000 will be accepted.

The overall accuracy of prediction will be 80%, i.e. Out of 100 houses, this application will be able to predict Sales prices for 80 houses accurately.

As per similar experiments done in past, we have seen prediction accuracy of 80% and more [1][2].

## 2.5 V2 TECHNOLOGY EXPLANATION

Explain what two technologies you intend to use--and why you feel they apply to your particular project. One of the two may be emphasized as the implementation and the other as an alternative or as a complement—discussed but not implemented if need be. If possible, show fragments of code execution. For example, if you are using TensorFlow, show that you have run some code. This can be simple—we just want you to break the ice with implementation.

Neural Network ML technology will be the main technology for the Bid price estimation. The Neural Net (NN) will be modelled with an input layer consisting of one neuron per feature in the data set. The output will be the sales price, hence just one value. The NN will have one or more hidden layers in between input and output layers.

Practically, the bidding process of houses on sale, do not have a systematic process. Interested buyers place their best price based on their judgement about the best value, at which they can ensure they are the winners in the bidding process. This uncertainty in the process makes it a suitable candidate for Neural Net, as neural nets do not have a model of how the process works.

Neural Nets are able to approximate the mapping of arbitrary nonlinear variables (like sq foot, number of rooms, etc.) [2]. Neural networks are good to model with nonlinear data with large number of inputs; for example, images. It is reliable in an approach of tasks involving many features [6]. Also, the both the inputs and outputs can be easily represented as vectors, which meets the requirement for applying Neural Nets [3]. Once trained, the predictions are pretty fast.

We find price estimations done with considerable accuracy using Neural Nets from past work on Singapore public housing [1]. As per Lipo Wang, the NN architecture 6-8-1 (number of neurons per layer), gave an MSE of 3.882 [1].

Another experiment done using Kaggle-kc house Dataset shows comparison between Multiple Linear Regression (MLR) and NN, where accuracy of NN was 81% with respect to accuracy of 69% or MLR [2].

Multiple Linear Regression [2] and SVM’s [4] would be alternate choices for this prediction as this is a regression problem. In both these technologies, we try to find a mathematical function or a hyper plane, which when supplied with an arbitrary input, returns an output which would the offer price for the house. We assume there is a correlation between the independent and dependent variables [2]. SVMs can model slightly more complex functions using kernels.

~~Finally, Neural Nets can handle multi-class problems by producing probabilities for each class. In contrast, SVMs handle these problems using independent one-versus-all classifiers where each produces a single binary output [7]. As this application will need to consider multiple input variables to determine the Sales price, we would be using Neural Nets for this application~~.

## 2.6 V2 DATA SOURCES

Explain whether or not your project requires data. If so, describe were you will obtain it. Be careful about this because you won’t have a project if it needs data and you have to spend too much time hunting and gathering it.

This project uses the data from Kaggle Website. Link to data set - <https://www.kaggle.com/c/house-prices-advanced-regression-techniques/overview>

This is a data set contains 79 variables describing aspects of residential homes in Ames, Iowa [3].

## 2.8 REFERENCES FOR PROPOSAL V2

Fill in, and also cite each of the following (e.g., “[2]”) within the text. References can include specific places in the notes and textbook. You are free to include references used in the prior assignment version. Keep in mind that this “use of resources” is a whole evaluation criterion.

[1] <https://towardsdatascience.com/house-prices-prediction-using-deep-learning-dea265cc3154> (bottom section)

[2]<https://www.researchgate.net/publication/309443696_Housing_price_prediction_using_neural_networks> ( Page 2, table with MSE data)

[3] Competition Description section at <https://www.kaggle.com/c/house-prices-advanced-regression-techniques/overview>

[4] Study Guide, Module 1, Introduction to Neural Nets

[5] Study Guide, Module 1, Introduction to Classification

[6]<https://subscription.packtpub.com/book/big_data_and_business_intelligence/9781788397872/1/ch01lvl1sec27/pros-and-cons-of-neural-networks>

[7] <https://www.pico.net/kb/advantages-of-artificial-neural-networks-over-support-vector-machines#:~:text=We%20can%20summarize%20the%20advantages,produces%20a%20single%20binary%20output.&text=SVM%20models%20are%20easier%20to%20understand>.

…

## 2.7 Instructor’s Evaluation of Assignment 2



# Assignment 3: Project Design, version 1

Keep in mind the evaluation matrix at the end as you do the work and use it to guide what you submit. Use no more than 5 pages of 12-point text excluding figures. You may include as many appendices as you wish for reference. Parts of these may be read as needed.

NOTE: The predicted Sales Price is the best bid price, hence these terms are used interchangeably in this document.

## 3.1 Final Functional requirements

List your final functional requirements, numbering them in the form DX and NX where:

D/N means “Definite” / “Nice to do” (two categories, not three)

X=L and the goal is a *learning* goal – OR – X=F and the goal is *functional*

A “learning goal” is not a traditional functional requirement. It’s a skill, technology, or tool that you want to know by the time the course is over. There should be 3-7 items in the “Definite” list and at least 4 in the “Nice to do” list. You will reference these numbered functional requirements in phase 3 when you will be asked to show what the project accomplished.

For example:

3.1.2 (DF) Wine Suggestion

The application shall suggest a wine with the dinner.

3.1.6 (NL) TPU Porting

The application will be ported to Google’s TPU Console to assess the obstacles in doing so, and to measure the difference in execution compared with my GPU-enabled laptop.

You will reference these numbered functional requirements throughout the rest of the term, when you will be asked to show what the project accomplished.

3.1.1 (DF) Sale Price Determination

Bid Estimator accepts the building type, the year the property was built, the area of the property (in sq. foot), number of bedrooms and bathrooms to determine Sales Price for the property.

### 3.1.2 (DF) Restrictions on input building types

Bid Estimator will allow Townhouse and Single Family as inputs for Building type.

