Project in Applied Machine Learning

Predicting Specific Drugs for Treating Covid-19 Patients

Team:Tech Zone

Team Member 1-Sayani Ghosh
Team Member 2-Poushali Chakrabarty
Team Member 3-Tanushree Mundha

Executive Summary

"Why" - Problem Statement

Covid protein Neocapsid-N, tends to attack human RNA,thereby replicating in huge number,which may turn fatal.But according to few researches, there exists few drugs which has high binding affinity with the covid protein.These drugs ,according to expectations,on entering body, will attract covid protein molecules to form bonds. If the binding affinity is hogh enough ,then the covid protein may leave human RNA in order to form more stable complex with the drug, thus making the patient corona free.Our project deals with this concept and finds out how much fruitful this method is. We have collected binding affinity values of few drugs, and also the number of cases in which these drugs were used to successfully treat corona patients.

"What" - exact ML classification or regression problem

We are planning to work on a multiple regression problem to determine the number of people cured in USA.

"How"-ML Techniques, Features used,Software/language used

Data Collection : We have collected data from many sources.

List of Important Features in the Dataset:

1. Drug Name
2. Docking Value
3. Binding Affinity
4. Number of People Cured in USA

Software/Technology Used:

1. Python (Data Analysis & Building Model - Pandas, Numpy,Scikit-learn)
2. Matplotlib - visualization

Conclusion - how did the model perform and what was result

1.) Our model score is 0.9905.

2.) The root mean square error value of our model is 0.0896.

Thus we found out that both binding affinity and docking values have direct impact on number of people getting cured.

But we feel that the model could be further improved if we only could have got a larger dataset, which could have further incresed the model score.

Project Description

Predict the structure of proteins and their interactions with chemical compounds to facilitate new antiviral drugs/vaccines or recommend current drugs. Methods here rely on applying machine learning to molecules such as proteins. This is a somewhat niche area that generally has a high learning curve to understand. However, breakthroughs here could potentially pave the way to vaccines or an effective antiviral. 1. Drug screening for Novel Coronavirus-2019:~ Here machine learing will be used to detect which antiviral will be used against corona virus.Firstly, protein-ligand interactions wil be done. then reactions will be made in between RNA of coronavirus and drugs to predict which drug is more adaptable in human body.

Covid protein has a tendency to form hydrogen bonds. So based on binding affinity, docking score, stability after forming complex, we need to decide which drug is more effective in forming hydrogen bonds with the covid protein. As soon as covid protein forms stable complex with the drug ligand, the bond formed between corona protein and human RNA breaks, thus becoming corona free. The corona-drug complex is eventually excreted out..

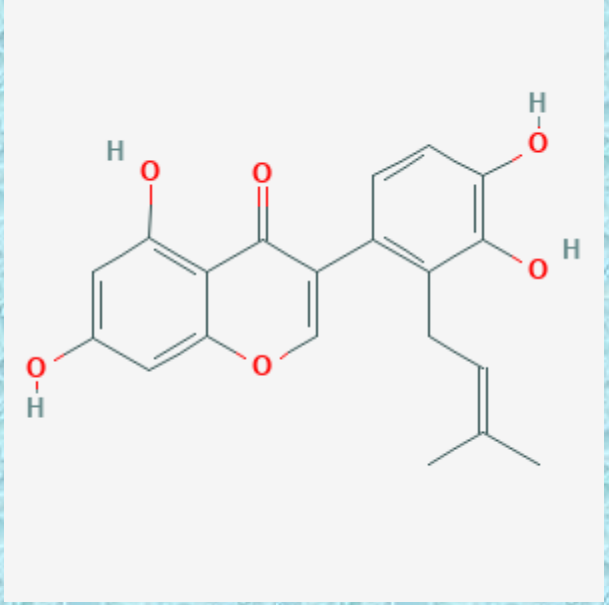
So type of data will be, binding affinity , docking score and stability measures of various compounds with corona protein.We will enter affinity, enthalpy,docking values and check stabilities off different drug-covid protein complex.The greater the stability of the complex is ,the more effective is the drug.

