Name: Allan Rodrigues
Class: TE IT A
Roll no: 59
Pid:191104

St. Francis Institute of Technology, Mumbai-400 103
Department of Information Technology

A.Y. 2021-2022
Class: TE-ITA/B, Semester: VI

Subject: **Data Science Lab**

Experiment – 6: To implement Classification.

1. **Aim:** To implement classification modeling and evaluate performance of classifiers.
2. **Objectives:** After study of this experiment, the student will be able to
   - Understand classification.
3. **Outcomes:** After study of this experiment, the student will be able to
   - Understand concepts of classification in data science.
4. **Prerequisite:** Fundamentals of Python Programming and Database Management System.
5. **Requirements:** Python Installation, Personal Computer, Windows operating system, Internet Connection, Microsoft Word.
6. **Pre-Experiment Exercise:**

   Brief Theory:

   - Concept of classification in machine learning**.** (Naive Byes, ID3, KNN, Random Forest)

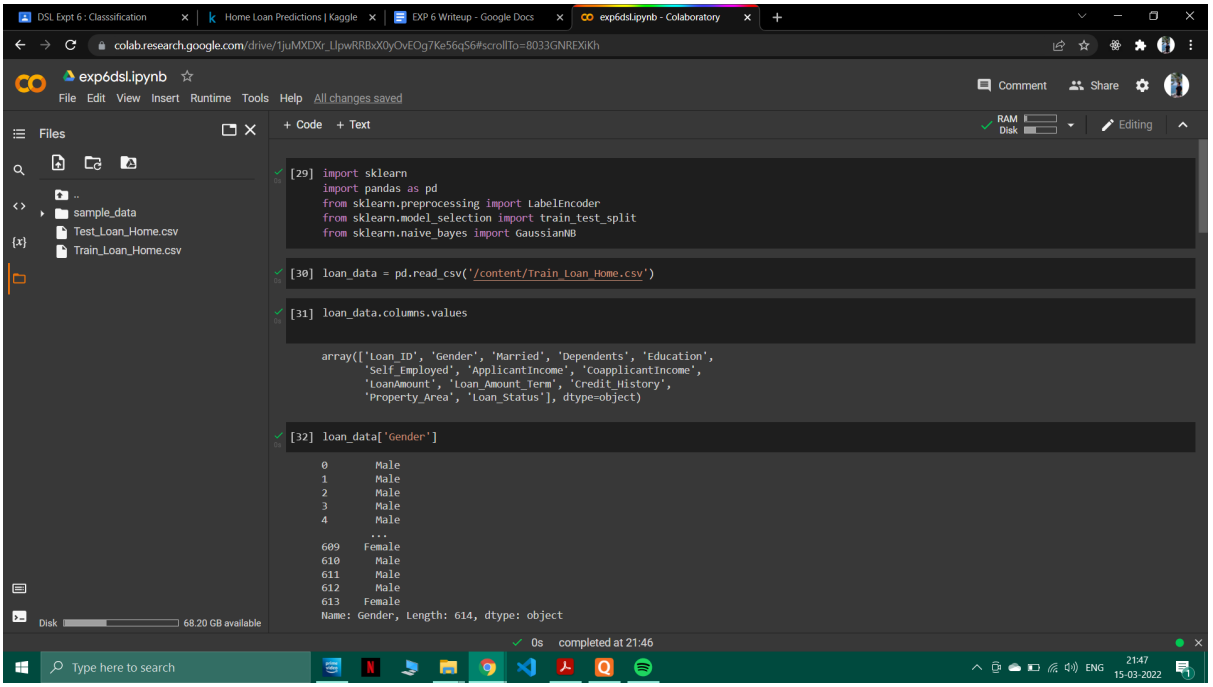
   Laboratory Exercise

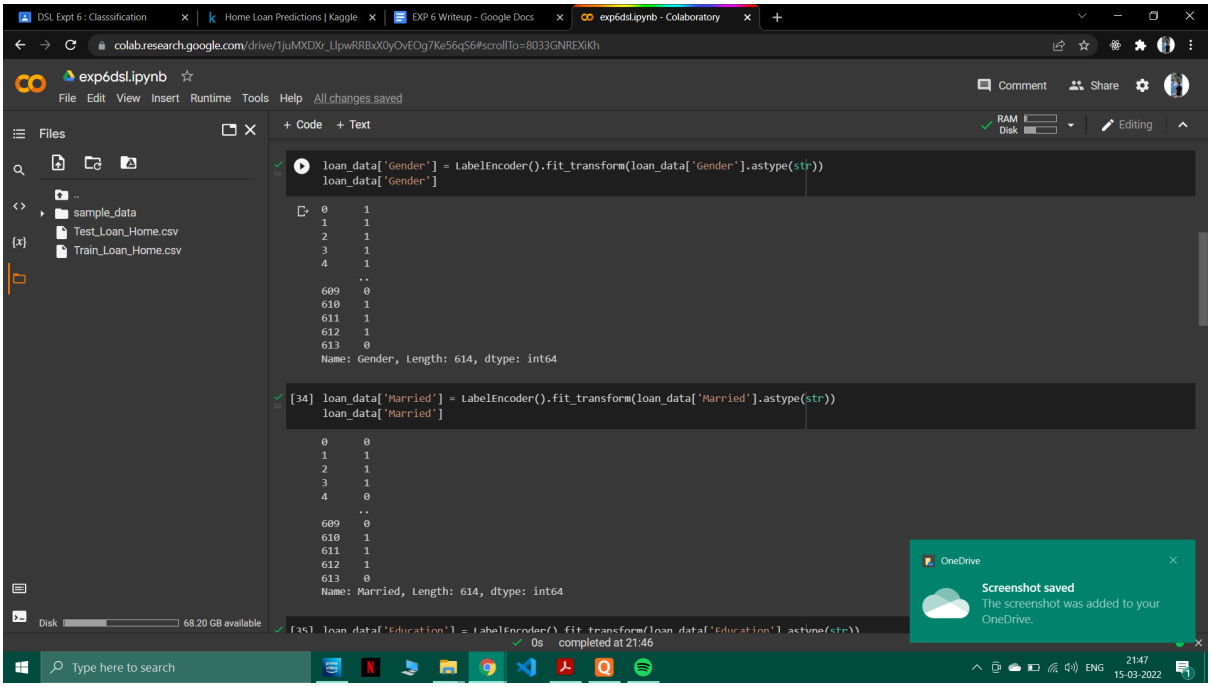
   A. **Procedure:** (Home _Loan Dataset)

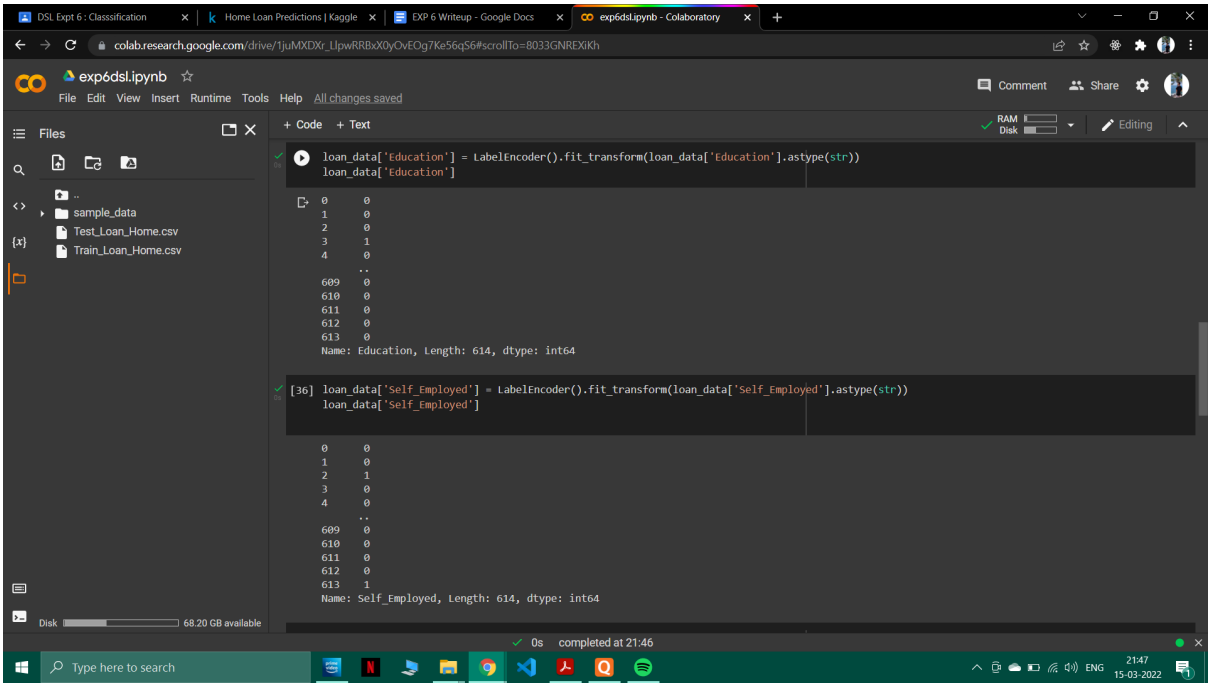
```
import sklearn
import pandas as pd
from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import GaussianNB

loan_data = pd.read_csv('home_loan_train.csv')

#print(loan_data.columns.values)

#print(loan_data['Gender'])
loan_data['Gender'] =
LabelEncoder().fit_transform(loan_data['Gender'].astype(str))
```