3.1.3 (DL) Applying Neural Nets for Regression problems

Neural nets will be the employed to predict the Sales Price for this project. The goal is to learn how Neural Nets are designed to solve regression problems.

3.1.4 (DL) Using Neural Network with Tensor Flow (Keras)

Neural Net models will be implemented using Keras from Tensor Flow python library.

The intent is to explore and learn a new technology during the course of this project.

3.1.5 (NL) Implementing SVM and Multiple Linear Regression using Tensor Flow

As this project can also be done using SVM and Linear Regression models, hence the goal will be to implement that using Tensor Flow library and compare accuracy of these models with respect to Neural Nets.

## 3.2 Design and Theory

Describe the design of your proposed system. Use at least one annotated diagram. Explain the theory behind your particular design without repeating or paraphrasing material already in the notes. Explain how two technologies—at least one implemented—will interface or compare. The reader should understand how you plan to fit the pieces together. Show this at a high level, as well as providing as much detail as you can at this point. Include at least one (meaningful) figure, for example, using boxes and arrows showing data flow as in the figure below (which you may edit to construct yuours if applicable).



Bid Estimate will be using Neural Nets for Sales Price prediction. The estimation process will involve the following components –

1. Housing Data – The csv file containing the housing data.
2. Data cleanup and pre-processing – This step will involve using Tensor Flow Transform to normalize the input features, along with converting stings to integers.
3. Forward Feed Neural Net – The actual prediction process, with the normalized input data to determine the Sales Price. The number of features which will be used as input hasn’t been finalized yet. Also, the number of hidden layers and number of neurons in the hidden layers are yet to be decided. The output layer contains only 1 node which is the final value evaluated by NN.

Based on the current understanding of Neural Nets, below is a high-level design of the BE.

Out of 79 features, relevant features will be selected for price prediction

Sales price will be predicted based on the Neural Net output.

Data Cleanup and Pre processing

Sales Price

Forward feed Neural Net

Housing Data

Forward Feed Neural Net Design (This is not the final architecture, number of input neurons, hidden layers are unknown at this time)

Output of Neural Net

Hidden layers (Number of layers and number of neurons are not finalized yet)

Input layer with features (Up to 79 neurons)

Multiple Linear Regression [2] and SVM’s [4] would be alternate choices for this prediction as this is a regression problem. In both these technologies, we try to find a mathematical function or a hyper plane, which when supplied with an arbitrary input, returns an output which would the offer price for the house. We assume there is a correlation between the independent and dependent variables [2]. SVMs can model slightly more complex functions using kernels.

Based on an experiment done using Kaggle-kc house Dataset shows comparison between Multiple Linear Regression (MLR) and NN, where accuracy of NN was 81% with respect to accuracy of 69% or MLR [2]. Hence NN is chosen as the ML model here. The other reason for choosing NN is for self-learning the challenges involved with implementation of Neural Nets using Tensor Flow.

## 3.3 Tools

Describe the tool(s) such as TensorFlow, Pyevolve, and Python that you will probably use, or explain why you will build from scratch. It is OK if you say "I will use tool 1 or tool 2." Support the fact that you have reasonably investigated and tried out tools. Explain your choice. Show samples that make you and us reasonably confident of your choices. Show that you understand how the tools work.

Keras from Tensor Flow Library will be used for the implementation of pre-processing and Neural Net modelling for this project.

For Pre-processing, TensorFlow Transform [1] will be used to –

* Normalize an input value by using the mean and standard deviation
* Convert strings to integers by generating a vocabulary over all of the input values

For the Neural Net implementation, Keras will be used. Input vectors and initial weight and bias vectors will be setup to initialize the NN. Based on the error calculated using MSE, the weights will be adjusted to reduce the error, and the process will be repeated until an accuracy of 80% is reached.

For this project accepted error will be $2000 above or below the sales price. So, for e.g. if the Sales Price for the house is $700,000, a predicted value of $680,000 - $720,000 will be accepted.

The overall accuracy of prediction will be 80%, i.e. Out of 100 houses, this application will be able to predict Sales prices for 80 houses accurately.

Below are couple of samples where Neural Network has been used with considerable accuracy for House Price prediction –

1. Singapore House price prediction, with a MSE of ~3.8 for a 6-8-1 architecture.[2]
2. Kaggle-kc house Dataset using NN, where accuracy of NN was 81% with respect to accuracy of 69% or MLR [3]

## 3.4 Risk Retirement

We don’t want you to get stuck in your project, so you should try to deal with the most serious obstacles well before you encounter them. Identify and prioritize the three top risks in carrying out the project. Try as best you can to retire (set to rest) the top one or two by the time you submit this, by means of experiments, prototypes, or work-arounds. Explain how you did this. Explain how you will retire the remaining ricks in advance.

3.4.1 First project on Neural Network and Tensor Flow

This is my first implementation of Neural Net as ML model and Tensor Flow as Technology. Hence, I am expecting challenges during implementation, which may restrict my model to reach 80% accuracy.

Based on the examples researched [2], [3], similar work has been done using Neural Network and Tensor Flow. So when stuck on a challenge, I would refer to these solutions for their approaches to the implementation.

3.4.2 Lack of sufficient data for training

There may be a case possible, where additional data would be required to train the NN. If such a case arises, housing data will be used from <https://www.redfin.com/news/data-center/> and Generative Adversarial Networks (GAN’s) can be used for additional data for NN.

## 3.5 Schedule

Explain in outline the steps you intend to take to carry out the project. Show the completion of the stages. Include a schedule, as detailed as can be reasonably foreseen.

Below table highlights the schedule of the project –

|  |  |  |
| --- | --- | --- |
| Step | Determination of completion | Tentative Completion Date |
| Finalize Input Data sets | 1 csv file containing the building type, the year the property was built, the area of the property (in sq. foot), number of bedrooms and bathrooms to determine Sales Price for the property. | 11/30 |
| Pre-Processing | Normalized input data, EDA to visualize data | 12/3 |
| Neural Net Setup and Prediction | Sales price is predicted with 80% accuracy | 12/10 |

## 3.6 References

Add to your references. Instructions as before.

