Summer Internship Report

Indian Oil Corporation Limited
(Guwahati)

(15th June to 14th July, 2019)

Apartment Price Prediction

Submitted By-

Abdul Mannan (Tezpur University)

Padmanabh Garg (Tezpur University)

Angshulekha Bora (Dayanand Sagar College of Engineering)

PREFACE

The knowledge of any subject is incomplete until it is done practically. Computer Science particularly requires a thorough knowledge of practical training for a comprehensive understanding. The progress is certainly based on the discovery of the new facts.

The knowledge of any subject is incomplete until it is done practically. Computer Science particularly requires a thorough knowledge of practical training for a comprehensive understanding. The progress is certainly based on the discovery of the new facts.

This report describes the work carried out by me during one-month internship at I.O.C.L., Noonmati (Guwahati Refinery). During this period, I have understood a lot of things related to the working of a refinery in its different divisions under Information and Systems Dept. This has developed a sense of confidence in me. This internship is proved to be a good practical experience and has also enhanced my technical knowledge. A lot of credit goes to my instructor who helped me all the way from the very beginning.

Overview Of IOCL, Guwahati

Indian oil and Corporation Limited, is an Indian state-owned oil and corporation with its headquarters in New Delhi, India. The Company is world's 119th largest public corporation, according to Forbes fortune 500 list and the largest public corporation in India when ranked by revenue. It has also earned reputation as **18th largest** petroleum company in the world and No. 1 petroleum trading company among the national oil companies in Asia-pacific region. Indian Oil and its subsidiaries account for a 49% share in the petroleum products market, 31% share in refining capacity and 67% downstream sector pipelines capacity in India. The Indian Oil Group of Companies owns and operates 11 of India's 23 refineries with a combined refining capacity of 80.7 million metric tons per year. In FY 2012 IO CL sold 75.66 million tons of petroleum products and reported a PBT of 37.54 billion, and the Government of India earned an excise duty of 232.53 billion and tax of 10.68 billion. The company is mainly controlled by Government of India which owns approximately 58.57% shares in the company. It is one of the seven Maharatna status companies of India, apart from Coal India Limited, NTPC Limited, Oil and Natural Gas Corporation, Steel Authority of India Limited, Bharat Heavy Electricals Limited and Gas Authority of India Limited.

Indian Oil operates the largest and the widest network of fuel 4 stations in the country, numbering about 20,575 (16,3 50 regular ROs & 4,225 Kisan Seva Kendra). It has also started Auto LPG Dispensing Stations (ALDS). It supplies Indane cooking gas to over 66.8 million households through a network of 5,934 Indane distributors. In addition, Indian Oil's Research and Development Center (R&D) at Faridabad supports, develops and provides the necessary technology solutions to the operating divisions of the corporation and its customers within the country and abroad.

Guwahati Refinery is the country's first Public Sector Refinery as well as Indian Oil's first Refinery serving the nation since 1962. It is known as GONGOTRI of Indian Oil. Built with Rumanian assistance, the initial crude processing capacity at the time of commissioning of this Refinery was 0.75 MMTPA and the Refinery was designed to process a mix of OIL and ONGC crude. The refining capacity was subsequently enhanced to 1.0 MMTPA and with INDMAX, the pilot plant for first in-house technology of Indian Oil, the ISOSIV and Hydrotreater the Refinery has been able to produce eco-friendly fuels. The Refinery produces various products and supplies them to North eastern India as well as beyond, up to Siliguri end through the Guwahati-Siliguri Pipeline, spanning 435 KM, which was the first Pipeline of Indian Oil and commissioned in 1964. Most of the products of Guwahati Refinery are evacuated through pipeline 5 and some quantity also through road transportation.

Quality LPG, Motor Spirit, Aviation Turbine Fuel, Superior Kerosene Oil, High Speed Diesel, Light Diesel Oil and Raw Petroleum Coke are the products of this Refinery.

In line with Indian Oil's responsibility towards the society, Guwahati Refinery has contributed yeomen service towards developing the community, which exists around it. The CSR agenda of the Refinery focuses on three broad areas of education, health care and providing water supply. Initiatives taken under these heads are participative in nature with community participation in a partnership model for ensuring sustainable development of the community.

Guwahati Refinery is also known for its sincere efforts on development as well as implementation of effective Safety, Health & Environment management systems and procedures along with good performance in occupational health and safety.

