Natural Gas Price Prediction

**Using Linear Regression Algorithm**

**Developed by:**AnuraagChelika,Shriya Madishetti,Yamini Lagishetti,Vaishnavi Mariyala

**Smart Bridge-Remote Summer Internship Program**

1. **INTRODUCTION**

The world total primary energy supply (TPES) by fuel in 2016 was as follows: oil (31.9%), coal (27.1%), natural gas (22.1%), biofuels and waste (9.8%), nuclear (4.9%), hydro (1.8%), and other (0.1%) [1]. Natural gas thus has the third-largest share among the TPES. Furthermore, natural gas production continues to grow at a higher pace, most notably with a 3.6% increase in 2017 compared to 2016 that constitutes the largest increase since 2010. In today’s world, concerns about air quality and climate change are growing, but renewable energy is expanding at a limited rate and low-carbon energy sources are hard to find in some areas. Natural gas offers many potential benefits as a solution to environmental problems. Natural gas generates heat, power, and mobility with fewer emissions, including both carbon-dioxide (CO2) emissions and air pollutants, than the other fossil fuels, helping to address widespread concerns over air quality. Because the natural gas energy causes less pollution to the environment than other kinds of energy resource, it has received much more recognition recently. Natural gas exploitation has significantly helped many countries to reduce CO2 emissions nationally and globally since 2014 [2]. Natural gas, which is one of the most important energy resources, is going to play an expanded role in the future of global energy due to its significant environmental benefits. Forecasting natural gas prices is a powerful and essential tool which has become more important for different stakeholders in the natural gas market, allowing them to make better decisions for managing the potential risk, reducing the gap between the demand and supply, and optimizing the usage of resources based on accurate predictions. Accurate natural gas price forecasting not only provides an important guide for effective implementation of energy policy and planning, but also is extremely significant in economic planning, energy investment, and environmental conservation. Therefore, researchers continue to study natural gas price forecasting models with great interest, with the aim of making predictions as accurate as possible in future. There are plenty of methods for analyzing and forecasting natural gas prices and machine learning is increasingly used. Machine learning algorithms can learn from historical relationships and trends in the data and make data-driven predictions or decisions.

1.1 **Overview**

Natural gas has been proposed as a solution to increase the security of energy supply and reduce environmental pollution around the world. Being able to forecast natural gas price benefits various stakeholders and has become a very valuable tool for all market participants in competitive natural gas markets. Machine learning algorithms have gradually become popular tools for natural gas price forecasting. In this paper, we investigate data-driven predictive models for natural gas price forecasting

based on common machine learning tools, which utilize machine learning approaches including support vector machines (SVM),Logistic Regression etc. We harness the method of cross-validation for model training and monthly natural gas spot price data from January 1977 to August 2020 evaluation. Results show that these two machine learning methods have different performance in predicting natural gas prices. However, overall ,linear regression algorithm prediction performance compared with SVM,linear Regression was selected as best algorithm based on accuracy

**1.2 Purpose**

In this project, we make use of pandas, numpy , matplotlib, and seaborn libraries using the open-source, object-oriented programming language python 3.0 and packages such as scikit-learn to predict the price of natural gas

And in the end, to predict natural gas price on particular day or not using the predictions from multiple machine learning algorithms and withdrawing the conclusions

2. **LITERATURE SURVEY**

Data mining is the process of analyzing data from different perspectives and extracting useful knowledge from it. It is the core of knowledge discovery process. The various steps involved in extracting knowledge from raw data. Different data mining techniques include classification, clustering, association rule mining, prediction and sequential patterns, neural networks, regression etc. Regression is a data mining technique used to predict a range of numeric values (also called continuous values ), given a particular dataset. For example, regression might be used to predict the future actions. Natural gas price prediction is particularly well suited to Regression Algorithm. In Regression , the independent variables are well trained to predict the dependent variable. And the model is evaluated.