[1] <https://www.tensorflow.org/tfx/tutorials/transform/census> (TensorFlow Transform)

[2] <https://towardsdatascience.com/house-prices-prediction-using-deep-learning-dea265cc3154> (bottom section)

[3]<https://www.researchgate.net/publication/309443696_Housing_price_prediction_using_neural_networks> ( Page 2, table with MSE data)

## 3.7 Evaluation of Assignment 3



# Assignment 4: Project Design Plus, version 2

Please limit this to 6 pages of 12-point text excluding figures, references, and appendices. This revision is your final view of the design prior to implementation (though you may still change it when you implement).

## 4.1 Version 2 Final Requirements

List your final requirements, refined again if appropriate, using the format of section 3.1 again.

4.1.1 (DF) Sale Price Determination

Bid Estimator accepts the building type, the year the property was built, the area of the property (in sq. foot), number of bedrooms and bathrooms to determine Sales Price for the property.

### 4.1.2 (DF) Restrictions on input building types

Bid Estimator will allow Townhouse and Single Family as inputs for Building type.

4.1.3 (DL) Applying Neural Nets for Regression problems

Neural nets will be the employed to predict the Sales Price for this project. The goal is to learn how Neural Nets are designed to solve regression problems.

4.1.4 (DL) Using Neural Network with Tensor Flow (Keras)

Neural Net models will be implemented using Keras from Tensor Flow python library.

The intent is to explore and learn a new technology during the course of this project.

4.1.5 (NL) Implementing SVM and Multiple Linear Regression using Tensor Flow

As this project can also be done using SVM and Linear Regression models, hence the goal will be to implement that using sci-kit python library and compare accuracy of these models with respect to Neural Nets.

## 4.2 V2 Design and Theory

Describe the final pre-implementation version of the design of your proposed system. Use annotated diagrams. Explain the theory behind your design. Explain how the two technologies will interface or compare. The reader should understand how you plan to fit the pieces together. Show this at a high level, as well as providing as much relevant detail as you can. Include at least one (meaningful) figure.

Bid Estimate will be using Neural Nets for Sales Price prediction. The estimation process will involve the following components –

1. Housing Data – The csv file containing the housing data.
2. Data cleanup and pre-processing – This step will involve using Tensor Flow Transform to normalize the input features, along with converting stings to integers.
3. Forward Feed Neural Net – The actual prediction process, with the normalized input data to determine the Sales Price. The Neural Net consists of the following –

* Hidden layer 1: 32 neurons, ReLU activation
* Hidden layer 2: 32 neurons, ReLU activation
* Output Layer: 1 neuron, Sigmoid activation

The reason for using 2 hidden layers, with 32 neurons each, has been chosen based on credible accuracy of 88% reported on similar experiments using this neural net architecture. [4]

Based on the current understanding of Neural Nets, below is a high-level design of the BE.

Feed Train data into Neural Net

Out of 79 features, relevant features will be selected for price prediction

Split data randomly into 80% train data and 20% test data

Data Cleanup and Pre processing, Feature Selection

Sales Price

Forward feed Neural Net

Housing Data

Output the Sales Price and measure accuracy of prediction

**Forward Feed Neural Net Design** (picture [4])

Diagram, schematic

Description automatically generated

79)

## 4.3 V2 Tools

Describe the tool(s) (e.g., TensorFlow) you will definitely use, or explain why you will build from scratch. Support the fact that you have reasonably investigated and tried out tools. Explain your choice. Show samples that make you and your reader reasonably confident of your choices.

## Keras from Tensor Flow Library will be used for the implementation of pre-processing and Neural Net modelling for this project. Pandas and Numpy will be used for vector processing. Matplotlib will be used for plotting the accuracy of learning models.

## For Pre-processing, TensorFlow Transform [1] will be used to –

* Normalize an input value by using the mean and standard deviation

## Convert strings to integers by generating a vocabulary over all of the input values

## Fill missing data with missing values from other data points

Data will be split into 80% training set and 20% testing set.

## For the Neural Net implementation, Keras will be used. A Sequential Keras model will be implemented using Dense Layers. Each dense layer will consist of 32 tensors each with ReLU activation functions. The output tensor will use sigmoid activation function.

For optimization of the model, we will use Gradient Descent (sgd optimizer) and use cross entropy loss function. The optimizer will track the accuracy during the training phase of the model.

## Input vectors and initial weight and bias vectors will be setup to initialize the NN. Based on the error calculated using MSE, the weights will be adjusted to reduce the error, and the process will be repeated until an accuracy of 80% is reached.

## For this project accepted error will be $2000 above or below the sales price. So, for e.g. if the Sales Price for the house is $700,000, a predicted value of $680,000 - $720,000 will be accepted.

## The overall accuracy of prediction will be 80%, i.e. Out of 100 houses, this application will be able to predict Sales prices for 80 houses accurately.

## Below are couple of samples where Neural Network has been used with considerable accuracy for House Price prediction –

## a. Singapore House price prediction, with a MSE of ~3.8 for a 6-8-1 architecture.[2]

## b. Kaggle-kc house Dataset using NN, where accuracy of NN was 81% with respect to accuracy of 69% or MLR [3]

For SVM and Linear Regression implementations, we would use Sci-Kit Linear Regression and Support Vector Classifiers. Preprocessing and data split will involve the same steps as discussed for NN above. Linear and Gaussian Kernels will be used for SVM and the models will be trained and tested. Accuracy for these models will be evaluated against the accuracy of Neural Net model.

## 4.4 Implementation Fragments

Show enough *parts* of your implementation—or a simplified form of it—to convince yourself and the reader that you will have the implementation of the definite requirements completed on time. Cut and paste commented code below.