CERTIFICATE

This report on **apartment and flat price prediction** has been submitted by the group consisting of **Angshulekha Bora** (Dayananda Sagar College of Engineering), **Abdul Mannan** (Tezpur University) and **Padmanabh Garg** (Tezpur University), 7th semester. The research for the same has been conducted under my supervision. Their work is satisfactory and is accepted. I hereby do declare that their work is fit to be presented to the Internship Committee for evaluation.

Certified By:

Mr. Manoj M. Parhate

DGM(IS)

Internship Supervisor

ACKNOWLEDGEMENT

The satisfaction that accompanies the successful completion of this project would be incomplete without mentioning the name of the people who made it possible, without whose constant guidance and encouragement would have made our efforts go in vain. We consider ourselves privileged to be able to express our gratitude and respect towards all those who guided us throughout the project. We convey our gratitude to our project guide Mr. Manoj M. Parhate, DGM(IS) and learning and development Centre Guwahati Refinery for providing encouragement, enormous support and guidance which was of a great help to complete this project successfully. We owe our wholehearted thanks to our respective colleges for providing us with the opportunity. Last, but not the least, we wish to thank our parents for their constant support and faith in us.

Introduction

Machine learning is an application of artificial intelligence (AI) that provides systems the ability to automatically learn and improve from experience without being explicitly programmed. **Machine learning focuses on the development of computer programs** that can access data and use it to learn for themselves.

The process of learning begins with observations or data, such as examples, direct experience, or instruction, in order to look for patterns in data and make better decisions in the future based on the examples that we provide. **The primary aim is to allow the computers learn automatically** without human intervention or assistance and adjust actions accordingly.

Machine learning algorithms are often categorized as supervised or unsupervised.

- **Supervised machine learning algorithms** can apply what has been learned in the past to new data using labeled examples to predict future events. Starting from the analysis of a known training dataset, the learning algorithm produces an inferred function to make predictions about the output values. The system is able to provide targets for any new input after sufficient training. The learning algorithm can also compare its output with the correct, intended output and find errors in order to modify the model accordingly.

- In contrast, **unsupervised machine learning algorithms** are used when the information used to train is neither classified nor labeled. Unsupervised learning studies how systems can infer a function to describe a hidden structure from unlabeled data. The system doesn't figure out the right output, but it explores the data and can draw inferences from datasets to describe hidden structures from unlabeled data.
- **Semi-supervised machine learning algorithms** fall somewhere in between supervised and unsupervised learning, since they use both labeled and unlabeled data for training – typically a small amount of labeled data and a large amount of unlabeled data. The systems that use this method are able to considerably improve learning accuracy. Usually, semi-supervised learning is chosen when the acquired labeled data requires skilled and relevant resources in order to train it / learn from it. Otherwise, acquiring unlabeled data generally doesn't require additional resources.

- **Reinforcement machine learning algorithms** is a learning method that interacts with its environment by producing actions and discovers errors or rewards. Trial and error search and delayed reward are the most relevant characteristics of reinforcement learning. This method allows machines and software agents to automatically determine the ideal behavior within a specific context in order to maximize its performance. Simple reward feedback is required for the agent to learn which action is best; this is known as the reinforcement signal.

  Machine learning enables analysis of massive quantities of data. While it generally delivers faster, more accurate results in order to identify profitable opportunities or dangerous risks, it may also require additional time and resources to train it properly. Combining machine learning with AI and cognitive technologies can make it even more effective in processing large volumes of information.

Problem Description

This project is going to be focusing on solving the problem of predicting house prices for house buyers and house sellers, and making it convenient for them to take rational decisions.

A house value is simply more than location and square footage. Like the features that make up a person, an educated party would want to know all aspects that give a house its value.

We are going to take advantage of all of the feature variables available to use and use it to analyze and predict house prices.

We are going to break everything into logical steps that allow us to ensure the cleanest, most realistic data for our model to make accurate predictions from:

1. Load Data and Packages
2. Analyzing the Test Variable (Sale Price)
3. Multivariable Analysis
4. Impute Missing Data and Clean Data
5. Feature Transformation/Engineering
6. Modeling and Predictions

Here we will be making the use of linear regression algorithm in predicting the prices.