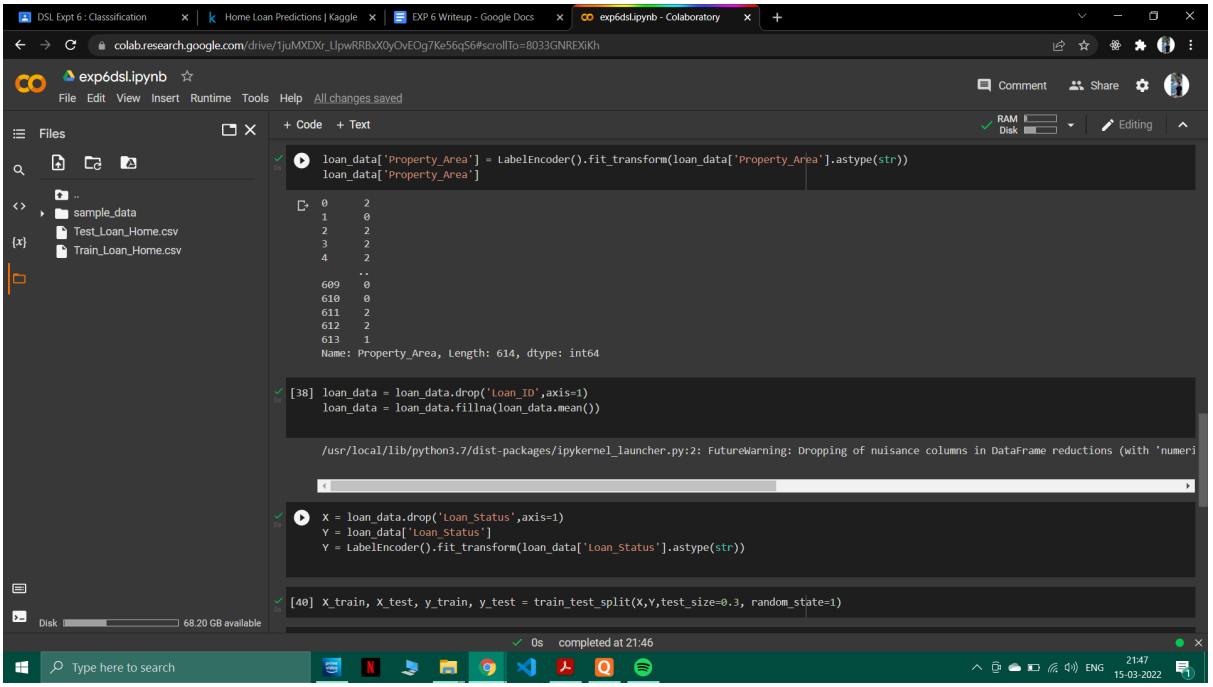
```
#print(loan_data['Gender'])

loan_data['Married'] =
LabelEncoder().fit_transform(loan_data['Married'].astype(str))
#print(loan_data['Married'])

loan_data['Education'] =
LabelEncoder().fit_transform(loan_data['Education'].astype(str))
#print(loan_data['Education'])

loan_data['Self_Employed'] =
LabelEncoder().fit_transform(loan_data['Self_Employed'].astype(str))
#print(loan_data['Self_Employed'])

loan_data['Property_Area'] =
LabelEncoder().fit_transform(loan_data['Property_Area'].astype(str))
#print(loan_data['Property_Area'])


loan_data = loan_data.drop('Loan_ID',axis=1)
loan_data = loan_data.fillna(loan_data.mean())

X = loan_data.drop('Loan_Status',axis=1)
Y = loan_data['Loan_Status']
Y = LabelEncoder().fit_transform(loan_data['Loan_Status'].astype(str))

X_train, X_test, y_train, y_test = train_test_split(X,Y,test_size=0.3,
random_state=1)

Naivebayes = GaussianNB()

b = Naivebayes.fit(X_train,y_train)
c = Naivebayes.predict(X_test)

print(c)

acc = sklearn.metrics.accuracy_score(y_test, c)
print(acc)
```

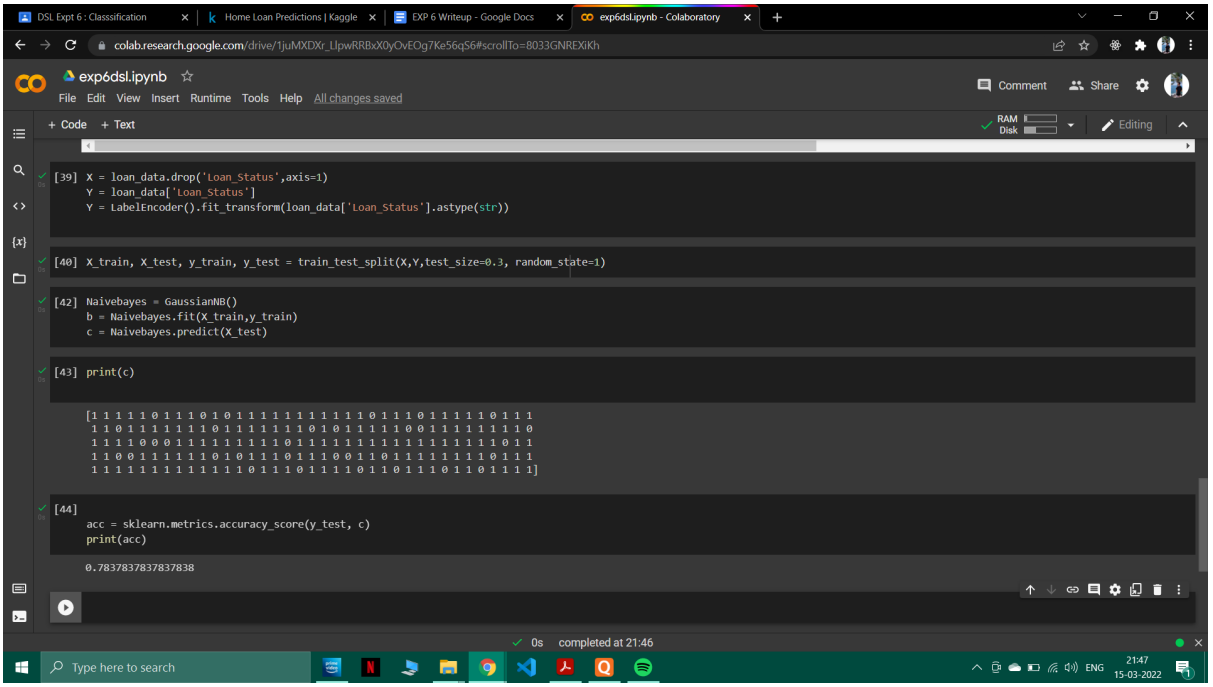
B. Paste Screenshots of above commands.


DSL Expt 6 : Classsification
Home Loan Predictions | Kaggle
EXP 6 Writeup - Google Docs
exp6dsl.ipynb - Colaboratory
colab.research.google.com/drive/1juMXDXr_LIpwRRBxX0yOvEOg7Ke56qS6#scrollTo=8033GNREXiKh
exp6dsl.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
Files
+ Code + Text
RAM
Disk
Editing
sample_data
Test_Loan_Home.csv
Train_Loan_Home.csv
[29] import sklearn
import pandas as pd
from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import GaussianNB
[30] loan_data = pd.read_csv('/content/Train_Loan_Home.csv')
[31] loan_data.columns.values
array(['Loan_ID', 'Gender', 'Married', 'Dependents', 'Education',
'Self_Employed', 'ApplicantIncome', 'CoapplicantIncome',
'LoanAmount', 'Loan_Amount_Term', 'Credit_History',
'Property_Area', 'Loan_Status'], dtype=object)
[32] loan_data['Gender']
0 Male
1 Male
2 Male
3 Male
4 Male
...
609 Female
610 Male
611 Male
612 Male
613 Female
Name: Gender, Length: 614, dtype: object
Disk
68.20 GB available
0s completed at 21:46
Type here to search
ENG
21:47
15-03-2022


DSL Expt 6 : Classsification
Home Loan Predictions | Kaggle
EXP 6 Writeup - Google Docs
exp6dsl.ipynb - Colaboratory
colab.research.google.com/drive/1juMXDXr_LIpwRRBxX0yOvEOg7Ke56qS6#scrollTo=8033GNREXiKh
exp6dsl.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
Files
+ Code + Text
RAM
Disk
Editing
sample_data
Test_Loan_Home.csv
Train_Loan_Home.csv
loan_data['Gender'] = LabelEncoder().fit_transform(loan_data['Gender'].astype(str))
loan_data['Gender']
0 1
1 1
2 1
3 1
4 1
..
609 0
610 1
611 1
612 1
613 0
Name: Gender, Length: 614, dtype: int64
[34] loan_data['Married'] = LabelEncoder().fit_transform(loan_data['Married'].astype(str))
loan_data['Married']
0 0
1 1
2 1
3 1
4 0
..
609 0
610 1
611 1
612 1
613 0
Name: Married, Length: 614, dtype: int64
OneDrive
Screenshot saved
The screenshot was added to your OneDrive.
Disk
68.20 GB available
0s completed at 21:46
Type here to search
ENG
21:47
15-03-2022


DSL Expt 6 : Classsification
Home Loan Predictions | Kaggle
EXP 6 Writeup - Google Docs
exp6dsl.ipynb - Colaboratory
colab.research.google.com/drive/1juMXDXr_LIpwRRBxX0yOvEOg7Ke56qS6#scrollTo=8033GNREXiKh
exp6dsl.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
Files
+ Code + Text
RAM
Disk
Editing
sample_data
Test_Loan_Home.csv
Train_Loan_Home.csv
loan_data['Education'] = LabelEncoder().fit_transform(loan_data['Education'].astype(str))
loan_data['Education']
0 0
1 0
2 0
3 1
4 0
609 0
610 0
611 0
612 0
613 0
Name: Education, Length: 614, dtype: int64
[36] loan_data['Self_Employed'] = LabelEncoder().fit_transform(loan_data['Self_Employed'].astype(str))
loan_data['Self_Employed']
0 0
1 0
2 1
3 0
4 0
609 0
610 0
611 0
612 0
613 1
Name: Self_Employed, Length: 614, dtype: int64
Disk 68.20 GB available
0s completed at 21:46
Type here to search
ENG
21:47
15-03-2022


DSL Expt 6 : Classsification
Home Loan Predictions | Kaggle
EXP 6 Writeup - Google Docs
exp6dsl.ipynb - Colaboratory
colab.research.google.com/drive/1juMXDXr_LIpwRRBxX0yOvEOg7Ke56qS6#scrollTo=8033GNREXiKh
exp6dsl.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
Files
+ Code + Text
RAM
Disk
Editing
sample_data
Test_Loan_Home.csv
Train_Loan_Home.csv
loan_data['Property_Area'] = LabelEncoder().fit_transform(loan_data['Property_Area'].astype(str))
loan_data['Property_Area']
0 2
1 0
2 2
3 2
4 2
609 0
610 0
611 2
612 2
613 1
Name: Property_Area, Length: 614, dtype: int64
[38] loan_data = loan_data.drop('Loan_ID',axis=1)
loan_data = loan_data.fillna(loan_data.mean())
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:2: FutureWarning: Dropping of nuisance columns in DataFrame reductions (with 'numeri
X = loan_data.drop('Loan_Status',axis=1)
Y = loan_data['Loan_Status']
Y = LabelEncoder().fit_transform(loan_data['Loan_Status'].astype(str))
[40] X_train, X_test, y_train, y_test = train_test_split(X,Y,test_size=0.3, random_state=1)
Disk 68.20 GB available
0s completed at 21:46
Type here to search
ENG
21:47
15-03-2022



8. Post-Experiments Exercise

A. Extended Theory: (Soft Copy)

- Types of classification.