Below is a SVM implementation showing the training and testing –

import pandas as pd

import numpy as np

import sklearn.svm as svm

from sklearn.preprocessing import StandardScaler

from sklearn.model\_selection import train\_test\_split

from sklearn.metrics import confusion\_matrix

# Split dataset

X\_train, X\_test, Y\_train, Y\_test = train\_test\_split(X, Y,

test\_size = 0.2)

# Scale X\_train values

scaler = StandardScaler()

scaler.fit(X\_train)

X\_train\_scaled = scaler.transform(X\_train)

# Create classifier

if kernel == 'poly':

svm\_classifier = svm.SVC(kernel=kernel, degree= 3)

else:

svm\_classifier = svm.SVC(kernel=kernel)

svm\_classifier.fit(X\_train\_scaled, Y\_train)

# Scale X\_test values

X\_test\_scaled = scaler.transform(X\_test)

# Predict

predicted = svm\_classifier.predict(X\_test\_scaled)

Below is the Neural Net implementation [4] –

[N.B. Neural Net implementation is a work in progress at this time]

from keras.models import Sequential  
from keras.layers import Dense

model = Sequential([  
 Dense(32, activation='relu', input\_shape=(10,)),  
 Dense(32, activation='relu'),  
 Dense(1, activation='sigmoid'),  
])

model.compile(optimizer='sgd',  
 loss='binary\_crossentropy',  
 metrics=['accuracy'])

## 4.4 V2 Risk Retirement

Identify and prioritize the 5 top risks in carrying out the project. Try as best you can to retire the top four risks by the time you submit this, by means of experiments, prototypes, or work-arounds. Explain how you did this. Explain how you will retire the remaining risks in advance. Update this from the version in Assignment 3.

4.4.1 First project on Neural Network and Tensor Flow

This is my first implementation of Neural Net as ML model and Tensor Flow as Technology. Hence, I am expecting challenges during implementation, which could restrict my model to reach 80% accuracy.

Based on the examples researched [2], [3], similar work has been done using Neural Network and Tensor Flow. So when stuck on a challenge, I would refer to these solutions for their approaches to the implementation.

The types of challenges I am expecting at this time are –

1. Keras Runtime error – Runtime errors could be resolved by first creating a simple NN implementation using Keras and make it up and running. This will give enough idea to know the steps involved for building a NN using Keras and applying that to the housing data set.
2. Low Accuracy on the model – This would involve tweaking the hyper parameters, in this case, using different features or increasing/decreasing the hidden layers to verify if the accuracy has improved. Given we have enough examples, this should be be a easy problem to solve. [2][3][4]

4.4.2 Lack of sufficient data for training

As we know, Neural Nets require lots of data to return better accuracy. If NN performs better on the training data set, i.e. if the MSE of the training data set is low as compared to the accuracy, when executed on the test data, this may mean that data quality used for training the model was not good, or overfitting has occurred with the training dataset.

In such cases, housing data will be used from <https://www.redfin.com/news/data-center/> and Generative Adversarial Networks (GAN’s) can be used for additional data for NN.

## 4.5 V2 Schedule

Explain in outline the updated steps you intend to take to carry out the project. Show the completion of the stages. Include a schedule, as detailed as can be reasonably foreseen.

Below table highlights the schedule of the project –

|  |  |  |
| --- | --- | --- |
| Step | Determination of completion | Tentative Completion Date |
| Finalize Input Data sets | 1 csv file containing the building type, the year the property was built, the area of the property (in sq. foot), number of bedrooms and bathrooms to determine Sales Price for the property. | 11/30 |
| Pre-Processing | Normalized input data, EDA to visualize data | 12/3 |
| SMV and Linear Regression Models | Evaluate SVM and Linear Regression models | 12/3 |
| Neural Net Setup and Prediction | Sales price is predicted with 80% accuracy | 12/10 |

## 4.6 V2 References

Add to your references. Instructions as above.

[1] <https://www.tensorflow.org/tfx/tutorials/transform/census> (TensorFlow Transform)

[2] <https://towardsdatascience.com/house-prices-prediction-using-deep-learning-dea265cc3154> (bottom section)

[3]<https://www.researchgate.net/publication/309443696_Housing_price_prediction_using_neural_networks> ( Page 2, table with MSE data)

[4] <https://hackernoon.com/build-your-first-neural-network-to-predict-house-prices-with-keras-3fb0839680f4>

## 4.7 Evaluation of Assignment 4



# Implementation version 1

## 5.1 Summary v1

In a paragraph or two, summarize the outcome of your project functionally and learning-wise. Avoid duplication with Section 5.3 below, so exclude details here.

Functionally, Bid Estimator (BE) was able to predict the house prices with an accuracy of ~85% using Neural Net Architecture. All the functional requirements have been met successfully. Currently, work is in progress to compare the Neural Net model with SVN model, and generate a comparison graph between the two of them.

Learning wise Feed forward neural net architecture understanding was established with the implementation of this project using Keras package from Tensor Flow Python library. The steps involving model creation, compilation, fitting and evaluation were done to understand on implementation of a Neural Net. SVM from Sci-kit library was implemented and he steps involved in building a SVM model was also manifested.

## 5.2. Report on Requirements v1

Please copy each definite requirement—"DX" (X = F or L)—from Assignment and explain the extent to which you accomplished each definite requirement *so far*. For each, include 1-4 sentences and screenshot of the relevant IO or key proof. Your effectiveness depends largely on how much you demonstrate that you learned.

4.1.1 (DF) Sale Price Determination

Bid Estimator accepts the building type, the year the property was built, the area of the property (in sq. foot), number of bedrooms and bathrooms to determine Sales Price for the property.

Below is the features used by Bid Estimator which are used for predicting the Sales Price of the property. The selected features include -

* Area of the property (in sq. foot) -> Lot Area
* Number of bed rooms -> BedroomAbvGr
* Number of bath rooms -> Full Bath, Half Bath

Year the property was built was not used in the prediction as so far, I have not been able to scale it uniformly with respect to other features. This is currently a work in progress.