Packages used

1. Pandas
2. NumPy
3. Warnings
4. XGBoost
5. SciPy
6. Sklearn
7. Collections
8. Pickle
9. Joblib

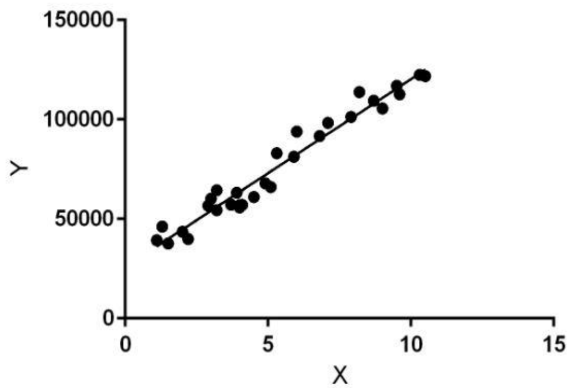
Algorithm used in the project

Linear Regression is a machine learning algorithm, which is based on **supervised learning**. It performs a **regression task**. Regression models a target prediction value based on independent variables. It is mostly used for finding out the relationship between variables and forecasting. Different regression models differ based on – the kind of relationship between dependent and independent variables, they are considering and the number of independent variables being used.

According to this definition, a house's price depends on parameters such as the number of bedrooms, living area, location, etc. If we apply artificial learning to these parameters we can calculate house valuations in a given geographical area.

The idea of regression is pretty simple: given enough data, you can observe the relationship between your target parameter (the output) and other parameters (the input), and then apply this relationship function to real observed data.

To show you how regression algorithm works we'll consider only one parameter – a home's living area – to predict price. It's logical to suppose that there is a linear relationship between area and price.



Linear regression performs the task to predict a dependent variable value (y) based on a given independent variable (x). So, this regression technique finds out a linear relationship between x (input) and y(output). Hence, the name is Linear Regression.

Hypothesis function for Linear Regression:

$$y = \theta_1 + \theta_2.x$$

While training the model we are given:

x: input training data (univariate – one input variable(parameter))

y: labels to data (supervised learning)

When training the model – it fits the best line to predict the value of y for a given value of x. The model gets the best regression fit line by finding the best θ1 and θ2 values.

θ1: intercept

θ2: coefficient of x

Once we find the best $\theta 1$ and $\theta 2$ values, we get the best fit line. So, when we are finally using our model for prediction, it will predict the value of y for the input value of x.

How machine learning can be incorporated in apartment price prediction:

As in the coming days population is increasing very drastically we need to develop some effective strategies to stabilize the residency conditions of the people. Also, it is very essential for buyers and sellers to mutually agree on a price so that the seller cannot take undue advantage from the buyer. So we have come up with one possible solution using regression algorithm in machine learning which takes the various characteristics of the apartments into account and using the training set provided, our project predicts the estimated price based on the dataset. The regression algorithm enables us to predict the undefined values present in the test data set. So, when the user gives the input about the various specifications about the apartment, a comparison is being made in the training set, the flat ID which matches, the corresponding sale price of the flat is printed.

Advantages of using Machine learning in the project:

1. Easily identifies trends and patterns

Machine Learning can review large volumes of data and discover specific trends and patterns that would not be apparent to humans. For instance, here in apartment price prediction, machine learning serves to understand the pattern in which the sale price is designated to be for specific aspects of the apartments and help predict the sale price of other similar apartments based on the pattern recognized which is untraceable by humans.

2. No human intervention needed (automation)

With ML, you don't need to babysit your project every step of the way. Since it means giving machines the ability to learn, it lets them make predictions and also improve the algorithms on their own. For instance, in our project, from the given sale price in the data set our algorithm determines the sale price of other apartments by learning from the present dataset and prints the predicted sale price.

3. Handling multidimensional and multi-variety data

Machine Learning algorithms are good at handling data that are multidimensional and multi-variety, and they can do this in dynamic or uncertain environments, as our data set has variable kind of data, our algorithm parses the multidimensional dataset in a very good manner and predicts the sale price accurately.

4. Wide Applications

You could be an e-trailer or a healthcare provider and make ML work for you. Where it does apply, it holds the capability to help deliver a much more personal experience to customers while also targeting the right customers.

Disadvantages of using Machine Learning in the project:

With all those advantages to its powerfulness and popularity, Machine Learning in predicting the sale price of apartments isn't perfect. The following factors serve to limit it:

1. Data Acquisition

Machine Learning requires massive data sets to train on, and these should be inclusive/unbiased, and of good quality. There can also be times when there is no information about some brand-new characteristic of an apartment which is just launched so the algorithm cannot predict the price of such apartment due to the unavailability of similar dataset.

2. Time and Resources

ML needs enough time to let the algorithms learn and develop enough to fulfill their purpose with a considerable amount of accuracy and relevancy. It also needs massive dataset to train itself and exhibit its functionality in prediction. This can mean additional requirements of computer power for you.

3. Interpretation of Results

Another major challenge is the ability to accurately interpret results generated by the algorithms. As it predicts the results, it is not that the predicted result is always cent percent correct. You must also carefully choose the algorithms for your purpose.