1. Binary Classification: Typically, binary classification tasks involve one class that is the normal state and another class that is the abnormal state. For example “not spam” is the normal state and “spam” is the abnormal state. The class for the normal state is assigned the class label 0 and the class with the abnormal state is assigned the class label 1.

2. Multi-Class Classification: Multi-class classification refers to those classification tasks that have more than two class labels. Unlike binary classification, multi-class classification does not have the notion of normal and abnormal outcomes. Instead, examples are classified as belonging to one among a range of known classes.
Examples include: Face classification, Optical character recognition

3. Multi-Label Classification: Multi-label classification refers to those classification tasks that have two or more class labels, where one or more class labels may be predicted for each example. Consider the example of photo classification, where a given photo may have multiple objects in the scene and a model may predict the presence of multiple known objects in the photo, such as “bicycle,” “apple,”

“person,” etc.

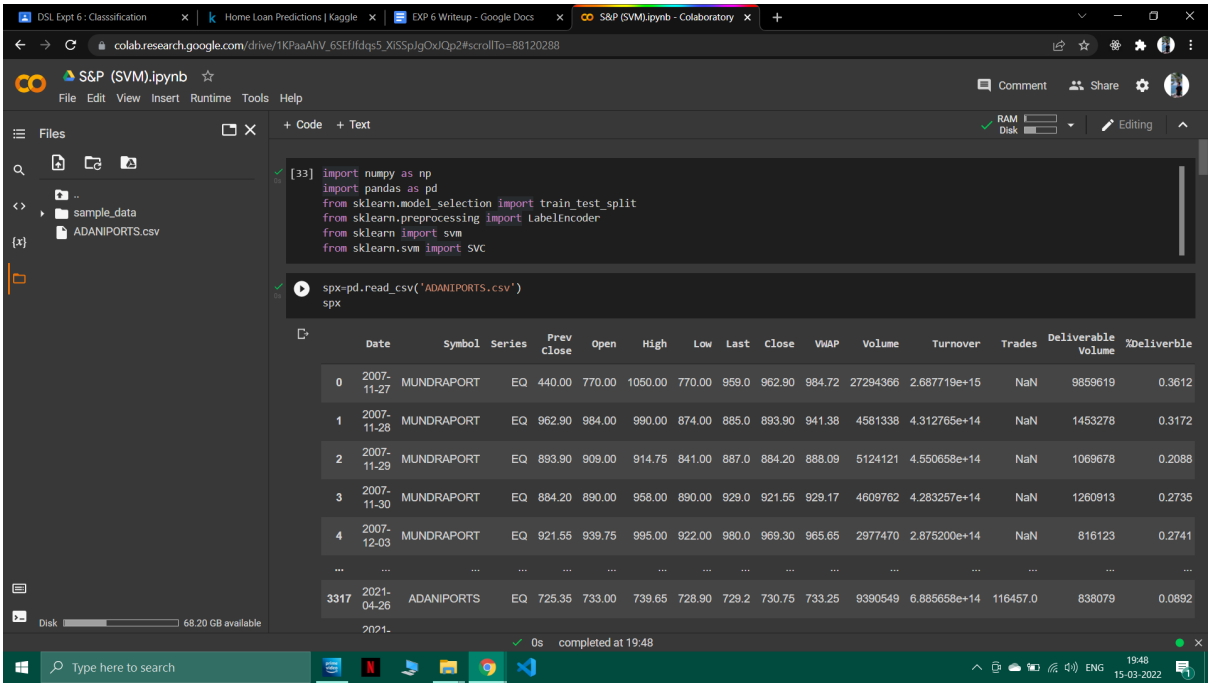
4. Imbalanced Classification: Imbalanced classification refers to classification tasks where the number of examples in each class is unequally distributed. Typically, imbalanced classification tasks are binary classification tasks where the majority of examples in the training dataset belong to the normal class and a minority of examples belong to the abnormal class.

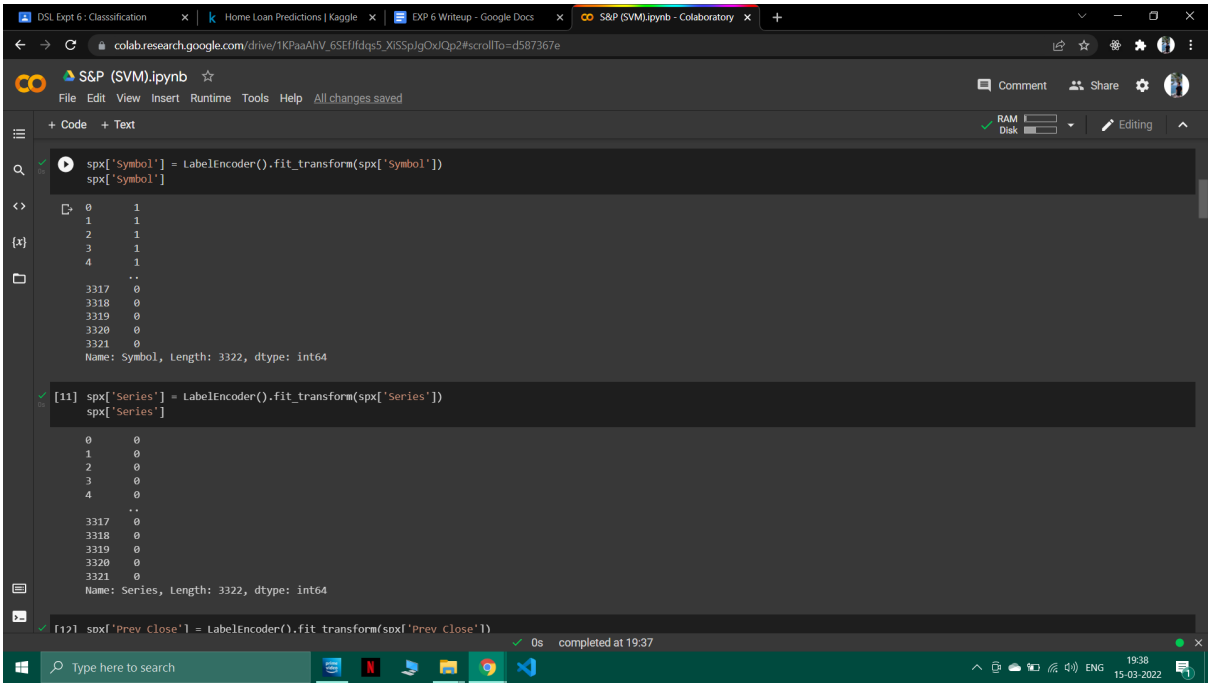
Examples include: Fraud Detection, Outlier Detection

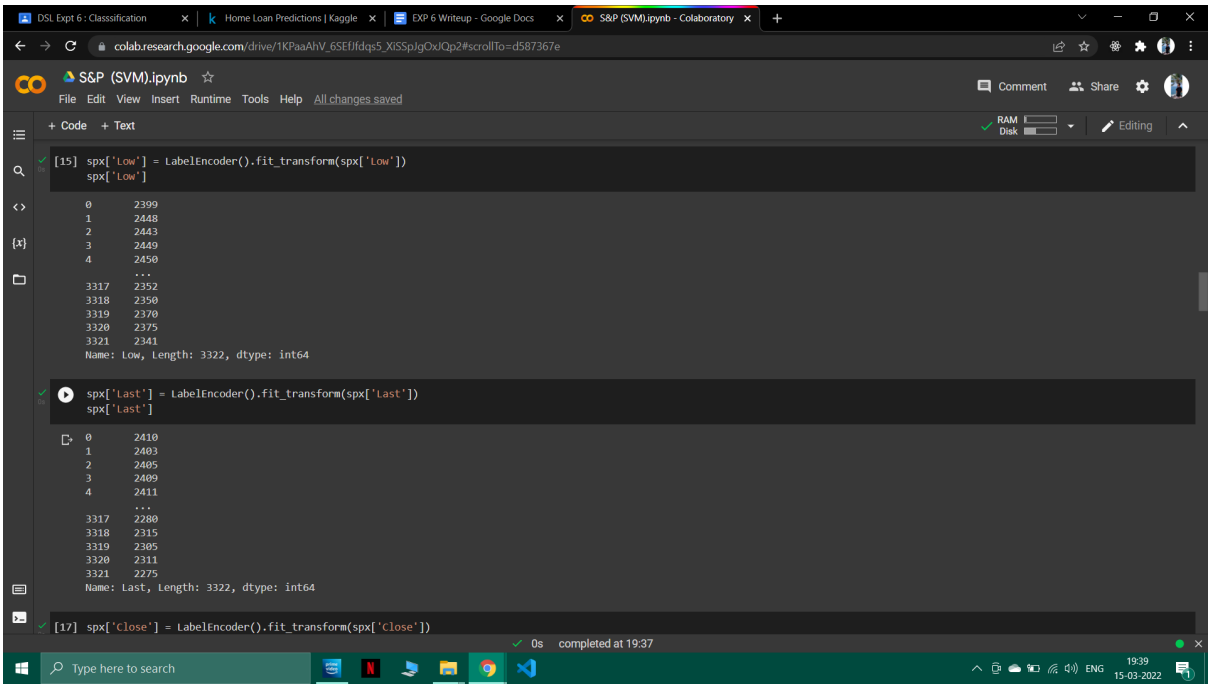
B. Questions:

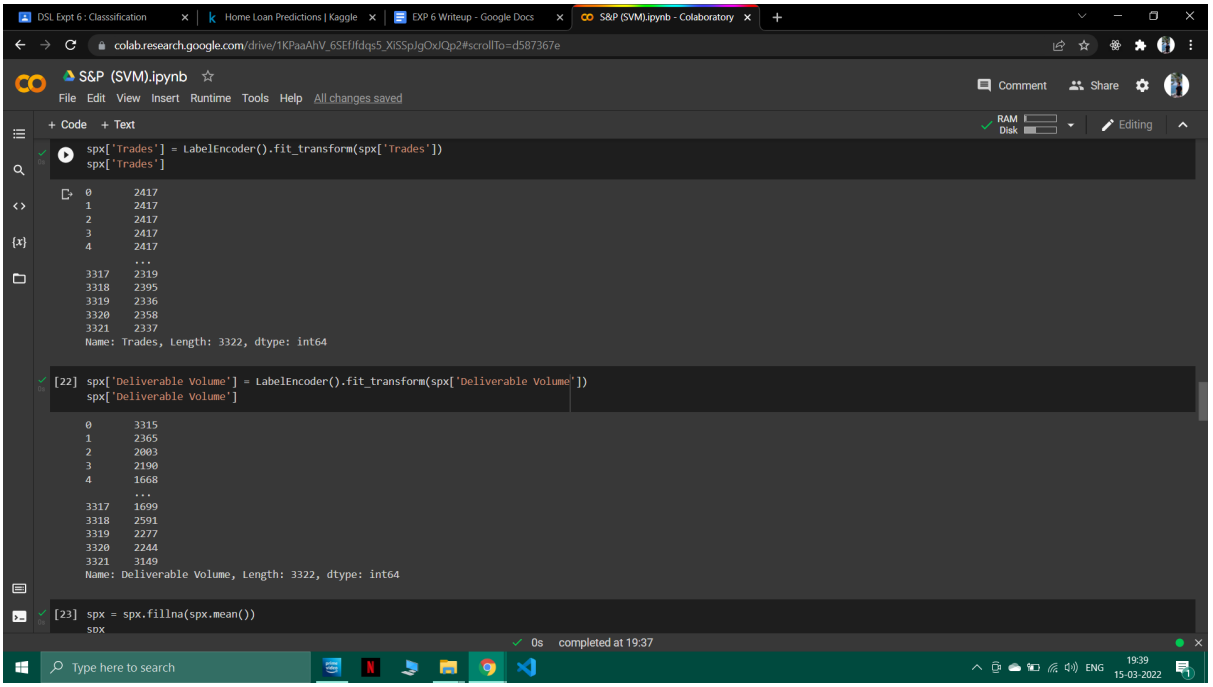
- Compare S&P500 dataset with other classifiers SVM and KNN.

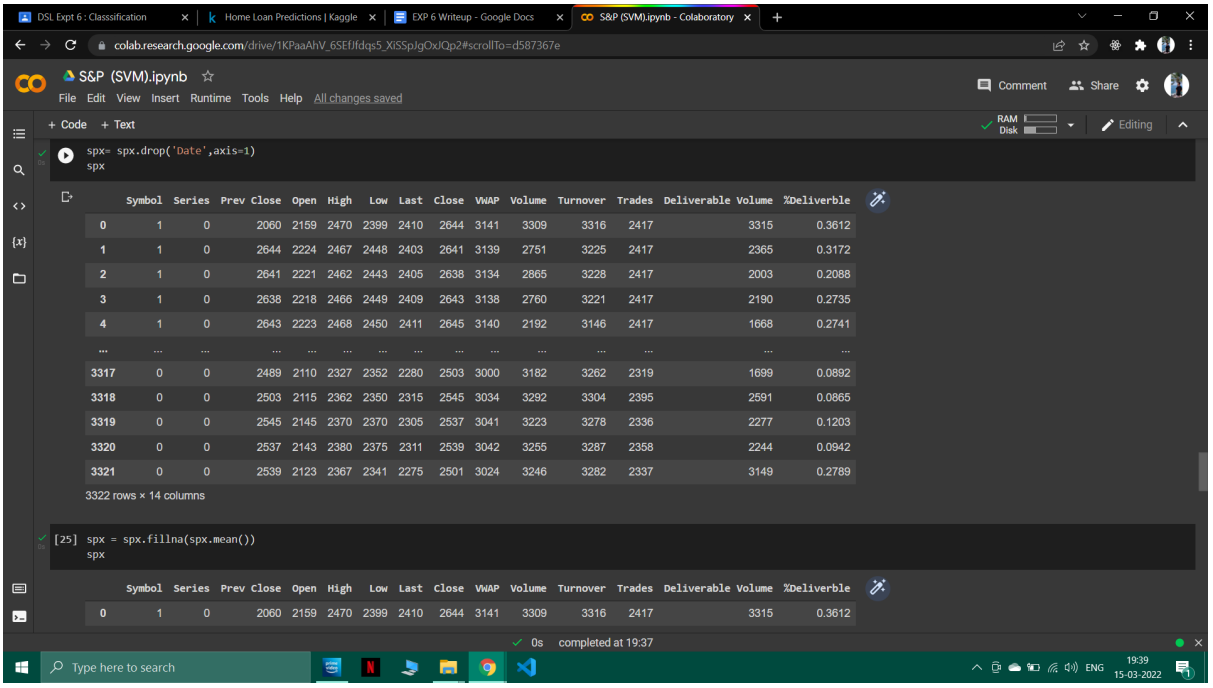
SVM:

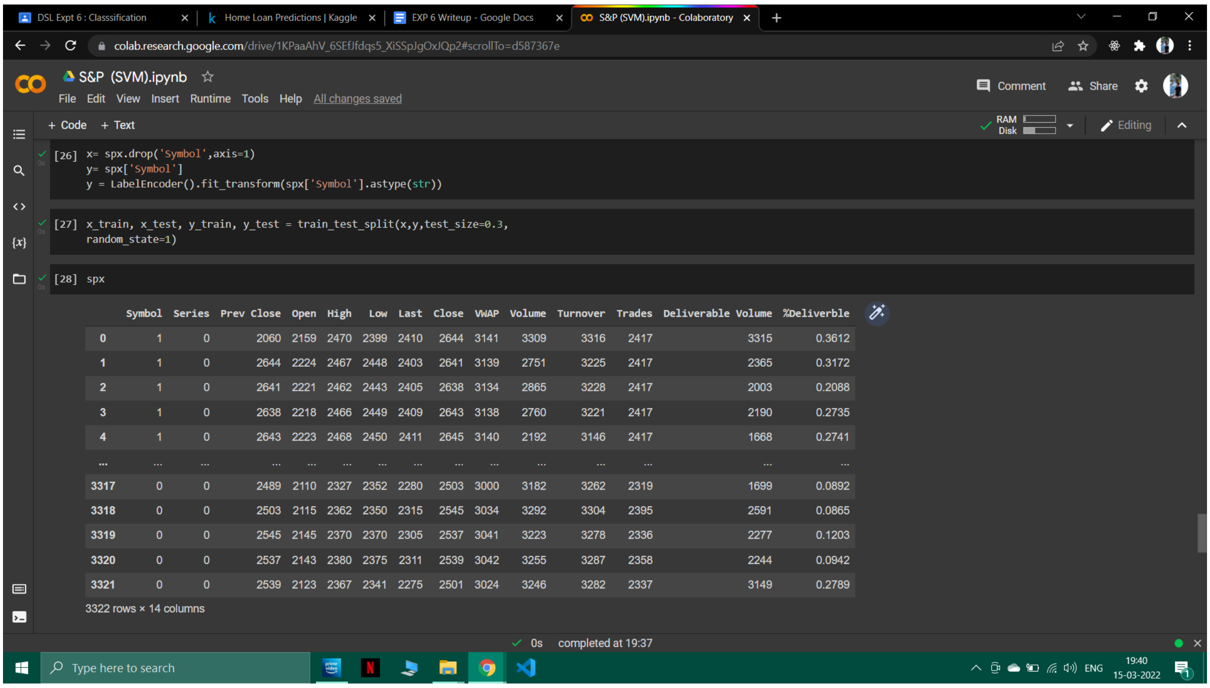



DSL Expt 6 : Classsification
Home Loan Predictions | Kaggle
EXP 6 Writeup - Google Docs
S&P (SVM).ipynb - Colaboratory
colab.research.google.com/drive/1KPaaAhV_6SEfJfdqs5_XiSSpJgOxJQp2#scrollTo=d587367e
S&P (SVM).ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
+ Code + Text
RAM
Disk
Editing
spx['Symbol'] = LabelEncoder().fit_transform(spx['Symbol'])
spx['Symbol']
0 1
1 1
2 1
3 1
4 1
..
3317 0
3318 0
3319 0
3320 0
3321 0
Name: Symbol, Length: 3322, dtype: int64
[11] spx['Series'] = LabelEncoder().fit_transform(spx['Series'])
spx['Series']
0 0
1 0
2 0
3 0
4 0
..
3317 0
3318 0
3319 0
3320 0
3321 0
Name: Series, Length: 3322, dtype: int64
0s completed at 19:37
Type here to search
ENG
19:38
15-03-2022


DSL Expt 6 : Classsification
Home Loan Predictions | Kaggle
EXP 6 Writeup - Google Docs
S&P (SVM).ipynb - Colaboratory
colab.research.google.com/drive/1KPaaAhV_6SEfJfdqs5_XiSSpJgOxJQp2#scrollTo=d587367e
S&P (SVM).ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
+ Code + Text
RAM
Disk
Editing
[15] spx['Low'] = LabelEncoder().fit_transform(spx['Low'])
spx['Low']
0 2399
1 2448
2 2443
3 2449
4 2450
...
3317 2352
3318 2350
3319 2370
3320 2375
3321 2341
Name: Low, Length: 3322, dtype: int64
spx['Last'] = LabelEncoder().fit_transform(spx['Last'])
spx['Last']
0 2410
1 2403
2 2405
3 2409
4 2411
...
3317 2280
3318 2315
3319 2305
3320 2311
3321 2275
Name: Last, Length: 3322, dtype: int64
[17] spx['Close'] = LabelEncoder().fit_transform(spx['Close'])
0s completed at 19:37
Type here to search
ENG
19:39
15-03-2022


DSL Expt 6 : Classsification
Home Loan Predictions | Kaggle
EXP 6 Writeup - Google Docs
S&P (SVM).ipynb - Colaboratory
colab.research.google.com/drive/1KPaaAhV_6SEfJfdqs5_XiSSpJgOxJQp2#scrollTo=d587367e
S&P (SVM).ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
+ Code + Text
RAM
Disk
Editing
spx['Trades'] = LabelEncoder().fit_transform(spx['Trades'])
spx['Trades']
0 2417
1 2417
2 2417
3 2417
4 2417
...
3317 2319
3318 2395
3319 2336
3320 2358
3321 2337
Name: Trades, Length: 3322, dtype: int64
[22] spx['Deliverable Volume'] = LabelEncoder().fit_transform(spx['Deliverable Volume'])
spx['Deliverable Volume']
0 3315
1 2365
2 2003
3 2190
4 1668
...
3317 1699
3318 2591
3319 2277
3320 2244
3321 3149
Name: Deliverable Volume, Length: 3322, dtype: int64
[23] spx = spx.fillna(spx.mean())
spx
0s completed at 19:37
Type here to search
ENG
19:39
15-03-2022