**2.1 Existing Problem**

The previous models have less accuracy and the predictions are not accuarate whereas this model is constrained with the lot of advantages and with a higher accuracy than any other model already proposed. In this model we used Machine learning algorithm named Linear Regression which give an accuracy 30% of the predicted problem and there is an user friendly user interface to check the price of natural will help various stakeholders for natural gas production

**2.2 Proposed Solution**

**Machine Learning ( Linear Regression):**

Linear Regression is one of the simplest and most common supervised machine learning algorithms that data scientists use for predictive modeling. We’ll use Linear regression to build a model that predicts the natural gas price. And also we have created an UI using the Flask for the natural gas price prediction and this UI will allow the users to predict their price very easily and the User interface is user friendly not at least one complication in using the interface, and it can be used just by entering some necessary details into the UI in real time it'll give the predicted value like the price of natural gas on that particular day

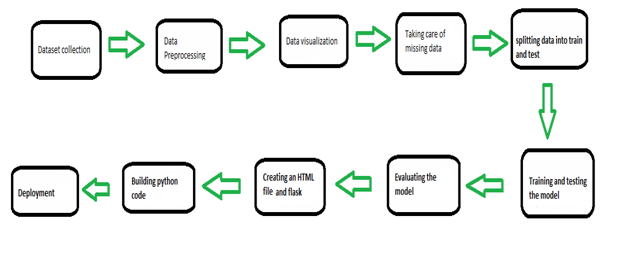
**3. THEORETICAL ANALYSIS**

While selecting the algorithm that gives an accurate prediction we gone through lot of algorithms which gives the results abruptly accurate and from them we selected only one algorithm for the prediction problem that is Linear Regression , it gives the output based on the independent variables accurately.

The peculiarity of this problem is collecting the natural gas price real time and working with the prediction at the same time, so we developed an user interface for different stakeholders who'll be accessing for the natural gas price prediction. There are several ways to check the Linear Regression model accuracy. Usually we use Root Mean Squared Error.We train Linear regression model by adding or removing the features to dataset, and see which one has the lowest RME- is the best one.

At first we got like lot of worst accuracies because we tried lot of algorithms for the best accurate algorithm , finally after all of that we tried the best suitable algorithm which gives the prediction accurately is Linear Regression Algorithm. And developed it to use as a real time prediction

**3.1 Block Diagram**

**3.2 Software Designing**

● Jupyter Notebook Environment and Google Colab

● Spyder IDE

● Machine Learning Algorithms

● Python (pandas, numpy, matplotlib, seaborn, sklearn)

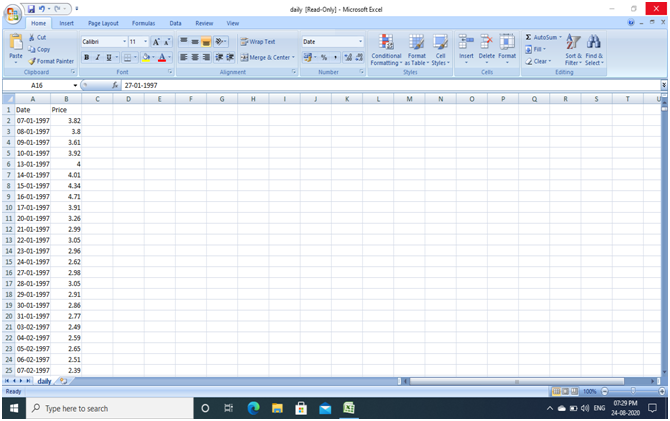
● HTML

● Flask

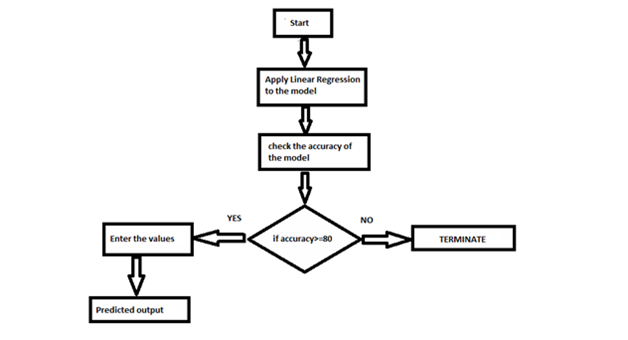
We developed this natural gas price prediction by using the Python language which is a interpreted and high level programming language and using the Machine Learning algorithms. for coding we used the Google Colab and Jupyter Notebook environment of the Anaconda distributions and the Spyder, it is an integrated scientific programming in the python language. For creating an user interface for the prediction we used the Flask. It is a micro web framework written in Python. It is classified as a micro framework because it does not require particular tools or libraries. It has no database abstraction layer, form validation, or any other components where pre-existing third-party libraries provide common functions, and a scripting language to create a webpage is HTML by creating the templates to use in the functions of the Flask and HTML.