4. **High error-susceptibility**

   Machine Learning is autonomous but highly susceptible to errors. Suppose you train an algorithm with data sets small enough to not be inclusive. You end up with biased predictions coming from a biased training set. This leads to irrelevant predicted price being displayed to customers. In the case of ML, such blunders can set off a chain of errors that can go undetected for long periods of time. And when they do get noticed, it takes quite some time to recognize the source of the issue, and even longer to correct it.

Future Applications

Machine Learning is currently one of the hottest topics in IT. Technologies such as digital, big data, Artificial Intelligence, automation and machine learning are increasingly shaping future of work and jobs. is a specific set of techniques that enable machines to learn from data, and make predictions? When the biases of our past and present fuel the predictions of the future, it's a tall order to expect AI to operate independently of human flaws.

Machine Learning Code

Program.py

```
# coding: utf-8

# In[1]:

import pandas as pd

import numpy as np

import warnings

import xgboost as xgb

from scipy.stats import skew

from scipy import stats

from scipy.stats.stats import pearsonr

from sklearn.linear_model import ElasticNet, Lasso,  BayesianRidge,
LassoLarsIC
```

```
from sklearn.ensemble import RandomForestRegressor,
GradientBoostingRegressor

from sklearn.kernel_ridge import KernelRidge

from sklearn.pipeline import make_pipeline

from sklearn.preprocessing import RobustScaler

from sklearn.base import BaseEstimator, TransformerMixin,
RegressorMixin, clone

from sklearn.model_selection import KFold, cross_val_score,
train_test_split

from sklearn.metrics import mean_squared_error

from scipy.stats import norm

from scipy.special import boxcox1p

from collections import Counter

from sklearn.preprocessing import LabelEncoder, OneHotEncoder

import pickle

import joblib

from sklearn.linear_model import LinearRegression,LassoCV, Ridge,
LassoLarsCV,ElasticNetCV

from sklearn.model_selection import GridSearchCV, cross_val_score,
learning_curve

from sklearn.ensemble import RandomForestRegressor, AdaBoostRegressor,
ExtraTreesRegressor, GradientBoostingRegressor

from sklearn.preprocessing import StandardScaler, Normalizer,
RobustScaler

warnings.filterwarnings('ignore')
```

```
# In[2]:

# Load train and Test set

train = pd.read_csv("./Connect/train.csv")

test = pd.read_csv("./Connect/test.csv")


# In[3]:

# Save the 'Id' column

train_ID = train['Id']

test_ID = test['Id']

# Now drop the 'Id' column since it's unnecessary for the prediction
process.

train.drop("Id", axis = 1, inplace = True)

test.drop("Id", axis = 1, inplace = True)


# In[4]:

train.head()


# In[5]:

test.head()


# In[6]:

# Checking Categorical Data

train.select_dtypes(include=['object']).columns
```

```
# In[7]:

# Checking Numerical Data

train.select_dtypes(include=['int64','float64']).columns


# In[8]:

cat = len(train.select_dtypes(include=['object']).columns)

num = len(train.select_dtypes(include=['int64','float64']).columns)


# In[9]:

# Correlation Matrix Heatmap

corrmat = train.corr()


# In[10]:

# Top 10 Heatmap

k = 10 #number of variables for heatmap

cols = corrmat.nlargest(k, 'SalePrice')['SalePrice'].index

cm = np.corrcoef(train[cols].values.T)


# In[11]:

most_corr = pd.DataFrame(cols)

most_corr.columns = ['Most Correlated Features']

most_corr
```

```
# In[12]:

# Combining Datasets

ntrain = train.shape[0]

ntest = test.shape[0]

y_train = train.SalePrice.values

all_data = pd.concat((train, test)).reset_index(drop=True)

all_data.drop(['SalePrice'], axis=1, inplace=True)


# In[13]:

# Find Missing Ratio of Dataset

all_data_na = (all_data.isnull().sum() / len(all_data)) * 100

all_data_na = all_data_na.drop(all_data_na[all_data_na ==
0].index).sort_values(ascending=False)[:30]

missing_data = pd.DataFrame({'Missing Ratio' :all_data_na})

missing_data


# In[14]:

all_data["PoolQC"] = all_data["PoolQC"].fillna("None")

all_data['GarageType'] = all_data['GarageType'].fillna('None')

all_data = all_data.drop(['Utilities'], axis=1)

all_data['NoF'] = all_data['NoF'].fillna("None")
```

```
# In[15]:

# Check if there are any missing values left

all_data_na = (all_data.isnull().sum() / len(all_data)) * 100

all_data_na = all_data_na.drop(all_data_na[all_data_na ==
0].index).sort_values(ascending=False)

missing_data = pd.DataFrame({'Missing Ratio' :all_data_na})

missing_data.head()


# In[16]:

all_data['NoF'].describe()


# In[17]:

all_data['NoF'] = all_data['NoF'].apply(str)

#Changing OverallCond into a categorical variable

all_data['OverallCond'] = all_data['OverallCond'].astype(str)


# In[18]:

cols = ['NoF', 'OverallCond', 'PoolQC', 'Street']

# Process columns and apply LabelEncoder to categorical features

lbl = LabelEncoder()

lbl.fit([y for x in all_data[cols].get_values() for y in x])

# Saving the label encoder to pickle file

output = open('Encoder.pkl', 'wb')

pickle.dump(lbl, output)
```

```
output.close()

all_data[cols] = all_data[cols].apply(lbl.transform)



# In[19]:

# We use the NumPy function log1p which applies log(1+x) to all
elements of the column

train["SalePrice"] = np.log1p(train["SalePrice"])

# Get the fitted parameters used by the function

(mu, sigma) = norm.fit(train['SalePrice'])

y_train = train.SalePrice.values



# In[20]:

numeric_feats = all_data.dtypes[all_data.dtypes != "object"].index

# Check the skew of all numerical features

skewed_feats = all_data[numeric_feats].apply(lambda x:
skew(x.dropna())).sort_values(ascending=False)

skewness = pd.DataFrame({'Skewed Features' :skewed_feats})

skewness.head()



# In[21]:

skewness = skewness[abs(skewness) > 0.75]

skewed_features = skewness.index

lam = 0.15

for feat in skewed_features:
```

```
    all_data[feat] = boxcox1p(all_data[feat], lam)
    all_data[feat] += 1
```