The Machine Learning model was able to predict the Sales Price with an accuracy of ~85%. Below is a screenshot of one of its executions.

![Graphical user interface, text

Description automatically generated]()

### 4.1.2 (DF) Restrictions on input building types

Bid Estimator will allow Townhouse and Single Family as inputs for Building type.

From the housing data set, we have selected building types which were only Townhouse and Single Family. The data set consisted of 84% data for Single Family homes and 8% data for Townhouse. Other building types were 9%. These 9% records were removed from the data set as part of pre processing.

4.1.3 (DL) Applying Neural Nets for Regression problems

Neural nets will be the employed to predict the Sales Price for this project. The goal is to learn how Neural Nets are designed to solve regression problems.

Feed Forward Neural Net Architecture was implemented for this project. A sequential model was designed with layers having different activation functions (more on this follows in Design), and a clear understanding on how Neural Nets are optimized using stochastic gradient descent was implemented. This gave me the learning opportunity to work with Neural Nets.

4.1.4 (DL) Using Neural Network with Tensor Flow (Keras)

Neural Net models will be implemented using Keras from Tensor Flow python library.

The intent is to explore and learn a new technology during the course of this project.

Keras from Tensor Flow Python library was used for this implementation. Having worked on Scikit Python library in past, I found the experience with Keras quite similar to Scikit, with methods like fit(). This learning will be useful to explore Neural Nets in future for other Machine Learning models.

## 5.3 Report on Design v1

Describe the design that you have used so far. Indicate how, where, and why it has differed (thus far) from your planned design. Describe its advantages and its shortcomings. Include a description of how the technologies you explored (not the tools—those are described below) have worked out so far. Include at least one diagram.

Bid Estimate will be using Neural Nets for Sales Price prediction. The estimation process will involve the following components –

1. Housing Data – The csv file containing the housing data.
2. Data cleanup and pre-processing – This step will involve using Tensor Flow Transform to normalize the input features, along with converting stings to integers.
3. Forward Feed Neural Net – The actual prediction process, with the normalized input data to determine the Sales Price. The Neural Net consists of the following –

* Hidden layer 1: 32 neurons, ReLU activation
* Hidden layer 2: 32 neurons, ReLU activation
* Output Layer: 1 neuron, Sigmoid activation

The reason for using 2 hidden layers, with 32 neurons each, has been chosen based on credible accuracy of 88% reported on similar experiments using this neural net architecture. [4]

Out of 79 features available in the dataset, 10 features were selected based on my understanding of the real estate pricing. The selected features impact the Sales Price for a property. As a future work, it would be nice to apply feature selection methods using Sklearn’s feature selection library.

Feed Train data into Neural Net

Out of 79 features, 10 features were selected for price prediction

Split data randomly into 80% train data and 20% test data

Data Cleanup and Pre processing, Feature Selection

Sales Price

Forward feed Neural Net

Housing Data

Output the Sales Price and measure accuracy of prediction

**Forward Feed Neural Net Design** (picture [4])

Diagram, schematic

Description automatically generated

The initial prosed design was followed as per discussed in Assignment 4. The goal at this time was to ensure that a working Neural Net model is implemented for the housing data set, which would now allow for experiments. As I understand, there are a lot of hyper parameters which can be modified for the current implementation, based on which the accuracy of the model would be impacted.

Advantages of this design –

* The design is simple and easy to understand. My clear purpose was to create a proof of concept that a regression problem can be solved using Neural Nets. The choice of 2 hidden layers was made for experimentation.
* The design meets the goal which was set out for the task. It is easy to comprehend for me at this time of my understanding on Neural Nets and Machine Learning.

Shortcomings of this design –

* This design takes an intuitive approach over the Feature Selection. Currently features are chosen based on my understanding of house pricing. The selected features are known to contribute to the value of a property. As an improvement to this, various automated feature selection techniques like sklearn’s SelectKBest or f\_regression can be used for feature selection.
* The output of this model is not intelligible. This is a common issue with all Neural Net architectures as the output of these architectures are hard to explain.
* At this moment the model is not considering the year of make for the property. I am working on this aspect of the design to include this into the implementation. The challenge is I am not able to scale it as compared to other features in the data set.

Technologies – Neural Net and SVM

For this design Neural Net and Support Vector Machines were used for the house price prediction. Neural Nets have been more accurate (~85%) as compared to a Polynomial Kernel used for SVM which whose accuracy was ~70%.

## 5.4 Tools v1

Describe the tool(s) that you are using. Show samples. Describe their advantages and their shortcomings. Limit: 1 page of 12-point Times New Roman.

Keras from Tensor Flow – For this project Keras was used for a Feed Forward Neural Net implementation.

For pre processing, python packages pandas and sklearn were used.

For plotting the accuracy and loss between validation and training sets, pyplot from matplotlib was used.

Below are the details of the implementation illustrating the tool usage –

Data from the csv file was read using pandas read\_csv() function. This function loads the data from the csv file into a data frame.

df = pd.read\_csv('housepricedata.csv')

The dataset contains column values which are in different scales. For e.g. Lot Area and Floor Area of a property are numeric values in a scale of thousands as compared to number of rooms and bathrooms which would be in a value between 2 to 6.

So, we normalize the data set, using the preprocessing functions of sklearn.

min\_max\_scaler = preprocessing.MinMaxScaler()

X\_scale = min\_max\_scaler.fit\_transform(X)

Here, min max scaler is used to normalize the data for all the features in our data set.

Now, we split the data into training and testing sets. Training sets are further broken into training and validation steps.