4. **EXPERIMENTAL INVESTIGATION**

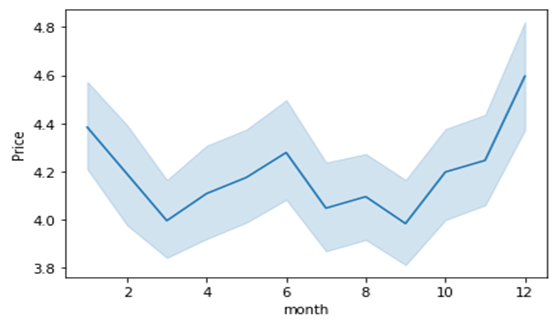
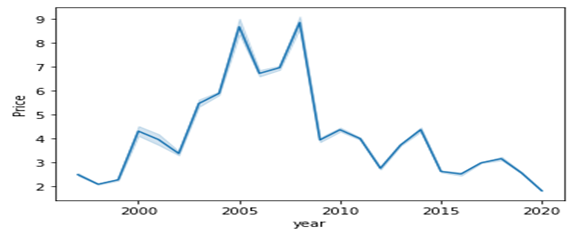
The dataset we used is derived from <https://github.com/datasets/natural-gas> It contains plenty of datasets which are real time. We choose natural gas price prediction dataset which contains 2 attributes and 5938 rows After that, the missing values are checked and the unwanted columns are deleted, and we have retained to attributes. Those are shown below.

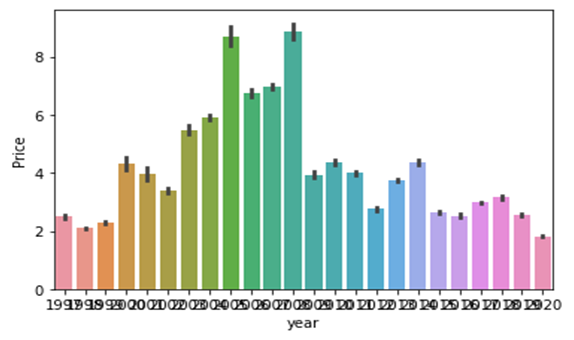


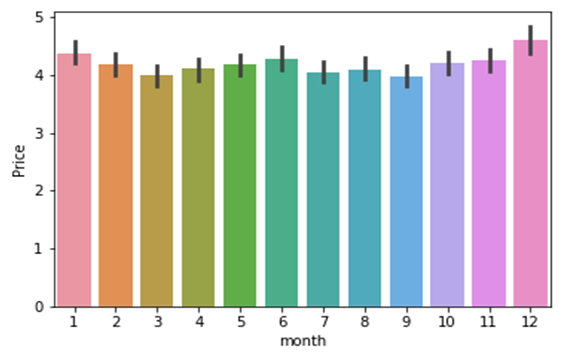
5. **FLOWCHART**

 **6. RESULT**

In this paper, the Linear Regression algorithm is used to predict its performance, and compared with another two machine learning methods namely the SVM, The obtained results are displayed in Table below. The results show that, the performance of Linear Regression have better performance than random forest and decision tree. The Linear Regression is best with an accuracy of 30% higher ,The given are theline plot of the dataset represents the correlation between attributes and thebar plot of each attribute







**7. ADVANTAGES AND DISADVANTAGES**

**Advantages:**

* It can work in real time and predicted as soon as the necessary details for prediction are given to the model.
* Natural gas price prediction has become a very valuable tool for all market participants in competitive natural gas markets
* The natural gas price advantage provides savings your family can count on.
* It is composed using the HTML and Python for the web usage in real time
* It is used for managing risks in investing in natural gas industry.