Aims and Objectives

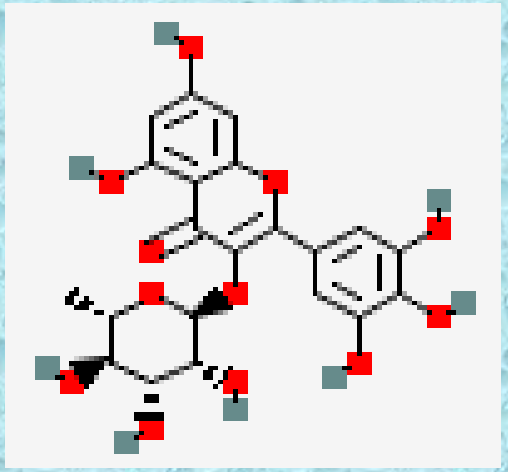
To detect the more effective drug using supervised learning.We have collected a dataset to train our model .Our input parameters are binding affinity,docking values,drug names, and number of people cured.Based on the dataset provided the model will predict the effectiveness of any drug in this concern on providing binding affinity and docking value of that drug. We further plot a bar diagram to show the comparative effectivity of the drugs in the dataset. We also plot scattered plots of number of people cured based on binding affinity of the drugs and that of docking values separately.We further calculate the model score and the root mean square error of our model ,which will depict the accuracy of our project.

Molecular Structureof Few Important Drugs

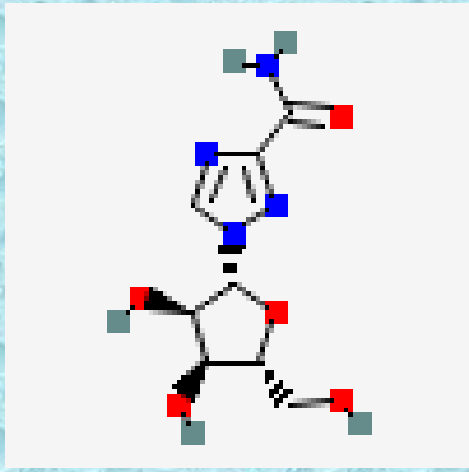
1. 5,7,3'4'-Tetrahydroxy-2'-(3,3-dimethylally)isoflavone