DSL Expt 6 : Classsification
Home Loan Predictions | Kaggle
EXP 6 Writeup - Google Docs
S&P (SVM).ipynb - Colaboratory
colab.research.google.com/drive/1KPaaAhV_6SEfJfdqs5_XiSSpJgOxJQp2#scrollTo=d587367e
S&P (SVM).ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
+ Code + Text
RAM
Disk
Editing
spx= spx.drop('Date',axis=1)
spx
Symbol Series Prev Close Open High Low Last Close VWAP Volume Turnover Trades Deliverable Volume %Deliverble
0 1 0 2060 2159 2470 2399 2410 2644 3141 3309 3316 2417 3315 0.3612
1 1 0 2644 2224 2467 2448 2403 2641 3139 2751 3225 2417 2365 0.3172
2 1 0 2641 2221 2462 2443 2405 2638 3134 2865 3228 2417 2003 0.2088
3 1 0 2638 2218 2466 2449 2409 2643 3138 2760 3221 2417 2190 0.2735
4 1 0 2643 2223 2468 2450 2411 2645 3140 2192 3146 2417 1668 0.2741
... ...
3317 0 0 2489 2110 2327 2352 2280 2503 3000 3182 3262 2319 1699 0.0892
3318 0 0 2503 2115 2362 2350 2315 2545 3034 3292 3304 2395 2591 0.0865
3319 0 0 2545 2145 2370 2370 2305 2537 3041 3223 3278 2336 2277 0.1203
3320 0 0 2537 2143 2380 2375 2311 2539 3042 3255 3287 2358 2244 0.0942
3321 0 0 2539 2123 2367 2341 2275 2501 3024 3246 3282 2337 3149 0.2789
3322 rows × 14 columns
[25] spx = spx.fillna(spx.mean())
spx
Symbol Series Prev Close Open High Low Last Close VWAP Volume Turnover Trades Deliverable Volume %Deliverble
0 1 0 2060 2159 2470 2399 2410 2644 3141 3309 3316 2417 3315 0.3612
0s completed at 19:37
Type here to search
ENG
19:39
15-03-2022

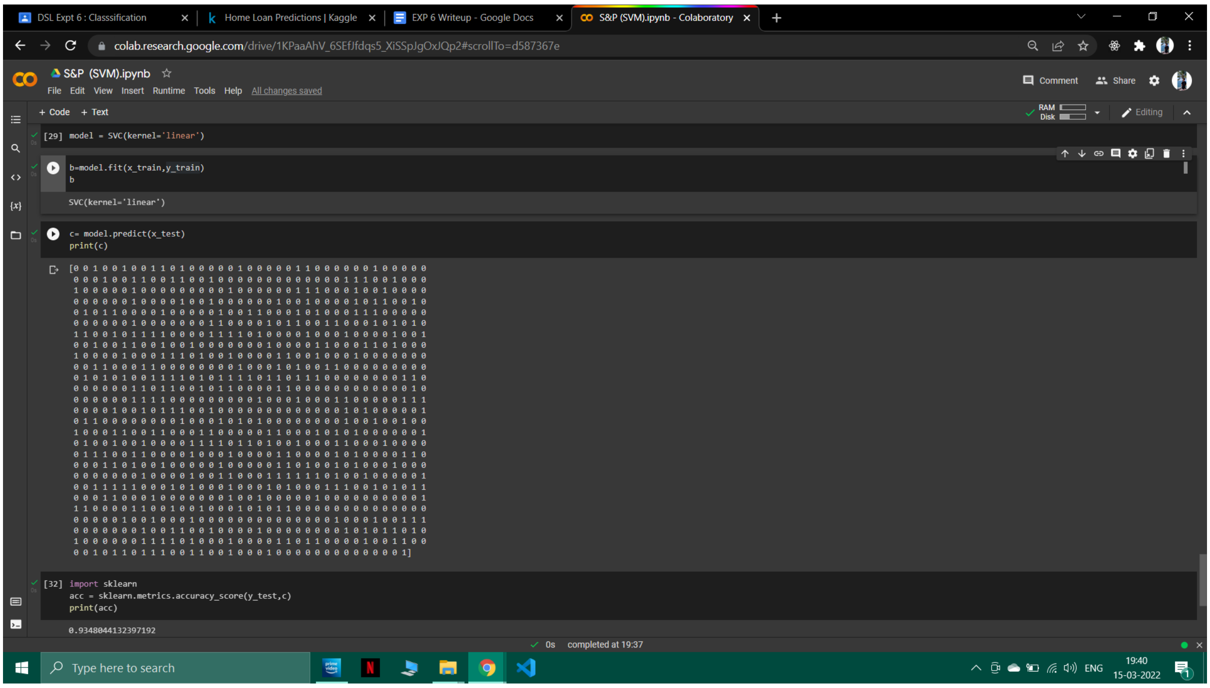


S&P (SVM).ipynb

```
[26] x= spx.drop('Symbol',axis=1)
     y= spx['Symbol']
     y = LabelEncoder().fit_transform(spx['Symbol'].astype(str))

[27] x_train, x_test, y_train, y_test = train_test_split(x,y,test_size=0.3,
     random_state=1)

[28] spx
```

	Symbol	Series	Prev Close	Open	High	Low	Last	Close	VWAP	Volume	Turnover	Trades	Deliverable Volume	%Deliverble
0	1	0	2060	2159	2470	2399	2410	2644	3141	3309	3316	2417	3315	0.3612
1	1	0	2644	2224	2467	2448	2403	2641	3139	2751	3225	2417	2365	0.3172
2	1	0	2641	2221	2462	2443	2405	2638	3134	2865	3228	2417	2003	0.2088
3	1	0	2638	2218	2466	2449	2409	2643	3138	2760	3221	2417	2190	0.2735
4	1	0	2643	2223	2468	2450	2411	2645	3140	2192	3146	2417	1668	0.2741
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
3317	0	0	2489	2110	2327	2352	2280	2503	3000	3182	3262	2319	1699	0.0892
3318	0	0	2503	2115	2362	2350	2315	2545	3034	3292	3304	2395	2591	0.0865
3319	0	0	2545	2145	2370	2370	2305	2537	3041	3223	3278	2336	2277	0.1203
3320	0	0	2537	2143	2380	2375	2311	2539	3042	3255	3287	2358	2244	0.0942
3321	0	0	2539	2123	2367	2341	2275	2501	3024	3246	3282	2337	3149	0.2789