For train/test and train/validation split, sklearn’s train\_test\_split libraries are used –

X\_train, X\_val\_and\_test, Y\_train, Y\_val\_and\_test = train\_test\_split(X\_scale, Y, test\_size=0.3)

X\_val, X\_test, Y\_val, Y\_test = train\_test\_split(X\_val\_and\_test, Y\_val\_and\_test, test\_size=0.5)

Now, the below steps follow for Neural net architecture, optimization and finally evaluation.

Sequential layers are constructed with 2 dense (hidden) layers with ReLU activation functions. These layers have 32 neurons each.

model = Sequential([

    Dense(32, activation='relu', input\_shape=(10,)),

    Dense(32, activation='relu'),

    Dense(1, activation='sigmoid'),

])

The hyper parameters for the models are configured using stochastic gradient descent for optimization, and binary\_crossentropy loss.

model.compile(optimizer='sgd',

              loss='binary\_crossentropy',

              metrics=['accuracy'])

Finally, the model is fit with the training data and validation data –

hist = model.fit(X\_train, Y\_train,

          batch\_size=32, epochs=100,

          validation\_data=(X\_val, Y\_val))

At this stage, our model is has completed the learning phase. At each epoch the weights of the neural nets have been altered to reduce the binary\_crossentropy loss.

Finally, we evaluate the model with the test data set.

model.evaluate(X\_test, Y\_test)[1]

Below are the steps used for SVM implementation –

# Scale X\_train values

scaler = StandardScaler()

scaler.fit(X\_train)

X\_train\_scaled = scaler.transform(X\_train)

# Create classifier

if kernel == 'poly':

svm\_classifier = svm.SVC(kernel=kernel, degree= 3)

else:

svm\_classifier = svm.SVC(kernel=kernel)

svm\_classifier.fit(X\_train\_scaled, Y\_train)

# Scale X\_test values

X\_test\_scaled = scaler.transform(X\_test)

# Predict

predicted = svm\_classifier.predict(X\_test\_scaled)

Advantages of Neural Nets over SVM –

I found Neural Net documentation easier to follow, and the steps easier to understand. SVM on the other hand was a bit tricky.

Also based on a quick run of both the models, based on the same preprocessing done on the data, we have a high accuracy for Neural Net as compared to SVM.

# Instructor’s Evaluation



12/1/2020 11:05 AM

# Implementation, version 2

Excluding appendices and figures, this response should not exceed 6 pages.

## 6.1 Summary v2

In a paragraph or two, summarize the outcome of your project functionally and learning-wise but avoid duplication with Section 6.3 below (which is detailed). Underline edited sentences and additions from v1, if any.

Bid Estimator (BE) aims at predicting whether a buyer should quote above or below the list price of a single family property. BE accepts the house configuration consisting of lot area, overall condition, total basement sq. ft, number of bathrooms, bedrooms, fireplaces and garage area to predict buyer’s decision.

Functionally, Bid Estimator (BE) was able to predict the house prices to be below or above the median value with an accuracy of ~85% using Neural Net Architecture and ~88% using SVM.

A prediction is accurate if the model is able to predict if the house price should be below or above the median value given house configuration. So, and accuracy of above 85% means, that the model is able to say with 85% certainty if the buyer should quote above or below the median price.

Learning wise Feed forward neural net architecture understanding was established with the implementation of this project using Keras package from Tensor Flow Python library. The steps involving model creation, compilation, fitting and evaluation were done to understand on implementation of a Neural Net. SVM from Sci-kit library was implemented and the steps involved in building a SVM model was also manifested.

## 6.2. Report on Requirements v2

Re-list and explain the extent to which you accomplished each definite requirement "DiX" (X = F or L), as well as any other fulfilled requirements. For each, include 1-4 sentences and screenshot(s) (including of command-line text). Underline edited sentences and additions from v1, if any.

4.1.1 (DF) Sale Price Determination

Bid Estimator accepts the building type, the year the property was built, the area of the property (in sq. foot), number of bedrooms and bathrooms to determine Sales Price for the property.

Below is the features used by Bid Estimator which are used for predicting the Sales Price of the property. The selected features include -

* Area of the property (in sq. foot) -> Lot Area
* Number of bed rooms -> BedroomAbvGr
* Number of bath rooms -> Full Bath, Half Bath

Below is a snapshot of the data set -



The Neural Net and SVM models were able to predict the bid price to be above or below the median price with an accuracy of 88% and 90% respectively.

Accuracy of prediction of Neural Net model -

![Graphical user interface, text

Description automatically generated]()

Accuracy of prediction of SVM model –

![Graphical user interface, text

Description automatically generated]()

## 6.3 Report on Design v2

Describe the design that you actually used. Indicate how, where, and why it differed from your planned design (we expect that these evolved over time). Describe the advantages and shortcomings of your design. Include at least one diagram to refer to in your narrative. Underline edited sentences and additions from v1, if any.

Bid Estimate will be using Neural Nets for Sales Price prediction. The estimation process will involve the following components –

1. Housing Data – The csv file containing the housing data.
2. Data cleanup and pre-processing – This step will involve using Tensor Flow Transform to normalize the input features, along with converting stings to integers.
3. Forward Feed Neural Net – The actual prediction process, with the normalized input data to determine the Sales Price. The Neural Net consists of the following –

* Hidden layer 1: 32 neurons, ReLU activation
* Hidden layer 2: 32 neurons, ReLU activation
* Output Layer: 1 neuron, Sigmoid activation

The reason for using 2 hidden layers, with 32 neurons each, has been chosen based on credible accuracy of 88% reported on similar experiments using this neural net architecture. [4]

Out of 79 features available in the dataset, 10 features were selected which impact the real estate pricing. As cost of land is part of the total cost of the property, hence a larger sq ft. of a property means higher property value. Similarly, houses with more bed rooms, tend to have higher prices as compared to houses with lesser bed rooms.

As a future work, it would be nice to apply feature selection methods using Sklearn’s feature selection library.