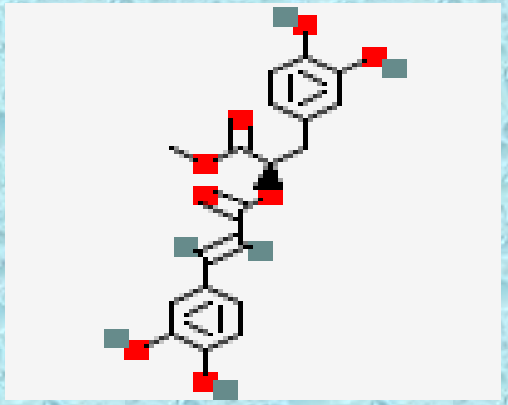
2. Myricitrin



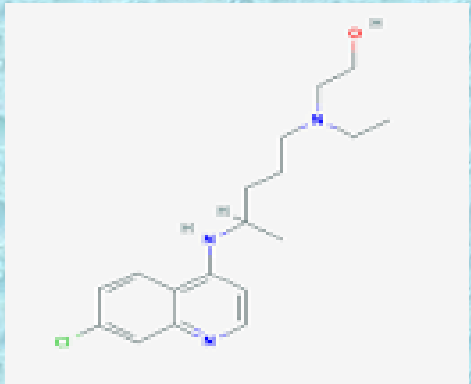
4. 37542,ribavirin



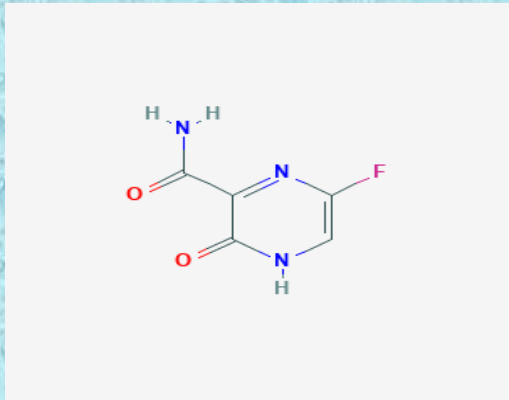
3. Methyl rosmarinate



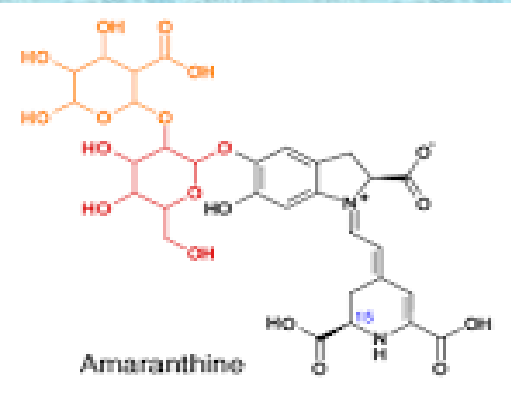
5. Hydroxychloroquine



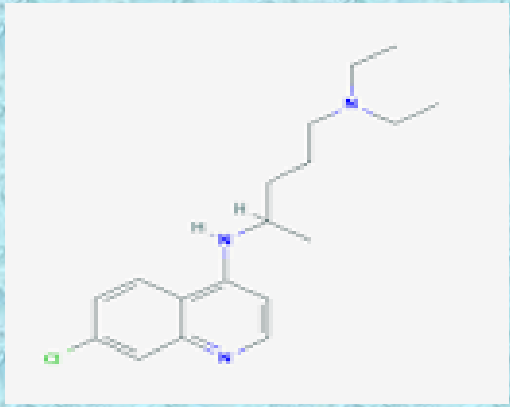
6. 492405,favipiravir



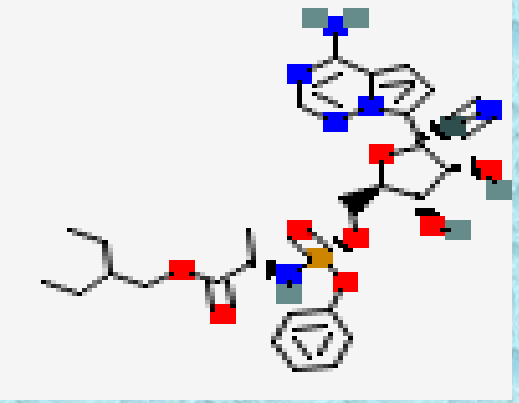
8. Amaranthin



7. Chloroquine



9. Remdesivir



Screenshots of the Code for our Project

In [4]:
```
import pandas as pd
import numpy as np
import matplotlib.ticker as ticker
import matplotlib.pyplot as plt
import math
from sklearn.metrics import mean_squared_error
df=pd.read_csv('cc.csv')

df
```

Out[4]:

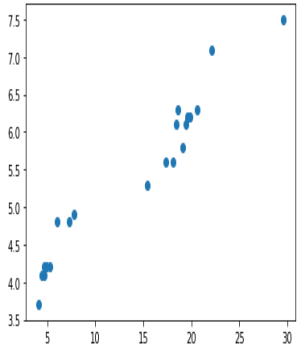
	Drug_Name	Docking_Value_in_-ve	BindingAffinity_in_-ve	Number_of_people_cured_in_USA(in10K)
0	5,7,3'4'-Tetrahydroxy-2'-(3,3-dimethylally)iso...	16.35	29.57	7.5
1	Myricitrin	15.64	22.13	7.1
2	Methyl rosmarinate	15.44	20.62	6.3
3	3,5,7,3',4',5'-hexahydroxyflavanone-3-O-beta-D...	14.42	19.10	5.8
4	(2S)-Eriodictyol7-O-(6''-O-galloyl)-beta-D-glu...	14.41	19.47	6.1
5	CalceolariosideB	14.36	19.87	6.2
6	Myricetin3-O-beta-D-glucopyranoside	13.70	18.42	6.1
7	Licoleafol	13.63	19.64	6.2
8	Amaranthin	12.67	18.14	5.6
9	Nelfinavir	12.20	17.31	5.6
10	Prulifloxacin	11.32	15.40	5.3

