# PROFESSIONAL TRAINING REPORT

**at**

**Sathyabama Institute of Science and Technology (Deemed to be University)**

Submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering Degree in Computer Science and Engineering

By

## Achanta Sai Bhavana Priya

**REG. NO. 39110016**

****

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**SCHOOL OF COMPUTING**

**SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY**

**JEPPIAAR NAGAR, RAJIV GANDHI SALAI,**

**CHENNAI – 600119, TAMILNADU**

**APRIL 2022**

|  |  |  |
| --- | --- | --- |
|  | **SATHYABAMA**  **INSTITUTE OF SCIENCE AND TECHNOLOGY** (DEEMED TO BE UNIVERSITY) **Accredited with Grade “A” by NAAC**  (Established under Section 3 of UGC Act, 1956)  JEPPIAAR NAGAR, RAJIV GANDHI SALAI  CHENNAI– 600119  [**www.sathyabama.ac.in**](http://www.sathyabama.ac.in) |  |

# DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

**BONAFIDE CERTIFICATE**

This is to certify that this Project Report is the bonafide work of **Achanta Sai Bhavana Priya (Reg. No: 39110016)** who carried out the project entitled “**GOLD PRICE PREDICTION USING MACHINE LEARNING ALGORITHMS”,** under my supervision from February 2022 to April 2022.

## Internal Guide

## Dr .M. SELVI

**Head of the Department**

**Dr .L. Lakshmanan**



## Submitted for Viva voce Examination held on

**InternalExaminer ExternalExaminer**

**DECLARATION**

I, **Achanta Sai Bhavana Priya** hereby declare that the project report entitled **GOLD PRICE PREDICTION USING MACHINE LEARNING ALGORITHMS** done by me under the guidance of **Dr .M. Selvi** is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering Degree in Computer Science and Engineering.

## DATE:10-04-2022

**A.Bhavana**

**PLACE:Rajahmundry SIGNATURE OF THE CANDIDATE**

**ACKNOWLEDGEMENT**

I am pleased to acknowledge my sincere thanks to **Board of Management** of **SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T. Sasikala M.E., Ph.D**, **Dean**, School of Computing, **Dr. S. Vigneshwari, M.E., Ph.D. and Dr. L. Lakshmanan, M.E., Ph.D., Heads of the Department** of **Computer Science and Engineering** for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr. M.Selvi.,** for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Computer Science and Engineering** who were helpful in many ways for the completion of the project

**CERTIFICATE**

# ABSTRACT

The demand for gold never ends. The gold rate trend shows that gold is one of the best investment plans. So it is wise to predict the trend of the gold rate.

Different statistical models can be used for suitable modelling and prediction of data. This paper summarizes the idea of gold rate prediction using the machine learning algorithms. It is always visible that the price of gold follows a nonlinear nature.

The prediction of prices is commanding for the proper financial and investment plans. The gold rate fluctuation can be modelled as an exponential curve.

Convolutional Neural Networks is one of the best solutions for solving nonlinearities in data and among them RNNs are best suited for time series predictions and estimations.

The dataset collected from the World Gold Council is used and

results have shown that the proposed architecture is one of the best financial forecasting methods.

|  |  |  |
| --- | --- | --- |
|  | **Table of Contents** |  |
|  |  |  |
| Chapter No. | **Title** | **Page No.** |
| 1 | Abstract | 6 |
|  | List of Figures | 15-16 |
|  | Fig1 | 15 |
|  | Fig2 | 16 |
| 2 | Introduction | 9 |
| 3 | Objective | 10 |
| 4 | System Architecture | 10 |
| 5 | Description of Implementation | 11 |
|  | **5.1** Data Collection | 11 |
|  | **5.2** Data pre-processing | 11 |
|  | **5.3** Feature Extraction | 11 |
|  | **5.4** Apply Machine learning algorithm | 11 |
|  | **5.5** Prediction of gold price | 12 |
| 6 | System Requirements | 12 |
|  | **6.1** Hardware Requirements | 12 |
|  | **6.2** Software Requirements | 12 |
| 7 | Methodology |  |
|  | **7.1** Logistic regression algorithm | 12 |
|  | **7.2** LogisticRegression Algorithm | 13 |
|  | **7.3** Steps in Logistic Regression | 14 |
| 8 | Results and Discussion | 15 |
| 9 | Conclusion | 17 |
| 10 | Code | 18 |