Predict the price of the property to be below/above median price

Feed Train data into Neural Net

Out of 79 features, 10 features were selected for price prediction

Split data randomly into 70% train data, 15% validation data and 15% test data

Predict quote to be below/above list price

Data Cleanup and Pre processing, Feature Selection

Forward feed Neural Net

Housing Data

Validate model with 15% validation data, to tune hyper parameters (# of hidden layers, # of nodes in hidden layers, etc.)

Predict the bid price for test data set and compute accuracy

**Forward Feed Neural Net Design** (picture [4])

Diagram, schematic

Description automatically generated

The initial prosed design was followed as per discussed in Assignment 4. The goal at this time was to ensure that a working Neural Net model is implemented for the housing data set, which would now allow for experiments. As I understand, there are a lot of hyper parameters which can be modified for the current implementation, based on which the accuracy of the model would be impacted.

Advantages of this design –

* The design is simple and easy to understand. My clear purpose was to create a proof of concept that a regression problem can be solved using Neural Nets. The choice of 2 hidden layers was made for experimentation.
* The design meets the goal which was set out for the task. It is easy to comprehend for me at this time of my understanding on Neural Nets and Machine Learning.

Shortcomings of this design –

* This design takes an intuitive approach over the Feature Selection. Currently features are chosen based on my understanding of house pricing. The selected features are known to contribute to the value of a property. As an improvement to this, various automated feature selection techniques like sklearn’s SelectKBest or f\_regression can be used for feature selection.
* The output of this model is not intelligible. This is a common issue with all Neural Net architectures as the output of these architectures are hard to explain.

Technologies – Neural Net and SVM

For this design Neural Net and Support Vector Machines were used for the house price prediction. Neural Nets have been more accurate (~85%) as compared to a Polynomial Kernel used for SVM which whose accuracy was ~70%.

## 6.4 Tools v2

Describe the tool(s) that you are using. Describe their advantages and their shortcomings. Underline edited sentences and additions from v1, if any.

## Keras from Tensor Flow Library was used for Neural Net modelling for this project. Pandas and Numpy will be used as data structures of storing in-memory data set required for processing. SciKit Learn’s pre-processing and model selection was used for scaling the data and splitting the data into training, validation and testing datasets. Matplotlib was used for plotting the accuracy of training and validation models.

## For Pre-processing, SciKit Learn’s Preprocessing library was used –

* Normalize the feature values between 0 and 1 using SciKit Preprocessing MinMaxScaler.

As part of the original plan, Tensor Flow’s pre processing was planned to be used, but the features with string values were less important as compared to the 10 selected features (as mentioned above), hence the below items was not required for this project -

## Convert strings to integers by generating a vocabulary over all of the input values

## Fill missing data with missing values from other data points

Data was split into 70% training data, 15% validation data and remaining 15% was treated as test data for evaluating the accuracy of the model.

## For the Neural Net implementation, Keras was used. A Sequential Keras model was implemented using Dense Layers. Each dense layer will consist of 32 tensors each with ReLU activation functions. The output tensor will use sigmoid activation function.

For optimization of the model, we will use Gradient Descent (sgd optimizer) and use cross entropy loss function. The optimizer will track the accuracy during the training phase of the model.

## Input vectors and initial weight and bias vectors will be setup to initialize the NN. Based on the error calculated using MSE, the weights will be adjusted to reduce the error, and the process was repeated 100 times reduce the error. During each iteration, the weights of the neural nets were adjusted.

## Once the model was trained, it was run against the validation set, to understand model’s effectiveness. Based on the execution on the validation set, 2 hidden layers were determined to be giving best results as compared to more/less hidden layers. Dense layers were added and evaluated, with same batch size and epochs settings to come up with the final architecture.

During the validation phase, drop outs were also tried on the hidden layers to observe improvement in model performance. But negligible difference in terms of accuracy was observed when drop outs were added. A drop out rate of 0.2 and 0.5 were tried for this project.

## Below are couple of samples where Neural Network has been used with considerable accuracy for House Price prediction –

## a. Singapore House price prediction, with a MSE of ~3.8 for a 6-8-1 architecture.[2]

## b. Kaggle-kc house Dataset using NN, where accuracy of NN was 81% with respect to accuracy of 69% or MLR [3]

For SVM, SciKit Learn’s SVM implementation was used. A 3rd degree Polynomial kernel was chosen. The steps mentioned above for pre processing was used as is for SVM implementation.

For hyperparameter tuning, following approaches were taken –

1. Using RBF kernel – The model quality degraded with this kernel. This kernel was used with alternately changing the gamma, and C. Third degree polynomial hyper plane was giving the best results.
2. Using higher degrees for polynomial kernel – Degrees 4 and 5 gave similar accuracies as compared to 3. Also, degree 3 passed the knee test. Higher degrees also, slowed the model performance hence, were not selected for this project.

## 6.5 Contrast between approaches

You were to include two technologies or approaches to your problem, and implement at least one. Contrast the two technologies (not the tools—those are described above) as they specifically relate to your project.

Neural Nets are modelled based on the way human brain’s neurons learn facts from data. This approach to learning is fundamentally different from Support Vector Machines (SVM), where the approach is to determine a function which divides the data set based on the classification.

The goal for SVM is to find a hyper plane, and data points around the plane (which act as Support vectors), which are at equal distances from the hyper plane, but at opposite sides of the hyper plane.

Neural Nets require large data sets for training. This is not so true for SVM.

Both these models allow hyper parameter tuning. With Neural Nets, we look for adding additional hidden layers, changing the activation function of the hidden layers, introducing drop outs to the architecture to avoid overfitting.

In case of SVM, model tuning usually involves changing the gamma, C values for RBF kernel or finding the degree of the hyper plane which gives best results.

SVM’s rely on distance calculations between the support vectors and the hyperplane. Neural nets do not have distance measurements as part of the algorithms. A neural net focuses on determining the weights to be assigned using back propagation techniques. The weights assigned determines the impact of the neurons from a previous layer to the current layer. Additionally, applying drop out further randomize the influence of a neuron from a previous layer on the current layer.