17	Remdesivir	7.59	4.96	4.2
18	Oseltamivir	8.90	4.70	4.1
19	Ritonavir	12.76	7.30	4.8
20	Chloroquine	9.60	5.30	4.2

In [4]:
```
import matplotlib.pyplot as plt
%matplotlib inline
```

In [6]:
```
plt.scatter(df['BindingAffinity_in_-ve'],df['Number_of_people_cured_in_USA(in10K)'])
```

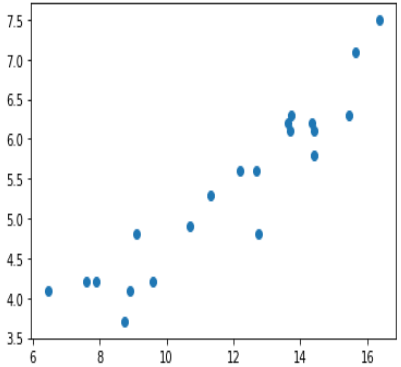
Out[6]:
```
<matplotlib.collections.PathCollection at 0xe93d8f0>
```



In [7]:
```
plt.scatter(df['Docking_Value_in_-ve'],df['Number_of_people_cured_in_USA(in10K)'])
```

In [7]:
```
plt.scatter(df['Docking_Value_in_-ve'],df['Number_of_people_cured_in_USA(in10K)'])
```

Out[7]:
```
<matplotlib.collections.PathCollection at 0xe9cd510>
```



In [8]:
```
x=df[['Docking_Value_in_-ve','BindingAffinity_in_-ve']]
y=df['Number_of_people_cured_in_USA(in10K)']
```

In [9]:
```
x
```

Out[9]:

	Docking_Value_in_-ve	BindingAffinity_in_-ve
0	16.35	29.57

```
y df[ Number_of_people_cured_in_USA(in10K) ]
```

In [9]:
```
x
```

Out[9]:

	Docking_Value_in_-ve	BindingAffinity_in_-ve
0	16.35	29.57
1	15.64	22.13
2	15.44	20.62
3	14.42	19.10
4	14.41	19.47
5	14.36	19.87
6	13.70	18.42
7	13.63	19.64
8	12.67	18.14
9	12.20	17.31
10	11.32	15.40
11	13.73	18.57
12	8.73	4.06
13	10.70	7.77
14	6.45	4.47
15	7.90	4.69
16	9.10	6.06

```
14    4.1
15    4.2
16    4.8
17    4.2
18    4.1
19    4.8
20    4.2
Name: Number_of_people_cured_in_USA(in10K), dtype: float64
```

In [45]:
```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.1)
```

In [39]:
```
x_train
```

Out[39]:

	Docking_Value_in_-ve	BindingAffinity_in_-ve
14	6.45	4.47
0	16.35	29.57
4	14.41	19.47
19	12.76	7.30
11	13.73	18.57
6	13.70	18.42
8	12.67	18.14
12	8.73	4.06
15	7.90	4.69
20	9.60	5.30

7	13.63	19.64

```
In [7]: len(x_train)
Out[7]: 18

In [8]: len(x_test)
Out[8]: 3
```

```
In [46]: from sklearn.linear_model import LinearRegression
         reg=LinearRegression()

In [47]: reg.fit(x_train,y_train)

Out[47]: LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None, normalize=False)

In [48]: y_pre=reg.predict(x_test)
         y_pre

Out[48]: array([4.10535146, 4.19919825, 6.02718282])

In [49]: y_test

Out[49]: 15    4.2
         18    4.1
         6     6.1
         Name: Number_of_people_cured_in_USA(in10K), dtype: float64

In [50]: reg.score(x_test,y_test)

Out[50]: 0.9905114247192736
```

```
Name: Number_of_people_cured_in_USA(in10K), dtype: float64
```