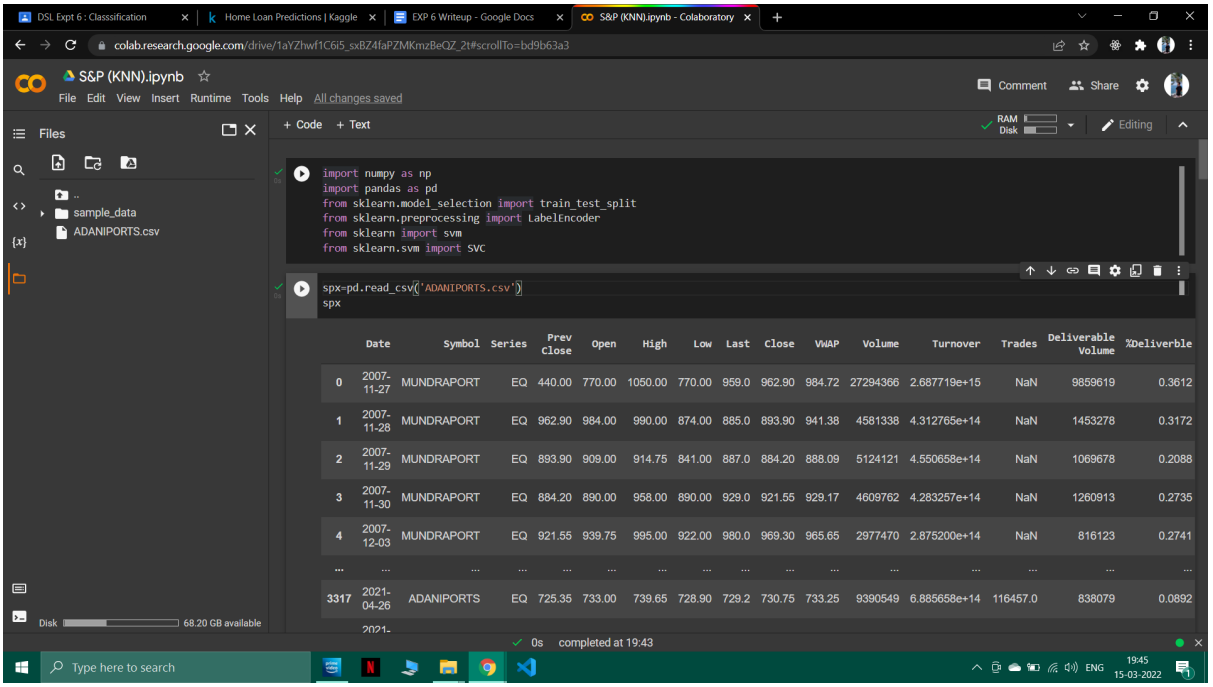
3322 rows × 14 columns



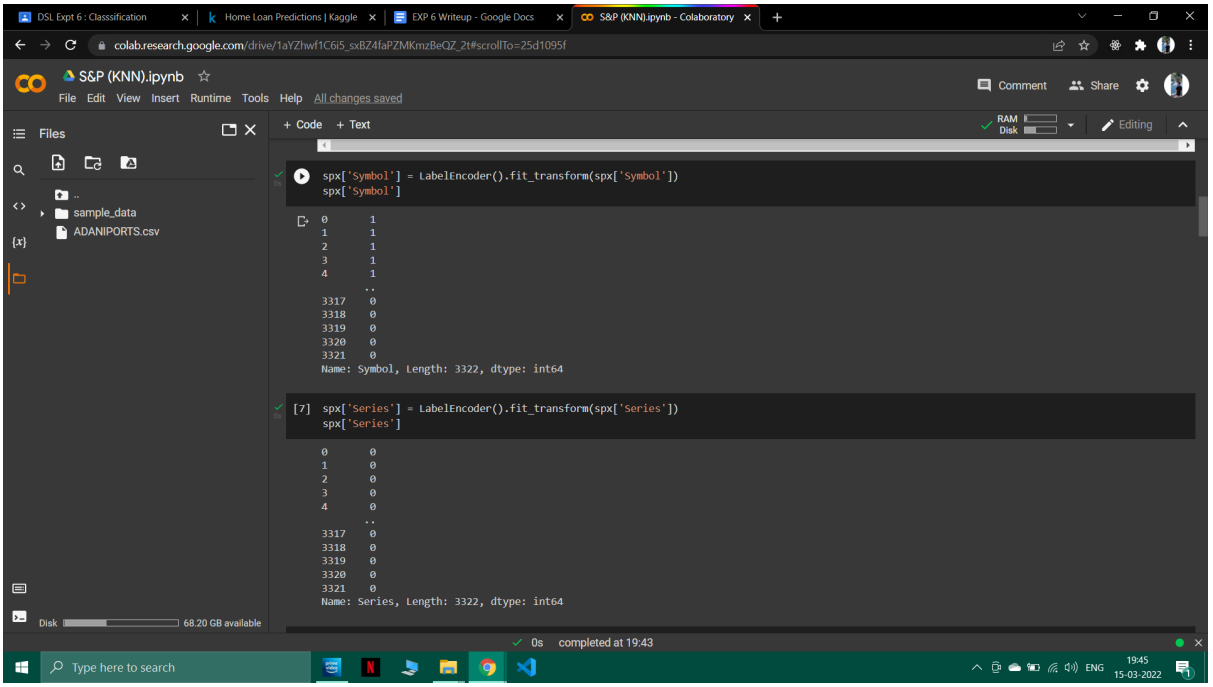
S&P (SVM).ipynb

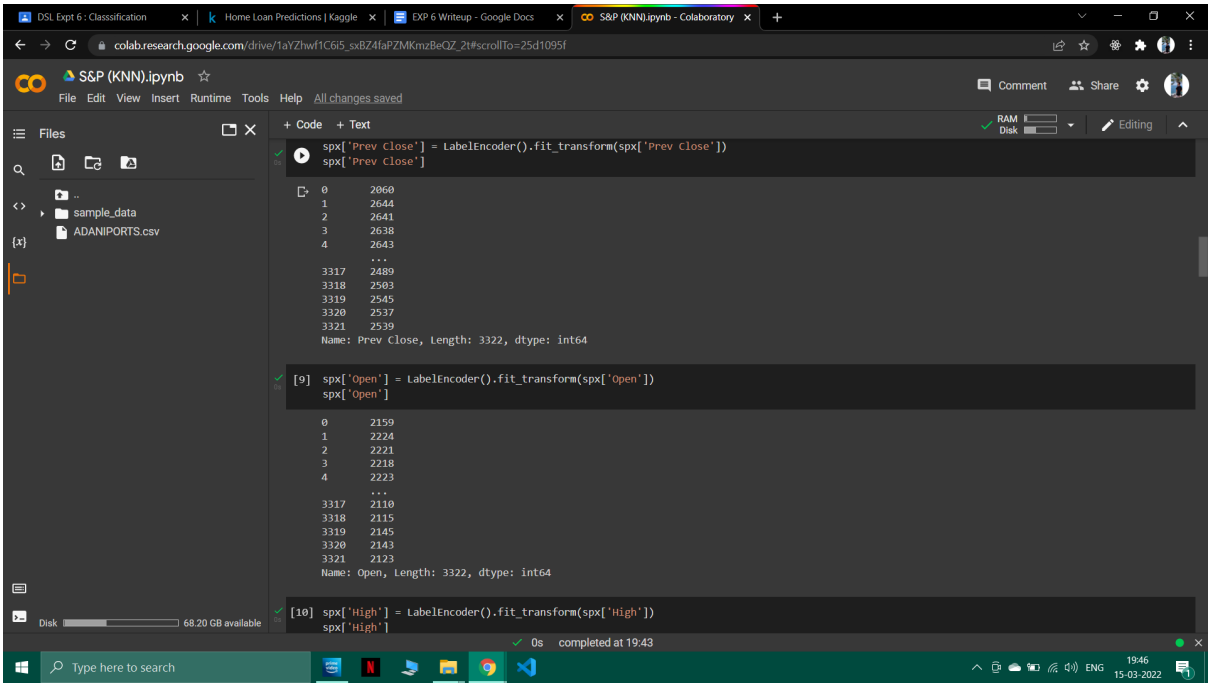
```
[29] model = SVC(kernel='linear')

b=model.fit(x_train,y_train)
b

SVC(kernel='linear')

c= model.predict(x_test)
print(c)

[0 0 1 0 0 1 0 0 1 1 0 1 0 0 0 0 0 1 0 0 0 0 0 1 1 0 0 0 0 0 0 1 0 0 0 0 0
 0 0 0 1 0 0 1 1 0 0 1 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 1 0 0 0
 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 1 1 1 0 0 0 1 0 0 1 0 0 0 0
 0 0 0 0 0 0 1 0 0 0 0 1 0 0 1 0 0 0 0 0 1 0 0 1 0 0 0 0 1 0 1 1 0 0 1 0
 0 1 0 1 1 0 0 0 0 1 0 0 0 0 0 1 0 0 1 1 0 0 0 1 0 1 0 0 0 1 1 1 0 0 0 0 0
 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 1 0 0 0 0 1 0 1 1 0 0 1 1 0 0 0 1 0 1 0 1 0
 1 1 0 0 1 0 1 1 1 1 0 0 0 0 1 1 1 1 0 1 0 0 0 0 1 0 0 0 1 0 0 0 0 1 0 0 1
 0 0 1 0 0 1 1 0 0 1 0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 1 1 0 0 0 1 1 0 1 0 0 0
 1 0 0 0 0 1 0 0 0 1 1 1 0 1 0 0 1 0 0 0 0 1 1 0 0 1 0 0 0 1 0 0 0 0 0 0 0
 0 0 1 1 0 0 0 1 1 0 0 0 0 0 0 0 1 0 0 0 1 0 1 0 0 1 1 0 0 0 0 0 0 0 0
 0 1 0 1 0 1 0 0 1 1 1 1 0 1 0 1 1 1 1 0 1 1 0 1 1 1 0 0 0 0 0 0 0 0 1 1 0
 0 0 0 0 0 0 1 1 0 1 1 0 0 1 0 1 1 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 1 0
 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0 1 1 0 0 0 0 0 1 1 1
 0 0 0 0 1 0 0 1 0 1 1 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0 0 0 0 0 1
 0 1 1 0 0 0 0 0 0 0 1 0 0 0 1 0 1 0 1 0 0 0 0 0 0 0 0 1 0 0 1 0 0 1 0 0
 1 0 0 0 1 1 0 0 1 1 0 0 0 1 1 0 0 0 0 0 1 1 0 0 0 1 0 1 0 1 0 0 0 0 0 0 1
 0 1 0 0 1 0 0 1 0 0 0 0 1 1 1 1 0 1 1 0 1 0 0 1 0 0 0 1 1 0 0 0 1 0 0 0 0
 0 1 1 1 0 0 1 1 0 0 0 0 1 0 0 0 1 0 0 0 0 1 1 0 0 0 0 1 0 1 0 0 0 0 1 1 0
 0 0 0 1 1 0 1 0 0 1 0 0 0 0 0 1 0 0 0 0 0 1 1 0 1 0 0 1 0 1 0 0 0 1 0 0 0
 0 0 0 0 0 0 0 1 0 0 0 0 1 0 0 1 1 0 0 0 1 1 1 1 1 1 0 1 0 0 1 0 0 0 0 0 1
 0 0 1 1 1 1 1 0 0 0 1 0 1 0 0 0 1 0 0 0 1 0 1 0 0 0 1 1 1 0 0 1 0 1 0 1 1
 0 0 0 1 1 0 0 0 1 0 0 0 0 0 0 0 1 0 0 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 1
 1 1 0 0 0 0 1 1 0 0 1 0 0 1 0 0 0 1 0 1 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 1 0 0 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 1 1 1
 0 0 0 0 0 0 0 1 0 0 1 1 0 0 1 0 0 0 0 1 0 0 0 0 0 0 0 0 1 0 1 0 1 1 0 1 0
 1 0 0 0 0 0 0 1 1 1 1 0 1 0 0 0 1 0 0 0 0 1 1 0 1 1 0 0 0 0 1 0 0 1 1 0 0
 0 0 1 0 1 1 0 1 1 1 0 0 1 1 0 0 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1]

[32] import sklearn
     acc = sklearn.metrics.accuracy_score(y_test,c)
     print(acc)

0.9348044132397192
```

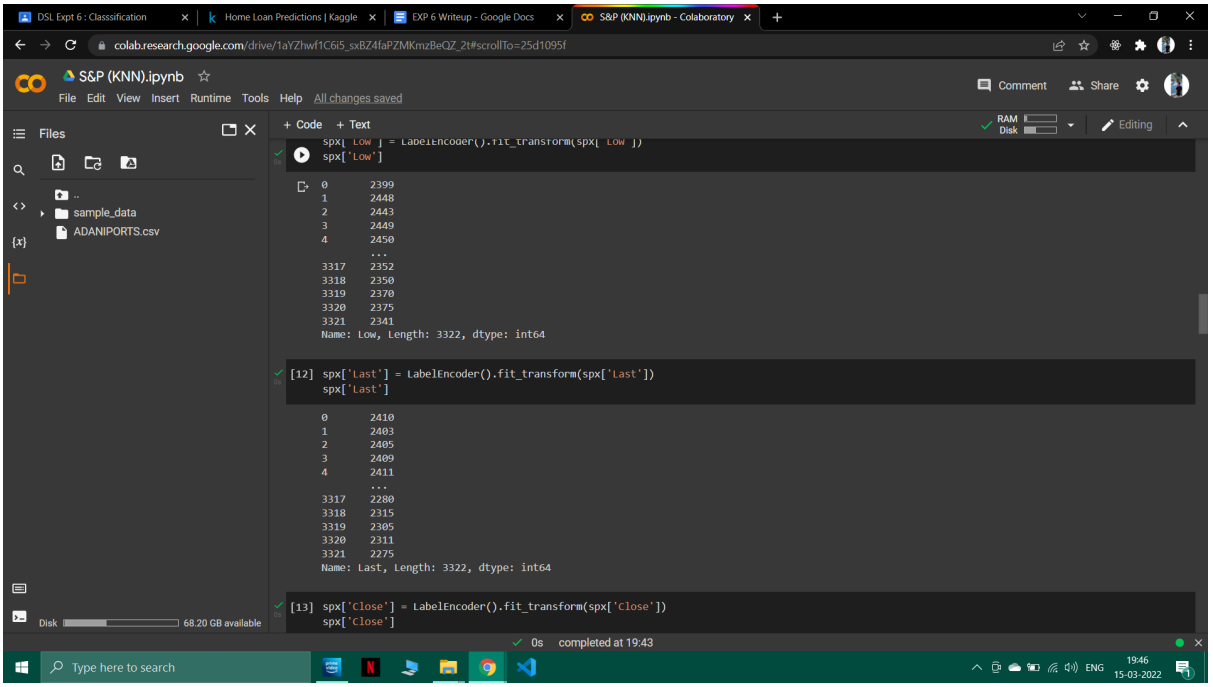
KNN:

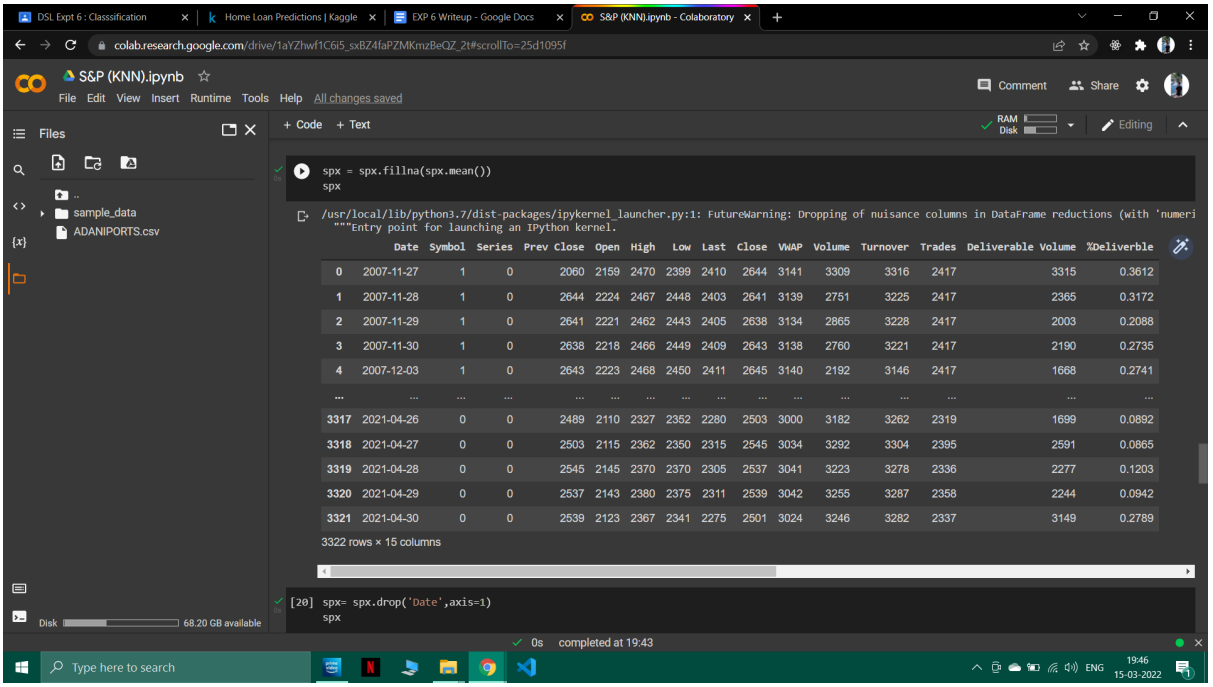


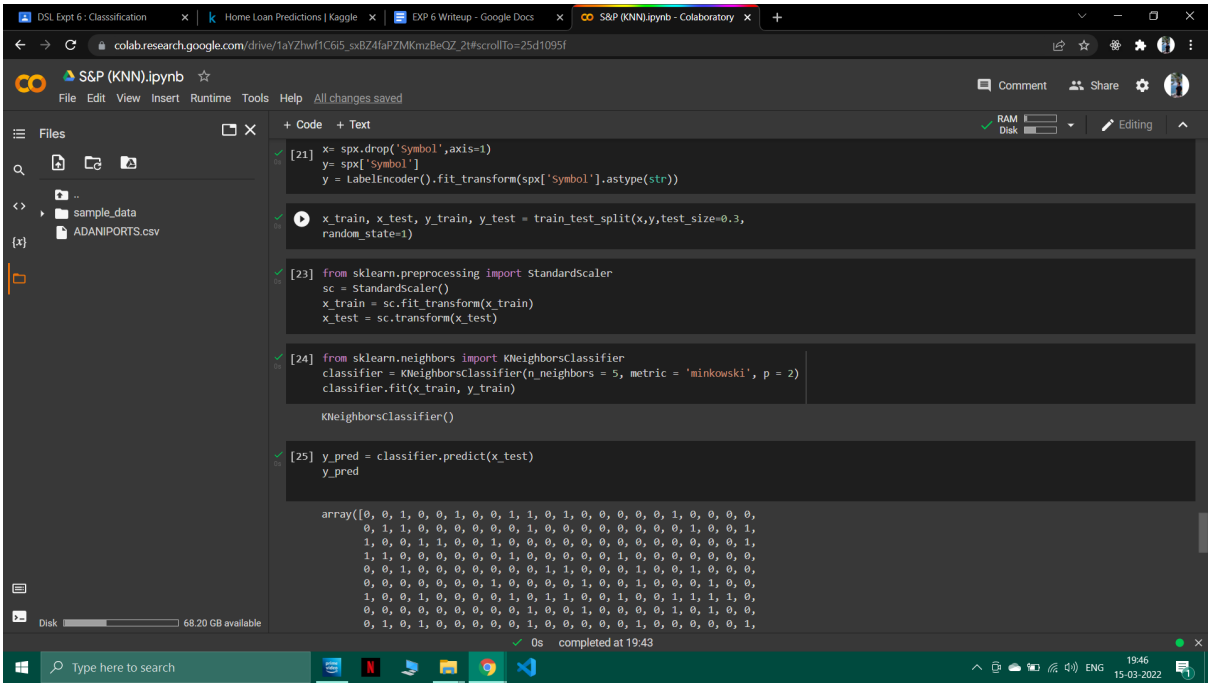
	Date	Symbol	Series	Prev Close	Open	High	Low	Last	Close	VWAP	Volume	Turnover	Trades	Deliverable Volume	%Deliverble
0	2007-11-27	MUNDRAPORT	EQ	440.00	770.00	1050.00	770.00	959.0	962.90	984.72	27294366	2.687719e+15	NaN	9859619	0.3612
1	2007-11-28	MUNDRAPORT	EQ	962.90	984.00	990.00	874.00	885.0	893.90	941.38	4581338	4.312765e+14	NaN	1453278	0.3172
2	2007-11-29	MUNDRAPORT	EQ	893.90	909.00	914.75	841.00	887.0	884.20	888.09	5124121	4.550658e+14	NaN	1069678	0.2088
3	2007-11-30	MUNDRAPORT	EQ	884.20	890.00	958.00	890.00	929.0	921.55	929.17	4609762	4.283257e+14	NaN	1260913	0.2735
4	2007-12-03	MUNDRAPORT	EQ	921.55	939.75	995.00	922.00	980.0	969.30	965.65	2977470	2.875200e+14	NaN	816123	0.2741
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
3317	2021-04-26	ADANIPORTS	EQ	725.35	733.00	739.65	728.90	729.2	730.75	733.25	9390549	6.885658e+14	116457.0	838079	0.0892
	2021-														

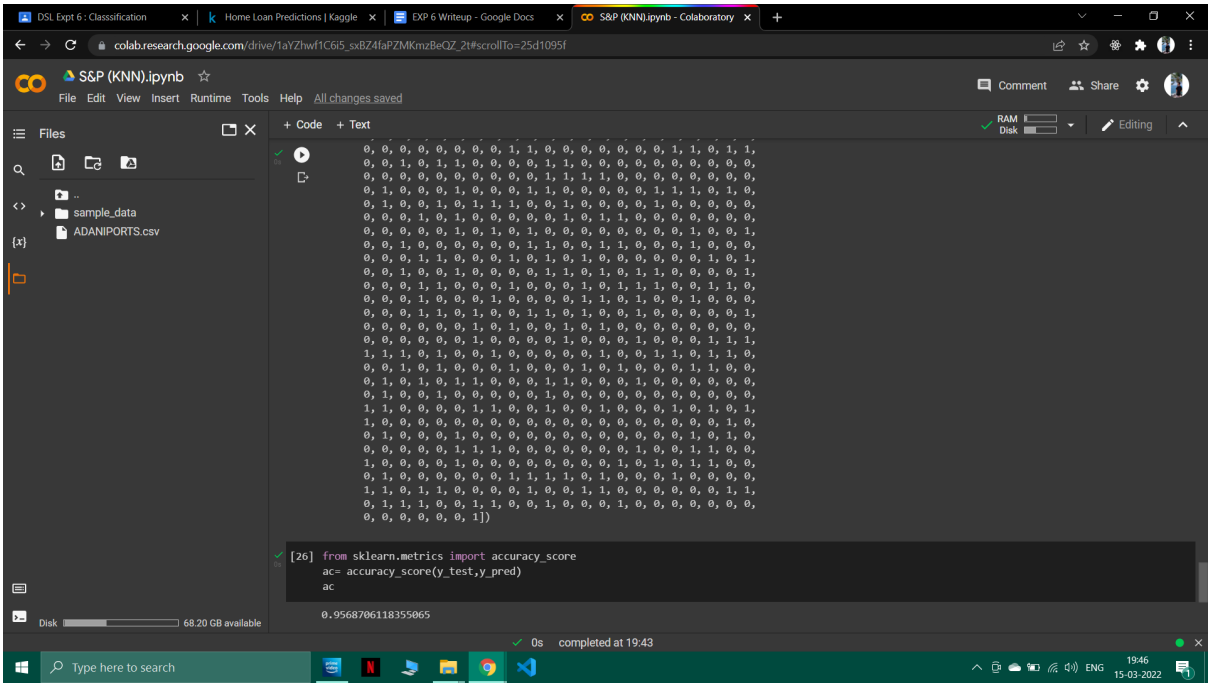



DSL Expt 6 : Classsification
Home Loan Predictions | Kaggle
EXP 6 Writeup - Google Docs
S&P (KNN).ipynb - Colaboratory
colab.research.google.com/drive/1aYZhwf1C6i5_sxBZ4faPZMKmzBeQZ_2t#scrollTo=25d1095f
S&P (KNN).ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
Files
sample_data
ADANIPORTS.csv
+ Code + Text
RAM
Disk
Editing
spx['Prev Close'] = LabelEncoder().fit_transform(spx['Prev Close'])
spx['Prev Close']
0 2060
1 2644
2 2641
3 2638
4 2643
...
3317 2489
3318 2503
3319 2545
3320 2537
3321 2539
Name: Prev Close, Length: 3322, dtype: int64
[9] spx['Open'] = LabelEncoder().fit_transform(spx['Open'])
spx['Open']
0 2159
1 2224
2 2221
3 2218
4 2223
...
3317 2110
3318 2115
3319 2145
3320 2143
3321 2123
Name: Open, Length: 3322, dtype: int64
[10] spx['High'] = LabelEncoder().fit_transform(spx['High'])
spx['High']
Disk 68.20 GB available
0s completed at 19:43
Type here to search
ENG
19:46
15-03-2022