 # In[22]:

```
labelencoder_dict = {}
onehotencoder_dict = {}
all_data_train = None
all_data_array = all_data.values
for i in range(0, all_data_array.shape[1]):
    if i in [1,2,3,4,5]:
        label_encoder = LabelEncoder()
        labelencoder_dict[i] = label_encoder
        feature = label_encoder.fit_transform(all_data_array[:,i])
        feature = feature.reshape(all_data_array.shape[0], 1)
        onehot_encoder = OneHotEncoder(sparse=False)
        feature = onehot_encoder.fit_transform(feature)
        onehotencoder_dict[i] = onehot_encoder
    else:
     feature = all_data_array[:,i].reshape(all_data_array.shape[0], 1)
    if all_data_train is None:
        all_data_train = feature
    else:
```

```
        all_data_train = np.concatenate((all_data_train, feature),
         axis=1)
joblib.dump(labelencoder_dict, 'labelencoder_dict.joblib')
joblib.dump(onehotencoder_dict, 'onehotencoder_dict.joblib')
```

 # In[23]:

```
train = all_data_train[:ntrain]
test = all_data_train[ntrain:]
```

 # In[24]:

 # Cross-validation with k-folds

```
n_folds = 5
def rmsle_cv(model):
    kf = KFold(n_folds, shuffle=True,
random_state=42).get_n_splits(train)
    rmse= np.sqrt(-cross_val_score(model, train, y_train,
scoring="neg_mean_squared_error", cv = kf))
    return(rmse)
```

 # In[25]:

```
model_xgb = xgb.XGBRegressor(colsample_bytree=0.2, gamma=0.0,
                             learning_rate=0.05, max_depth=6,
                             min_child_weight=1.5, n_estimators=7200,
                             reg_alpha=0.9, reg_lambda=0.6,
```

```
                                        subsample=0.2,seed=42, silent=1,
                                        random_state =7)
```

 # In[26]:

```
score = rmsle_cv(model_xgb)
```

 # In[27]:

```
def rmsle(y, y_pred):
    return np.sqrt(mean_squared_error(y, y_pred))
```

 # In[28]:

```
model_xgb.fit(train, y_train)
joblib.dump(model_xgb, 'xgboost_model.joblib')
xgb_train_pred = model_xgb.predict(train)
xgb_pred = np.expm1(model_xgb.predict(test))
```

 # In[29]:

```
print('RMSLE score on train data:')
print(rmsle(y_train, xgb_train_pred*0.10 ))
```

 # In[30]:

#Example

```
XGBoost = 1/(0.1177)
```

 # In[31]:

```
ensemble = xgb_pred*XGBoost
```

 # In[32]:

```
sub = pd.DataFrame()

sub['Id'] = test_ID

sub['SalePrice'] = ensemble

sub.to_csv('submission.csv',index=False)

print("The Sale Price for the test.csv file is updated successfully
and stored in submission.csv file\n")
```