In [50]:
```
reg.score(x_test,y_test)
```

Out[50]:
```
0.9905114247192736
```

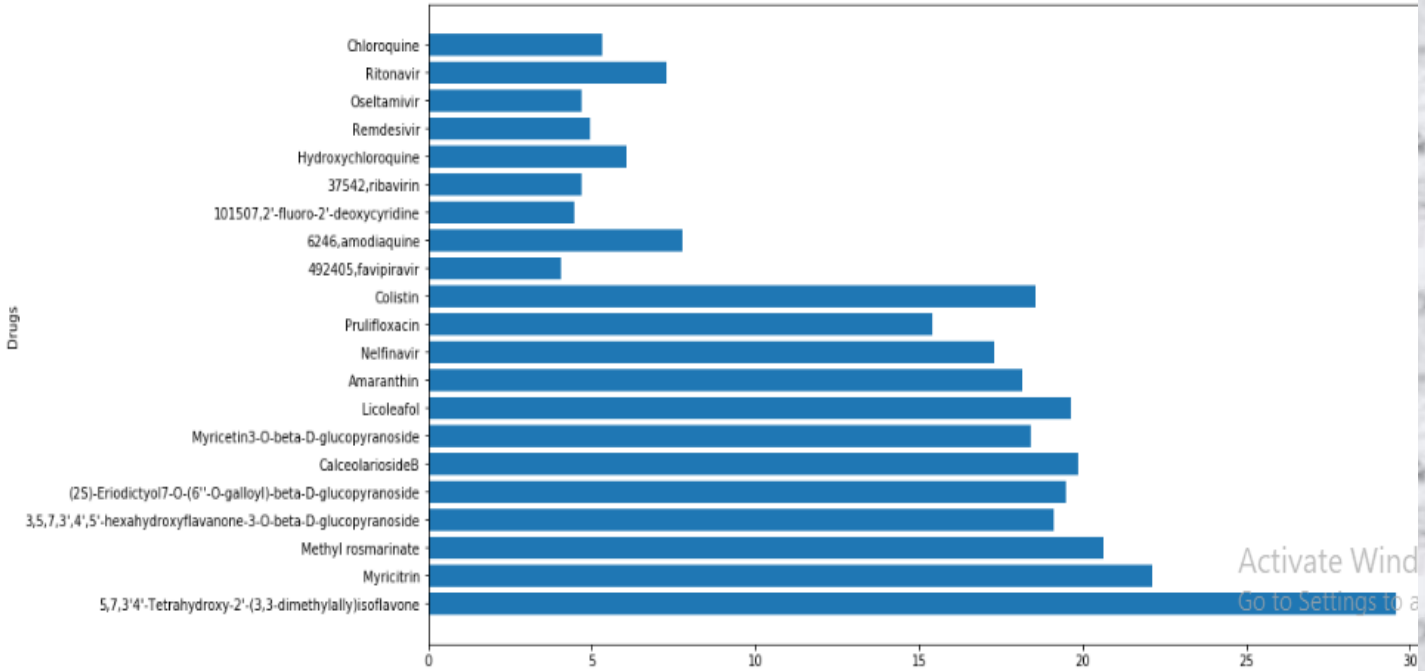
In [51]:
```
import sklearn
mse=sklearn.metrics.mean_squared_error(y_test,y_pre)
rmse=math.sqrt(mse)
rmse
```