**Disadvantages:**

* It predicts the output within the range but not accurate value.
* It could not work anywhere like an web-application, if one is using other should be quiet
* It has always been a difficult task to predict the exact daily price
* Many factors such as political events, general economic conditions, and traders’ expectations may have an influence on the spot price index.
* Gives only 30 % accuracy for the price prediction

8. APPLICATIONS

* we can also use this model to predict , different sectors like business, bank, salaries etc...
* So we use Machine Learning Algorithms to analyze the data and propose what stakeholdersand business companies need to achieve their needs.
* Forecast the natural gas price benefits various stakeholders and business persons whether to buy or invest in natural gas industry with our model

9**. CONCLUSION**

In this paper, the Linear Regression algorithm is adopted to build a UI model for predicting natural gas price and the results are compared with other . The model shows that Linear Regression performs best than the other svm in the prediction of price. There is no definitive guide of which algorithms to be used. What may work on some datasets may not work on others. Therefore, always check the accuracy and predict with the dataset values.

**10. FUTURE SCOPE**

This Linear Regression model can also be used in the future predictions like weather forecast, job prediction, salary prediction etc. In further study, we will try to conduct experiments on larger data sets or try to tune the model so as to achieve the state -of-art performance of the model and a great UI support system making it complete web application

11. **BIBLIOGRAPHY**

● Adichie, J. N. (1967). Estimates of regression parameters based on rank test. Annals of Mathematical Statistics, 38, 894-904.

● Aia, M. A., Goldsmith, R. L., and Mooney, R. W. (1961). Predicting stoichiometric CaHP04 • 2H20.lndllstrial and Engineering Chemistry, 53, January, 55-57.

●Adcock, R. J. (1878). A problem in least squares. Analyst, 5, 53-54.

● Andrews, D. F. (1974). A robust method for multiple linear regression. Technometrics. 16. 523-531.

●Andrews, D. E, and D. Pregibon (1978). Finding the outliers that matter. Journal of the Royal Statistical Society, Series B, 4, 84-93.

●Barnett, V. D. (1967). A note on linear structural relationships when both residual variances arc known. Biometrika, 54, 670-672.

●Dobson. A. J. (1990). Introduction to Generalized Linear Models. London: Chapman and Hall.

●Marquardt, D. W. (1963). An algorithm for least squares estimation of nonlinear parameters. Journal of the Society for Industrial and Applied Mathematics, 11,431-441.

●McCullagh, P., and J. A. NeIder (1989). Generalized Linear Models, 2nd ed. London: Chapman and Hall.

●Schemper. M. (1990). The explained variation in proportional hazards regression. Biometrika, 77,216-218.

**APPENDIX**

**HTML:**

<!DOCTYPE html>

<html>

<!--From https://codepen.io/frytyler/pen/EGdtg-->

<head>

<meta charset="UTF-8">

<title>ML API</title>

<link href='https://fonts.googleapis.com/css?family=Pacifico' rel='stylesheet' type='text/css'>

<link href='https://fonts.googleapis.com/css?family=Arimo' rel='stylesheet' type='text/css'>

<link href='https://fonts.googleapis.com/css?family=Hind:300' rel='stylesheet' type='text/css'>

<link href='https://fonts.googleapis.com/css?family=Open+Sans+Condensed:300' rel='stylesheet' type='text/css'>

<link rel="stylesheet" href="../static/css/style.css">

<style>

.login {

top: 20%;

}

</style>

</head>

<body>

<font face = "Comic sans MS" size ="10">

Natural Gas Price Predication</font><br />

<!--<h2>Natural Gas Price Prediction</h2>-->

<div class="login">

<!-- Main Input For Receiving Query to our ML -->

<form action="{{ url\_for('y\_pred')}}" method="post">

<label for="year">YEAR</label>

<input type="text" name="year" placeholder="Enter year(yyyy)" id="year" required="required" <label for="month">MONTH</label>

<input type="text" name="month" placeholder="Enter month(mm)" id="month" required="required" /

<label for="day">DAY</label>

<input type="text" name="day" placeholder="Enter day(dd)" id="day" required="required" /> <button type="reset" class="btn btn-primary btn-large">Clear</button>

<button type="submit" class="btn btn-primary btn-large">Predict</button

<br>

<br> {{ prediction\_text }}

</form>

</div>

</body>

</html>

**CSS:**

@import url(https://fonts.googleapis.com/css?family=Open+Sans);

.btn {

display: inline-block;

margin-top: 1rem;

\*display: inline;