## 6.6 What did *not* work well

We want to see that you understand limitations, not just benefits. Explain the most significant aspects of your project that fell short of your plans or desires.

Below are the areas where I would have like to explore more on this project –

1. The initial idea of this project is to find a best bid price (a dollar value), for the quoted sales price of the property. But due to lack of data sets available with both the sales price and bid prices for homes sold within the past years, it became challenging to create a single data set with this correlation.

The current project implementation does a great job in predicting if the property will be sold below or above the median price, which is the closest we could achieve for this term project.

1. While working on the hyperparameter tuning, I was planning to get a deeper understanding on the Drop outs, or effects of increasing dense layers. But for this project, I was not able to see much effects on the neural nets by adding drop outs, or increasing the number of hidden layers.

## 6.7 What *did* work well

In paragraph form, explain the most significant aspects of your project that met or exceeded your plans or desires.

The accuracy of both the models were as per expectation. The accuracy of this project has been defined as follows –

*Given a set of data points (mentioned in Summary section), how accurate is the prediction of the model, to tell us if the house with these data points would be sold above or below list price.*

Below were the accuracy number for –

SMV – 90%, Neural Nets – 88 %

These accuracy numbers give us a solid ground work, which would help home buyers take decisions on whether to quote higher or lower than the list price of the property.

## 6.8 Sample Source

Supply key excerpts from your source code—or what comes closest to “source code.” Limit: 2 pages of 12-point Times New Roman equivalent. Include an explanation of where the excerpts fit in your implementation. These are counted as figures, and do not count towards the total page limit.

Loading dataset, and data pre processing (this step is same for SVM and Neural Nets)

## Text Description automatically generated

SVM model setup, and accuracy measurement –

Text

Description automatically generated

Neural Net architecture –

![Text

Description automatically generated]()

Training the neural Net model and determining accuracy –

Text

Description automatically generated

Graph showing the reduction of loss as compared between training and validation data sets.

A picture containing chart

Description automatically generatedText

Description automatically generated

## 6.9 Source

Refer the reader to your source code (or what comes closest to it) and input where possible.

The source code of this project is attached with the Assignment submission.

Here are the files attached with this document –

Final\_NN.py

import pandas as pd

from sklearn import preprocessing

from sklearn.model\_selection import train\_test\_split

import tensorflow.keras as keras

from tensorflow.keras.models import Sequential

from tensorflow.keras.layers import Dense

import matplotlib.pyplot as plt

df = pd.read\_csv('housepricedata.csv')

dataset = df.values

df.head()

# Feature and classifier

X = dataset[:,0:10]

Y = dataset[:,10]

# Pre processing

min\_max\_scaler = preprocessing.MinMaxScaler()

X\_scale = min\_max\_scaler.fit\_transform(X)

# Split to train-test

X\_train, X\_val\_and\_test, Y\_train, Y\_val\_and\_test = train\_test\_split(

X\_scale, Y, test\_size=0.3)

# Split test further to validation and test

X\_val, X\_test, Y\_val, Y\_test = train\_test\_split(

X\_val\_and\_test, Y\_val\_and\_test, test\_size=0.5)

print(X\_train.shape, X\_val.shape, X\_test.shape, Y\_train.shape,

Y\_val.shape, Y\_test.shape)

# Construct Neural Net Architecture

model = Sequential([

Dense(32, activation='relu', input\_shape=(10,)),

Dense(32, activation='relu'),

Dense(1, activation='sigmoid'),

])

# Compile model (set gradient descent optimizer)

model.compile(optimizer='sgd',

loss='binary\_crossentropy',

metrics=['accuracy'])

# Train the Neural Net

hist = model.fit(X\_train, Y\_train,

batch\_size=32, epochs=100,

validation\_data=(X\_val, Y\_val))

# Evaluate accuracy

model.evaluate(X\_test, Y\_test)[1]

# Visualize the accuracy

plt.plot(hist.history['loss'])

plt.plot(hist.history['val\_loss'])

plt.title('Model loss')

plt.ylabel('Loss')

plt.xlabel('Epoch')

plt.legend(['Train', 'Val'], loc='upper right')

plt.show()

final\_SVM.py

import numpy as np

import matplotlib.pyplot as plt

import pandas as pd

from sklearn import preprocessing

from sklearn.preprocessing import StandardScaler

from sklearn.model\_selection import train\_test\_split

from sklearn.svm import SVR

from sklearn.metrics import r2\_score

import matplotlib.pyplot as plt

import seaborn as sns

import sklearn.svm as svm

# Load data

df = pd.read\_csv('housepricedata.csv')

dataset = df.values

# Pre processing

min\_max\_scaler = preprocessing.MinMaxScaler()

X = dataset[:,0:10]

Y = dataset[:,10]

# Split between train-test

X\_scale = min\_max\_scaler.fit\_transform(X)

X\_train, X\_val\_and\_test, Y\_train, Y\_val\_and\_test = train\_test\_split(X\_scale,

Y, test\_size=0.3)

# Classify above/below median score using

# Polnomial Kernel

svm\_classifier = svm.SVC(kernel='poly', degree= 3)

svm\_classifier.fit(X\_train, Y\_train)

predicted = svm\_classifier.predict(X\_val\_and\_test)

# Accuracy of model

svm\_classifier.score(X\_val\_and\_test, Y\_val\_and\_test)

## 6.10 Presentation

Make a 3 minute video (5 minute maximum if absolutely needed) presentation of your results, including a demonstration, and point your facilitator to the location of the video.

URL to video –

<https://2159741-5.kaf.kaltura.com/media/1_m6k9wvmc>

Presentation in the video is uploaded in Assignments sections.

# Evaluation



## Appendix 1

For voluminous material, as needed—to be read on an as-needed basis only. (References in at least one place within the paper.)

-