**2.INTRODUCTION**

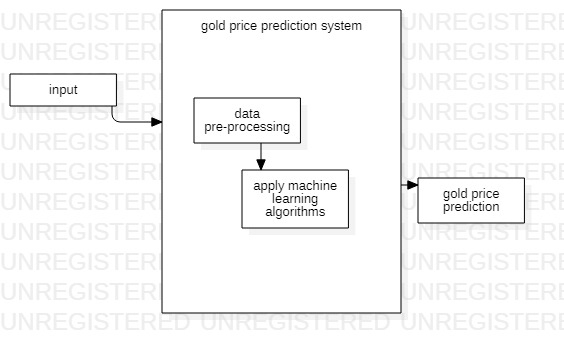
The fluctuations in the market, economic policies, epidemics, weather conditions, etc are some of the factors that greatly affect it. The financial area is highly complex and nonlinear with profuse factors influencing the estimation.

So the financial data always shows a nonlinear behaviour.The nonlinearity in the data can be extracted to analyse the behaviour and is an exquisite nowadays so that proper financial plans can be executed.

People used to go behind the gold price analysis even in the early seventies and they analysed that government policies in the US are a key factor in gold pricing.

**3.OBJECTIVE**

* The main objective of the system is to predict gold price by collecting the data set and by applying machine learning algorithms.

**4.SYSTEM ARCHITECTURE **

**5.DESCRIPTION OF IMPLEMENTATION:**

**5.1 Data Collection:** The data set is collected from the kaggle dataset which is an open source which consists of 6 attributes and entries more than 2k.

* 1. **Data pre-processing:** The dataset which is uploaded is previewed and the process of finding out missing values or checking if any redundancy is present checked and remove all null values, redundancy and noise data is the dataset.
  2. **Feature Extraction:** Here the important features i.e from the dataset which features are required for the prediction of gold price is required are identified.
  3. **Apply Machine learning algorithm:** After training the dataset machine learning algorithms are applied to the selected features then it will predict the gold price prediction.
  4. **Prediction of gold price:** After applying machine learning algorithms it will predict gold price by comparing with each and every year and provides accuracy.

**6. SYSTEM REQUIREMENTS**

**6.1 HARDWARE REQUIRMENTS**

* **HardDisk : 40 Gb**
* **Ram : 512 Mb**

**6.2 SOFTWARE REQUIRMENTS**

* **Operating system : Window 7(32 bit)**
* **Coding Language : Python**
* **IDE : Jupyter Notebook**

**7. METHODOLOGY**

* Random forest algorithm
* Logistic regression algorithm

**7.1 Logistic regression algorithm:**

* Logistic regression is one of the most popular Machine Learning algorithms, which comes under the Supervised Learning technique. It is used for predicting the categorical dependent variable using a given set of independent variables.
* Logistic regression predicts the output of a categorical dependent variable. Therefore, the outcome must be a categorical or discrete value. It can be either Yes or N
* o, 0 or 1, true or False, etc. but instead of giving the exact value as 0 and 1, it gives the probabilistic values which lie between 0 and 1.
* Logistic Regression is much similar to the Linear Regression except that how they are used. Linear Regression is used for solving Regression problems, whereas Logistic regression is used for solving the classification problems.

**7.2 Logistic regression equation:**

Logistic Regression in Machine Learning

Logistic Regression in Machine Learning

Logistic Regression in Machine Learning

**7.3 Steps in Logistic Regression:**

To implement the Logistic Regression using Python, we will use the same steps as we have done in previous topics of Regression. Below are the steps:

1. Data Pre-processing step

2. Fitting Logistic Regression to the Training set

3. Predicting the test result

4. Test accuracy of the result(Creation of Confusion matrix)

5. Visualizing the test set result.

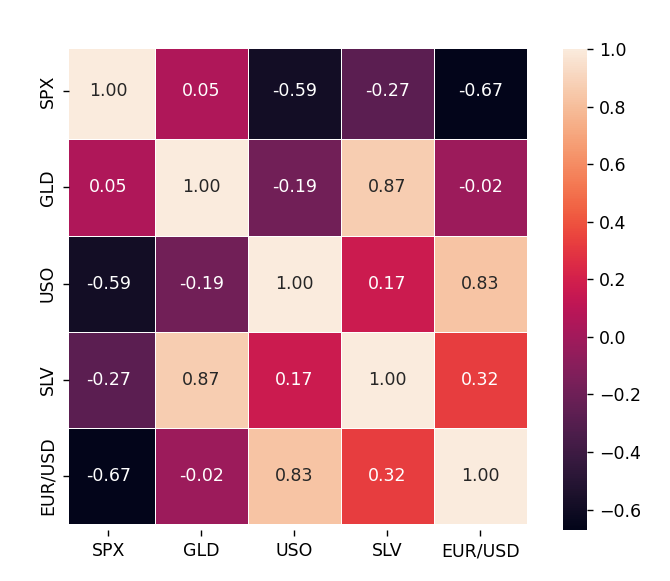


Fig1

**RESULTS AND DISCUSSIONS**

* To obtain an authenticate dataset, the website of the World Gold Council is accessed and the data from the year 1987 to July 2020, including yearly, quarterly, monthly, and daily gold price in INR, US dollar, EURO, RMB, HK dollar, etc. per ounce. is downloaded. To be more specific, dataset specified in daily gold price in INR per ounce is chosen to make analysis predict.

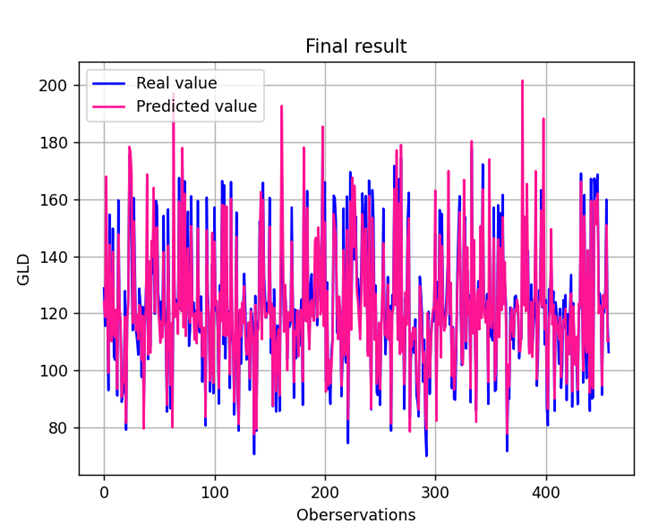


Fig2

We’ve got an accuracy of almost 98%

**CONCLUSION**

* In this system, an Random forest and logistic regression is used to predict the nature of the gold rate. The results indicate that our proposed model outplays traditional methods such as machine learning algorithms.
* Moreover, it would be better to test different machine learning algorithm to predict gold price is better to find with the gold price with achieving high accuracy.

**Code**

# LinearRegression is a machine learning library for linear regression

from sklearn.linear\_model import LinearRegression

from sklearn.metrics import r2\_score

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

from sklearn.metrics import mean\_squared\_error

# pandas and numpy are used for data manipulation

import pandas as pd

import numpy as np

from math import sqrt

from numpy import log

from pandas import Series

from statsmodels.tsa.arima\_model import ARMA

from statsmodels.tsa.arima\_model import ARIMA

from statsmodels.tsa.stattools import adfuller, arma\_order\_select\_ic

from statsmodels.graphics.tsaplots import plot\_acf

from statsmodels.graphics.tsaplots import plot\_pacf

import statsmodels as sm

# matplotlib and seaborn are used for plotting graphs

import matplotlib.pyplot as plt

from matplotlib.dates import date2num

import seaborn as sns

from datetime import datetime

import subprocess

# fix\_yahoo\_finance is used to fetch data

# import fix\_yahoo\_finance as yf

ds\_gold = 'Indian rupee'

ds\_etf = 'Close'

date\_format = '%Y-%m-%d'

df = pd.read\_csv("data\_inr.csv")

df = df[['Name', ds\_gold]]

df['Name'] = [datetime.strptime(i, date\_format) for i in df['Name']]

df.set\_index('Name')