App.py

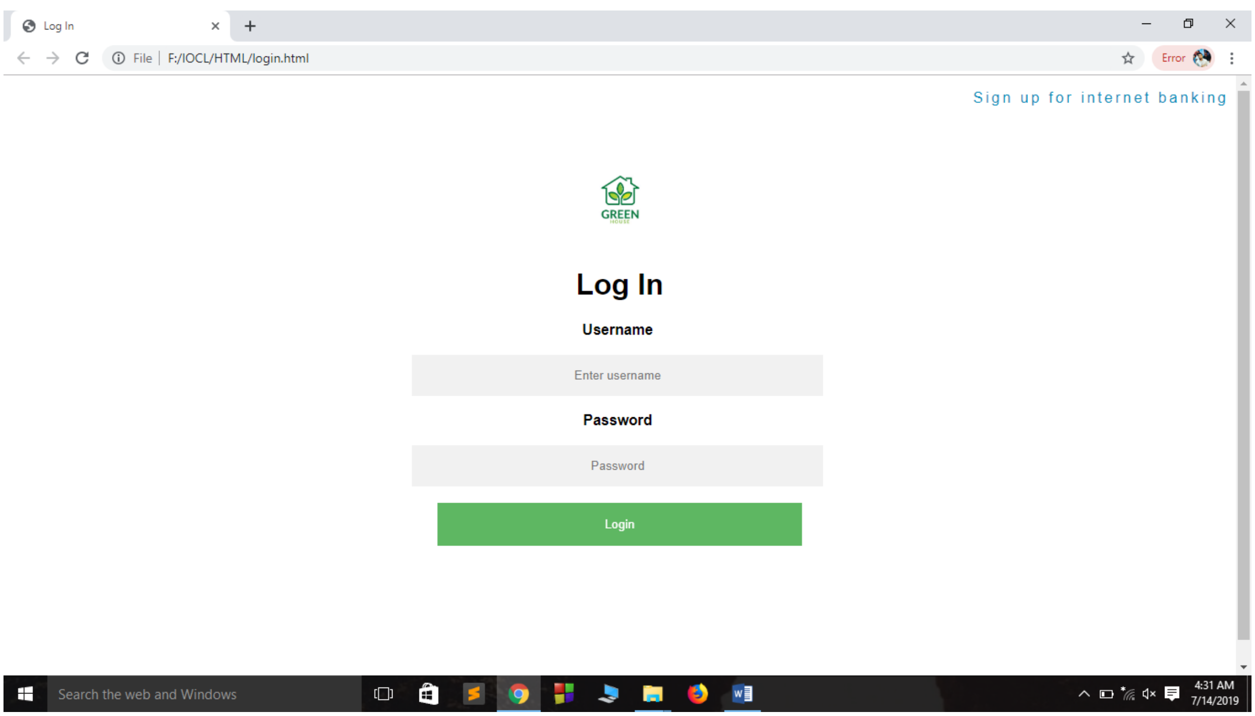
```
from flask import Flask, render_template, request

from scipy.special import boxcox1p

import numpy as np

import pickle

import joblib

app = Flask(__name__)

@app.route('/')

def index():

      return render_template('Details.html')

@app.route('/hello', methods=['POST'])

def hello():

      story = str(request.form['story'])
```