Out[51]:
```
0.08963068896504403
```

Out[51]: 0.08963068896504403

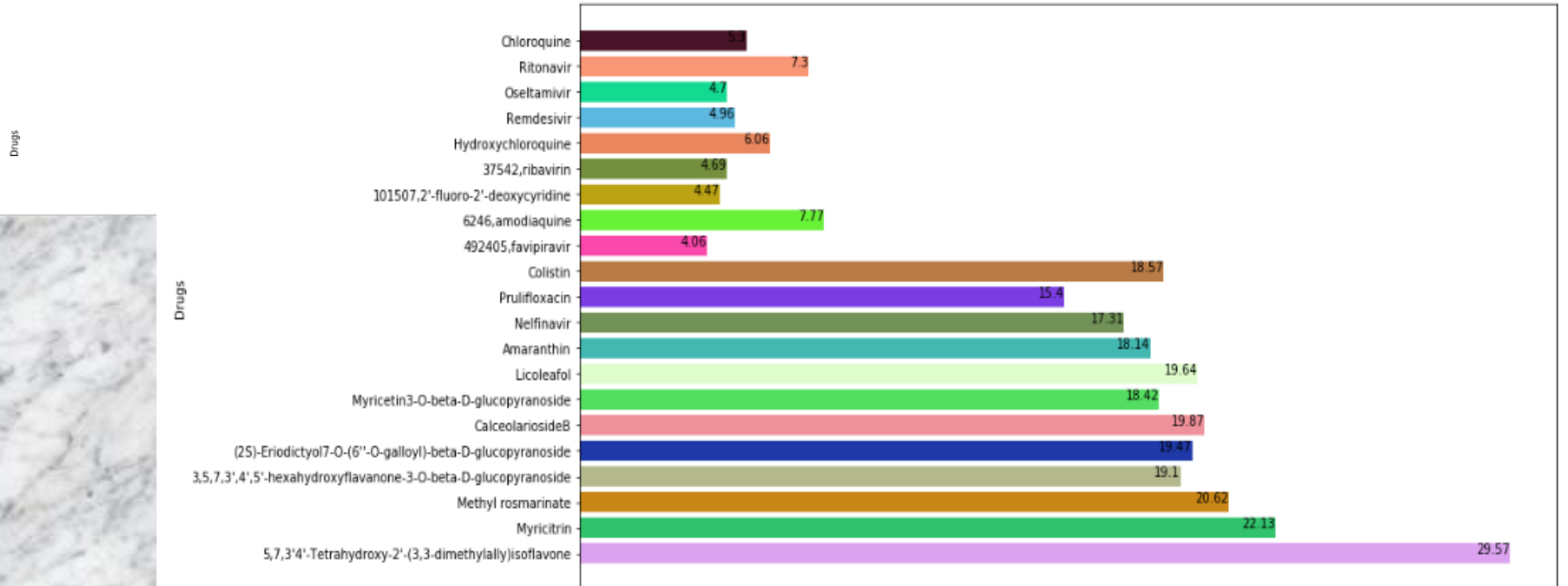
In [52]:
```
fig,ax=plt.subplots(figsize=(15,8))
ax.barh(df['Drug_Name'],df['BindingAffinity_in_-ve'])
plt.xlabel('Effectivity')
plt.ylabel('Drugs')
```

Out[52]: Text(0, 0.5, 'Drugs')