# df.index = pd.to\_datetime(df.index, format=date\_format)

print(df.columns)

dd =df

"""\* Drop rows with missing values"""

df = df.dropna()

df[ds\_gold].hist()

plt.show()

log\_transform = log(df[ds\_gold])

print(min(log\_transform), max(log\_transform))

sns.set()

sns.distplot(df[ds\_gold])

plt.show()

plt.plot(df['Name'], df[ds\_gold])

plt.show()

plt.plot(df['Name'], log\_transform)

plt.show()

# Finding moving average of past 3 days and 9 days

df['S\_1'] = df[ds\_gold].shift(1).rolling(window=3).mean()

df['S\_2'] = df[ds\_gold].shift(1).rolling(window=12).mean()

df = df.dropna()

X = df[['S\_1', 'S\_2']]

X.head()

plt.plot(df['Name'], df['S\_1'])

plt.plot(df['Name'], df["S\_2"])

plt.show()

# dependent variable

y = df[ds\_gold]

y.head()

# Split into train and test

t = 0.2

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

X, y, test\_size=t, shuffle=False)

# Performing linear regression

linear = LinearRegression().fit(X\_train, y\_train)

print("Gold Price =", round(linear.coef\_[0], 2), "\* 2 Month Moving Average", round(

linear.coef\_[1], 2), "\* 1 Month Moving Average +", round(linear.intercept\_, 2))

# Predict prices

predicted\_price = linear.predict(X\_test)

predicted\_price = pd.DataFrame(

predicted\_price, index=y\_test.index, columns=['price'])

predicted\_price.plot(figsize=(10, 5))

y\_test.plot()

plt.legend(['predicted\_price', 'actual\_price'])

plt.ylabel("Gold Price")

plt.show()

# Calculate R square and rmse to check goodness of fit

r2\_score = linear.score(X\_test, y\_test)\*100

print("R square for regression", float("{0:.2f}".format(r2\_score)))

sqrt(mean\_squared\_error(y\_test,predicted\_price))

# Check stationarity

X = df[ds\_gold]

split = len(X) // 2

X1, X2 = X[0:split], X[split:]

mean1, mean2 = X1.mean(), X2.mean()

var1, var2 = X1.var(), X2.var()

print('mean1=%f, mean2=%f' % (mean1, mean2))

print('variance1=%f, variance2=%f' % (var1, var2))

result\_of\_adfuller = adfuller(df[ds\_gold])

print('ADF Statistic: %f' % result\_of\_adfuller[0])

print('p-value: %f' % result\_of\_adfuller[1])

print('Critical Values:')

for key, value in result\_of\_adfuller[4].items():

print('\t%s: %.3f' % (key, value))

# Now taking log transform

log\_transform = log(df[ds\_gold])

result\_of\_adfuller = adfuller(log\_transform)

print('ADF Statistic: %f' % result\_of\_adfuller[0])

print('p-value: %f' % result\_of\_adfuller[1])

print('Critical Values:')

for key, value in result\_of\_adfuller[4].items():

print('\t%s: %.3f' % (key, value))

# To remove trends, differencing of order 1

k = df[ds\_gold].diff()

plt.plot(df['Name'], k)

plt.show()

# print(k.head())

k = k.dropna()

# check stationarity after differencing

result\_of\_adfuller = adfuller(k)

print('ADF Statistic: %f' % result\_of\_adfuller[0])

print('p-value: %f' % result\_of\_adfuller[1])

print('Critical Values:')

for key, value in result\_of\_adfuller[4].items():

print('\t%s: %.3f' % (key, value))

# So now we can say with 1 % confidence level that its stationary

# We can do other stuff now

# Again regression

df[ds\_gold] = k

# Finding moving average of past 3 days and 9 days

df['S\_1'] = df[ds\_gold].shift(1).rolling(window=3).mean()

df['S\_2'] = df[ds\_gold].shift(1).rolling(window=12).mean()

df = df.dropna()

X = df[['S\_1', 'S\_2']]

X.head()

print(X.head())

plt.plot(df['Name'], df['S\_1'])

plt.plot(df['Name'], df["S\_2"])

plt.show()

df['S\_1'] = df[ds\_gold].shift(1).rolling(window=3).mean()

df['S\_2'] = df[ds\_gold].shift(1).rolling(window=12).mean()