\*zoom: 1;

width: 40%;

padding: 4px 10px 4px;

margin-bottom: 0;

font-size: 13px;

line-height: 18px;

color: #333333;

text-align: center;

text-shadow: 0 1px 1px rgba(255, 255, 255, 0.75);

vertical-align: middle;

background-color: #f5f5f5;

background-image: -moz-linear-gradient(top, #ffffff, #e6e6e6);

background-image: -ms-linear-gradient(top, #ffffff, #e6e6e6);

background-image: -webkit-gradient(linear, 0 0, 0 100%, from(#ffffff), to(#e6e6e6));

background-image: -webkit-linear-gradient(top, #ffffff, #e6e6e6);

background-image: -o-linear-gradient(top, #ffffff, #e6e6e6);

background-image: linear-gradient(top, #ffffff, #e6e6e6);

background-repeat: repeat-x;

filter: progid: dximagetransform.microsoft.gradient(startColorstr=#ffffff, endColorstr=#e6e6e6, GradientType=0);

border-color: #e6e6e6 #e6e6e6 #e6e6e6;

border-color: rgba(0, 0, 0, 0.1) rgba(0, 0, 0, 0.1) rgba(0, 0, 0, 0.25);

border: 1px solid #e6e6e6;

-webkit-border-radius: 4px;

-moz-border-radius: 4px;

border-radius: 4px;

-webkit-box-shadow: inset 0 1px 0 rgba(255, 255, 255, 0.2), 0 1px 2px rgba(0, 0, 0, 0.05);

-moz-box-shadow: inset 0 1px 0 rgba(255, 255, 255, 0.2), 0 1px 2px rgba(0, 0, 0, 0.05);

box-shadow: inset 0 1px 0 rgba(255, 255, 255, 0.2), 0 1px 2px rgba(0, 0, 0, 0.05);

cursor: pointer;

\*margin-left: .3em;

}

.btn:hover,

.btn:active,

.btn.active,

.btn.disabled,

.btn[disabled] {

background-color: #e6e6e6;

}

.btn-large {

padding: 9px 14px;

font-size: 15px;

line-height: normal;

-webkit-border-radius: 5px;

-moz-border-radius: 5px;

border-radius: 5px;

}

.btn:hover {

color: #333333;

text-decoration: none;

background-color: #e6e6e6;

background-position: 0 -15px;

-webkit-transition: background-position 0.1s linear;

-moz-transition: background-position 0.1s linear;

-ms-transition: background-position 0.1s linear;

-o-transition: background-position 0.1s linear;

transition: background-position 0.1s linear;

}

.btn-primary,

.btn-primary:hover {

text-shadow: 0 -1px 0 rgba(0, 0, 0, 0.25);

color: #ffffff;

}

.btn-primary.active {

color: rgba(255, 255, 255, 0.75);

}

.btn-primary {

background-color: #4a77d4;

background-image: -moz-linear-gradient(top, #6eb6de, #4a77d4);

background-image: -ms-linear-gradient(top, #6eb6de, #4a77d4);

background-image: -webkit-gradient(linear, 0 0, 0 100%, from(#6eb6de), to(#4a77d4));

background-image: -webkit-linear-gradient(top, #6eb6de, #4a77d4);

background-image: -o-linear-gradient(top, #6eb6de, #4a77d4);

background-image: linear-gradient(top, #6eb6de, #4a77d4);

background-repeat: repeat-x;

filter: progid: dximagetransform.microsoft.gradient(startColorstr=#6eb6de, endColorstr=#4a77d4, GradientType=0);

border: 1px solid #3762bc;

text-shadow: 1px 1px 1px rgba(0, 0, 0, 0.4);

box-shadow: inset 0 1px 0 rgba(255, 255, 255, 0.2), 0 1px 2px rgba(0, 0, 0, 0.5);

}

.btn-primary:hover,

.btn-primary:active,

.btn-primary.active,

.btn-primary.disabled,

.btn-primary[disabled] {

filter: none;

background-color: #4a77d4;

}

.btn-block {

width: 100%;

display: block;

}

\* {

-webkit-box-sizing: border-box;

-moz-box-sizing: border-box;

-ms-box-sizing: border-box;

-o-box-sizing: border-box;

box-sizing: border-box;

}

\* {

box-sizing: border-box;