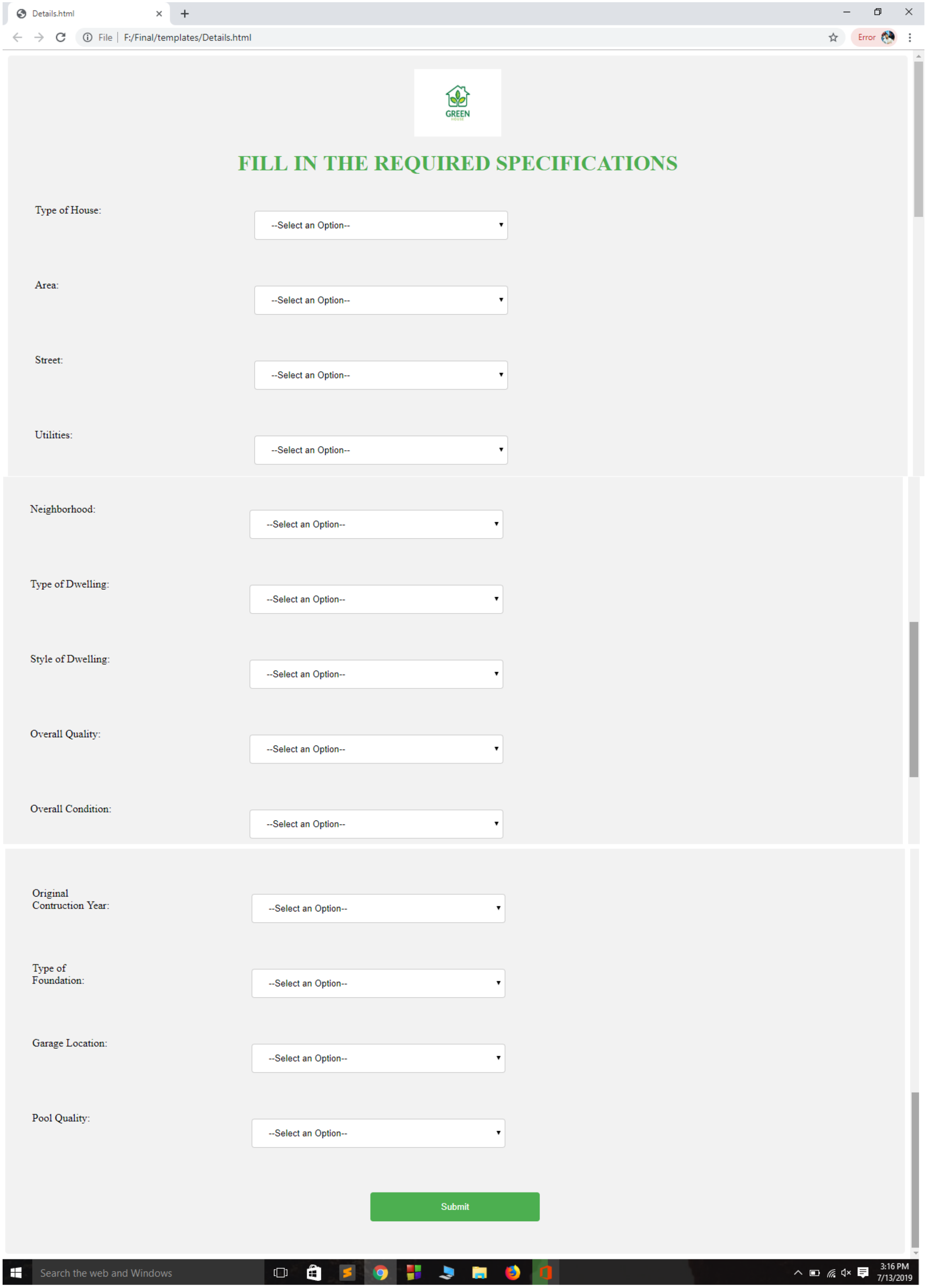
```
area = request.form['area']

street = request.form['street']

utilities = request.form['utilities']

neighbor = request.form['neighbor']

bldgtype = request.form['bldgtype']

housestyle = request.form['housestyle']

quality = request.form['quality']

condition = str(request.form['condition'])

year = request.form['year']

foundation = request.form['foundation']

garage = request.form['garage']

pool = request.form['pool']


pkl_file = open('Encoder.pkl', 'rb')

lbl = pickle.load(pkl_file)

pkl_file.close()


test = [street,condition,pool,street]

print(test)

x = lbl.transform(test)

street = x[0]

condition = x[1]

pool = x[2]
```