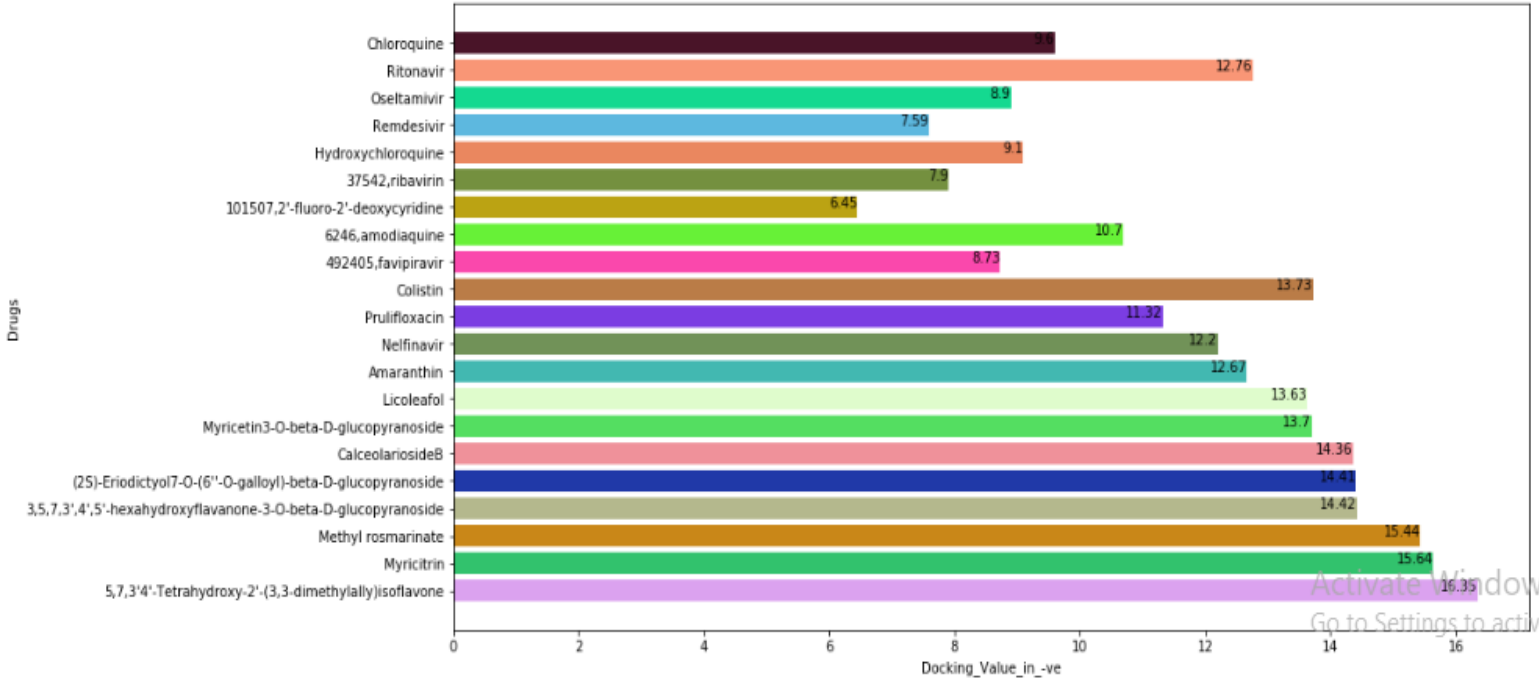


In [6]:
```
from random import randint
import random
c_code=[]
random.seed(1000)
for i in range(len(df.Drug_Name.unique())):
    c_code.append('#%06X' % randint(0,0xFFFFFF))
colors=dict(zip(df.Drug_Name.unique(),c_code))
```

In [10]:
```
fig,ax=plt.subplots(figsize=(15,8))
plt.xlabel('BindingAffinity_in_-ve')
plt.ylabel('Drugs')
ax.barh(df['Drug_Name'],df['BindingAffinity_in_-ve'],color=[colors[x] for x in df['Drug_Name']])
for i, (value,name) in enumerate(zip(df['BindingAffinity_in_-ve'],df['Drug_Name'])):
    ax.text(value, i,   value,           ha='right')
```



In [11]:
```
fig,ax=plt.subplots(figsize=(15,8))
plt.xlabel('Docking_Value_in_-ve')
plt.ylabel('Drugs')
ax.barh(df['Drug_Name'],df['Docking_Value_in_-ve'],color=[colors[x] for x in df['Drug_Name']])
for i, (value,name) in enumerate(zip(df['Docking_Value_in_-ve'],df['Drug_Name'])):
    ax.text(value, i,   value,           ha='right')
```



Conclusion

From the project output it is very much evident that the effectivity of the drugs is directly proportional to its binding affinity and docking value. The higher these values are, the greater is the effectivity of the corresponding drug. Thus our approach may serve as an important aspect in corona related researches. Our model shows high model score and low rmse value. We have using linear regression using multiple parameters.

Our model could have been further improved if we could have got a larger dataset.We could get only a few drug details as not much resources in this regard is available in the internet.

Acknowledgement

We have collected our dataset from various websites, namely:
1)www.kaggle.com
2)www.researchgate.com
3)www.mdpi.com
4)pubs.acs.org

Thank You