}

body {

background-repeat: no-repeat;

width: 100%;

height: 100%;

font-family: 'Open Sans', sans-serif;

background: #092756;

color: #fff;

font-size: 18px;

text-align: center;

letter-spacing: 1.2px;

background: -moz-radial-gradient(0% 100%, ellipse cover, rgba(104, 128, 138, .4) 10%, rgba(138, 114, 76, 0) 40%), -moz-linear-gradient(top, rgba(57, 173, 219, .25) 0%, rgba(42, 60, 87, .4) 100%), -moz-linear-gradient(-45deg, #670d10 0%, #092756 100%);

background: -webkit-radial-gradient(0% 100%, ellipse cover, rgba(104, 128, 138, .4) 10%, rgba(138, 114, 76, 0) 40%), -webkit-linear-gradient(top, rgba(57, 173, 219, .25) 0%, rgba(42, 60, 87, .4) 100%), -webkit-linear-gradient(-45deg, #670d10 0%, #092756 100%);

background: -o-radial-gradient(0% 100%, ellipse cover, rgba(104, 128, 138, .4) 10%, rgba(138, 114, 76, 0) 40%), -o-linear-gradient(top, rgba(57, 173, 219, .25) 0%, rgba(42, 60, 87, .4) 100%), -o-linear-gradient(-45deg, #670d10 0%, #092756 100%);

background: -ms-radial-gradient(0% 100%, ellipse cover, rgba(104, 128, 138, .4) 10%, rgba(138, 114, 76, 0) 40%), -ms-linear-gradient(top, rgba(57, 173, 219, .25) 0%, rgba(42, 60, 87, .4) 100%), -ms-linear-gradient(-45deg, #670d10 0%, #092756 100%);

background: -webkit-radial-gradient(0% 100%, ellipse cover, rgba(104, 128, 138, .4) 10%, rgba(138, 114, 76, 0) 40%), linear-gradient(to bottom, rgba(57, 173, 219, .25) 0%, rgba(42, 60, 87, .4) 100%), linear-gradient(135deg, #670d10 0%, #092756 100%);

filter: progid: DXImageTransform.Microsoft.gradient( startColorstr='#3E1D6D', endColorstr='#092756', GradientType=1);

}

.login {

width: 35%;

margin: 1rem auto;

}

h1 {

color: #fff;

font-size: 4rem;

text-shadow: 0 0 10px rgba(0, 0, 0, 0.3);

letter-spacing: 1px;

text-align: center;

padding: 0;

margin-top: 10px;

}

input {

text-align: center;

width: 100%;

margin-bottom: 1rem;

background: rgba(0, 0, 0, 0.3);

border: none;

outline: none;

padding: 10px;

font-size: 13px;

color: #fff;

text-shadow: 1px 1px 1px rgba(0, 0, 0, 0.3);

border: 1px solid rgba(0, 0, 0, 0.3);

border-radius: 4px;

box-shadow: inset 0 -5px 45px rgba(100, 100, 100, 0.2), 0 1px 1px rgba(255, 255, 255, 0.2);

-webkit-transition: box-shadow .5s ease;

-moz-transition: box-shadow .5s ease;

-o-transition: box-shadow .5s ease;

-ms-transition: box-shadow .5s ease;

transition: box-shadow .5s ease;

}

input:focus {

box-shadow: inset 0 -5px 45px rgba(100, 100, 100, 0.4), 0 1px 1px rgba(255, 255, 255, 0.2);

}

@media screen and (max-width: 700px) {

.login {

width: 60%;

}

h1 {

font-size: 2rem;

}

}

APP.PY:

from flask import Flask, request, render\_template

import pickle

model = pickle.load(open("F:/flaskapp/price.pkl", 'rb'))

app = Flask(\_\_name\_\_)

@app.route('/')

def home():

return render\_template('index.html')

@app.route('/y\_pred',methods=['POST'])

def y\_pred():

''

For rendering results on HTML GUI

'''

year =request.form['year']

month =request.form['month']

day =request.form['day']

data=[[ int(year),int(month),int(day)]]

prediction = model.predict(data)

print(prediction)

output=prediction[0][0]

return render\_template('index.html', prediction\_text=output)

if \_\_name\_\_ == "\_\_main\_\_":

app.run(debug=True)