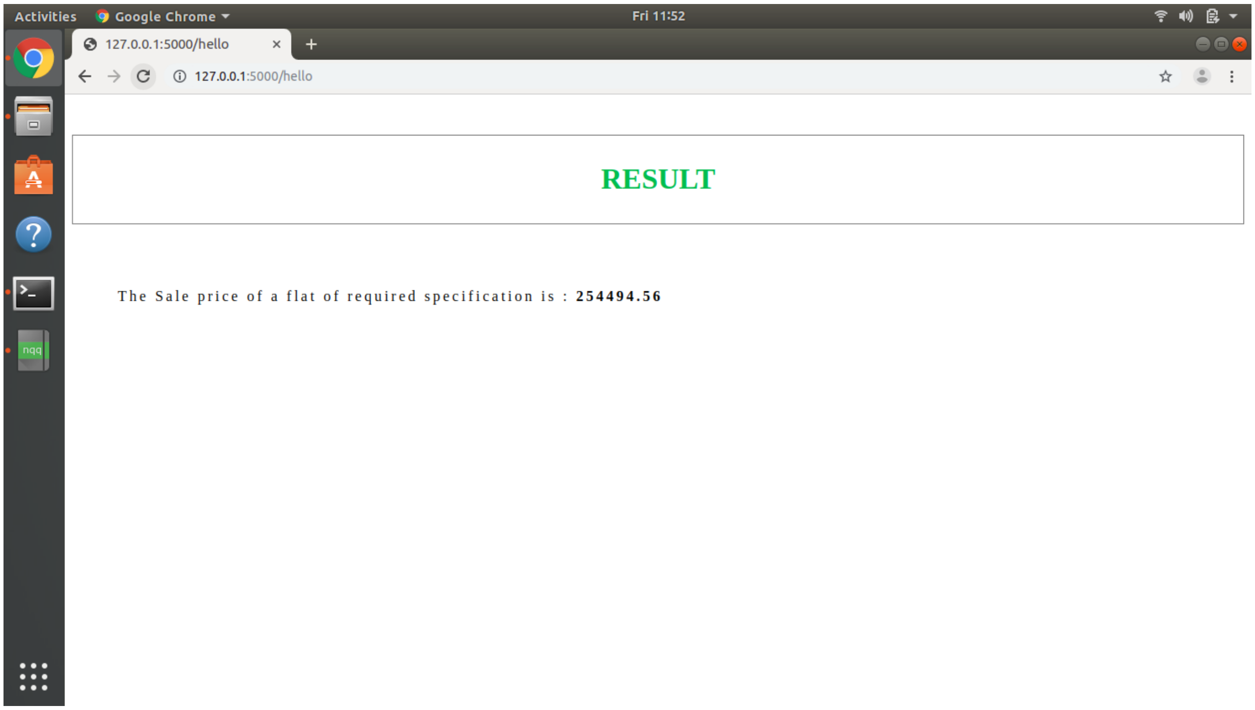
```
street = x[3]

area = boxcox1p(float(area), 0.15) + 1

story = boxcox1p(float(story), 0.15) + 1

condition = boxcox1p(float(condition), 0.15) + 1

quality = boxcox1p(float(quality), 0.15) + 1

year = boxcox1p(float(year), 0.15) + 1

test_data = np.asarray ([[area,bldgtype,foundation,garage,

      housestyle,neighbor,story,condition,quality,

      pool,street,year]])

labelencoder_dict = joblib.load('labelencoder_dict.joblib')

onehotencoder_dict = joblib.load('onehotencoder_dict.joblib')

model = joblib.load('xgboost_model.joblib')

encoded_data = None

for i in range(0,test_data.shape[1]):

      if i in [1,2,3,4,5]:

            label_encoder =  labelencoder_dict[i]

            feature = label_encoder.transform(test_data[:,i])

            feature = feature.reshape(test_data.shape[0], 1)

            onehot_encoder = onehotencoder_dict[i]

            feature = onehot_encoder.transform(feature)

      else:

        feature = test_data[:,i].reshape(test_data.shape[0], 1)

      if encoded_data is None:
```

```
                encoded_data = feature
            else:
                encoded_data=np.concatenate((encoded_data,feature,axis=1)
        price = np.expm1(model.predict(encoded_data))
        print(price[0]) #This is your answer
        variable = price[0];
        return render_template("result.html", result = variable)
if ("__name__ == '__main__"):
        app.run(debug = True)
```

Screenshots




Details.html
File | F:/Final/templates/Details.html
Error
GREEN
FILL IN THE REQUIRED SPECIFICATIONS
Type of House:
--Select an Option--
Area:
--Select an Option--
Street:
--Select an Option--
Utilities:
--Select an Option--
Neighborhood:
--Select an Option--
Type of Dwelling:
--Select an Option--
Style of Dwelling:
--Select an Option--
Overall Quality:
--Select an Option--
Overall Condition:
--Select an Option--
Original Contruction Year:
--Select an Option--
Type of Foundation:
--Select an Option--
Garage Location:
--Select an Option--
Pool Quality:
--Select an Option--
Submit
Search the web and Windows
3:16 PM
7/13/2019



Conclusions

Through the project that we were assigned, we gained good hands on experience from scratch. Dealing with bugs and debugging them were extensive and time consuming but at the end of the day, the experience that we gained from it is invaluable. Working as team has helped us in learning the importance of efficient team work and how working together with good synchronization can make complicated tasks seem easier. At last but not the least, we would like to thank our course instructor, Mr. Manoj M, Parhate for giving us this opportunity to work on such an exciting assignment as this and helping us in the process to gain important skills that will prove to be of great use in future prospects.