DSL Expt 6 : Classsification
Home Loan Predictions | Kaggle
EXP 6 Writeup - Google Docs
S&P (KNN).ipynb - Colaboratory
colab.research.google.com/drive/1aYZhwf1C6i5_sxBZ4faPZMKmzBeQZ_2t#scrollTo=25d1095f
S&P (KNN).ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
Files
sample_data
ADANIPORTS.csv
+ Code + Text
RAM
Disk
Editing
spx['Low'] = LabelEncoder().fit_transform(spx['Low'])
spx['Low']
0 2399
1 2448
2 2443
3 2449
4 2450
...
3317 2352
3318 2350
3319 2370
3320 2375
3321 2341
Name: Low, Length: 3322, dtype: int64
[12] spx['Last'] = LabelEncoder().fit_transform(spx['Last'])
spx['Last']
0 2410
1 2403
2 2405
3 2409
4 2411
...
3317 2280
3318 2315
3319 2305
3320 2311
3321 2275
Name: Last, Length: 3322, dtype: int64
[13] spx['Close'] = LabelEncoder().fit_transform(spx['Close'])
spx['Close']
Disk 68.20 GB available
0s completed at 19:43
Type here to search
ENG
19:46
15-03-2022

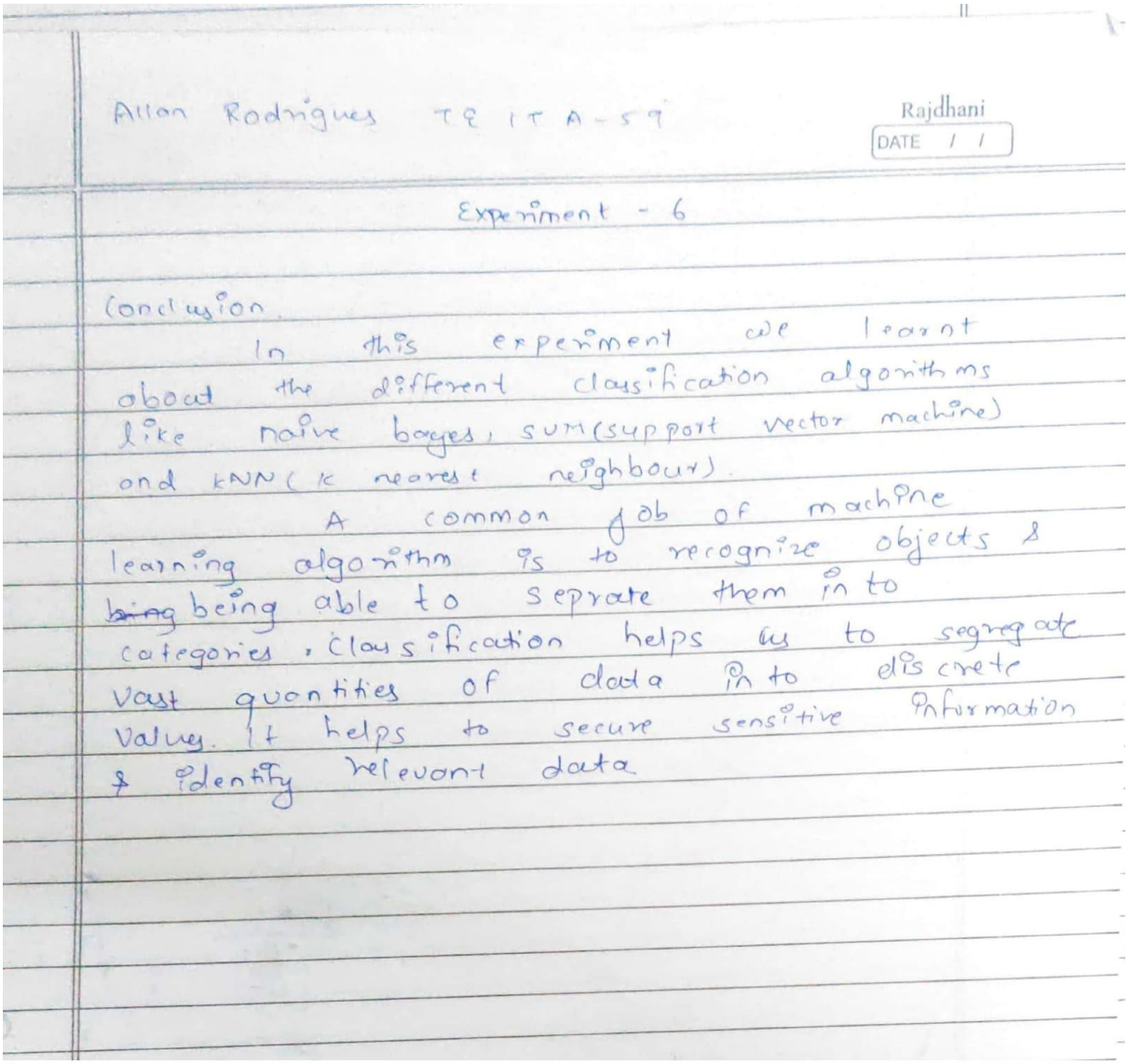

DSL Expt 6 : Classsification
Home Loan Predictions | Kaggle
EXP 6 Writeup - Google Docs
S&P (KNN).ipynb - Colaboratory
colab.research.google.com/drive/1aYZhwf1C6i5_sxBZ4faPZMKmzBeQZ_2t#scrollTo=25d1095f
S&P (KNN).ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
Files
sample_data
ADANIPORTS.csv
+ Code + Text
RAM
Disk
Editing
spx = spx.fillna(spx.mean())
spx
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:1: FutureWarning: Dropping of nuisance columns in DataFrame reductions (with 'numeri
"""Entry point for launching an IPython kernel.
Date Symbol Series Prev Close Open High Low Last Close VWAP Volume Turnover Trades Deliverable Volume %Deliverble
0 2007-11-27 1 0 2060 2159 2470 2399 2410 2644 3141 3309 3316 2417 3315 0.3612
1 2007-11-28 1 0 2644 2224 2467 2448 2403 2641 3139 2751 3225 2417 2365 0.3172
2 2007-11-29 1 0 2641 2221 2462 2443 2405 2638 3134 2865 3228 2417 2003 0.2088
3 2007-11-30 1 0 2638 2218 2466 2449 2409 2643 3138 2760 3221 2417 2190 0.2735
4 2007-12-03 1 0 2643 2223 2468 2450 2411 2645 3140 2192 3146 2417 1668 0.2741
3317 2021-04-26 0 0 2489 2110 2327 2352 2280 2503 3000 3182 3262 2319 1699 0.0892
3318 2021-04-27 0 0 2503 2115 2362 2350 2315 2545 3034 3292 3304 2395 2591 0.0865
3319 2021-04-28 0 0 2545 2145 2370 2370 2305 2537 3041 3223 3278 2336 2277 0.1203
3320 2021-04-29 0 0 2537 2143 2380 2375 2311 2539 3042 3255 3287 2358 2244 0.0942
3321 2021-04-30 0 0 2539 2123 2367 2341 2275 2501 3024 3246 3282 2337 3149 0.2789
3322 rows × 15 columns
[20] spx= spx.drop('Date',axis=1)
spx
Disk 68.20 GB available
0s completed at 19:43
Type here to search
ENG
19:46
15-03-2022


DSL Expt 6 : Classsification
Home Loan Predictions | Kaggle
EXP 6 Writeup - Google Docs
S&P (KNN).ipynb - Colaboratory
colab.research.google.com/drive/1aYZhwf1C6i5_sxBZ4faPZMKmzBeQZ_2t#scrollTo=25d1095f
S&P (KNN).ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
Files
sample_data
ADANIPORTS.csv
+ Code + Text
RAM
Disk
Editing
[21] x= spx.drop('Symbol',axis=1)
y= spx['Symbol']
y = LabelEncoder().fit_transform(spx['Symbol'].astype(str))
x_train, x_test, y_train, y_test = train_test_split(x,y,test_size=0.3,
random_state=1)
[23] from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
x_train = sc.fit_transform(x_train)
x_test = sc.transform(x_test)
[24] from sklearn.neighbors import KNeighborsClassifier
classifier = KNeighborsClassifier(n_neighbors = 5, metric = 'minkowski', p = 2)
classifier.fit(x_train, y_train)
KNeighborsClassifier()
[25] y_pred = classifier.predict(x_test)
y_pred
array([0, 0, 1, 0, 0, 1, 0, 0, 1, 1, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0,
0, 1, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1,
1, 0, 0, 1, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1,
1, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0,
0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 1, 0, 0,
1, 0, 0, 1, 0, 0, 0, 0, 1, 0, 1, 1, 0, 0, 1, 0, 0, 1, 1, 1, 1, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 1, 0, 1, 0, 1, 0,
0, 1, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1,
Disk 68.20 GB available
0s completed at 19:43
Type here to search
ENG
19:46
15-03-2022



C. Conclusion:

Write the significance of the topic studied in the experiment.



Allan Rodrigues TE IT A-59

Rajdhani

DATE / /

Experiment - 6

Conclusion

In this experiment we learnt about the different classification algorithms like naive bayes, SVM(support vector machine) and KNN(k nearest neighbour).

A common job of machine learning algorithm is to recognize objects & being able to seprate them in to categories. Classification helps us to segregate vast quantities of data in to discrete values. It helps to secure sensitive information & identify relevant data.

D. References:

1. Machine Learning Classification Strategy In Python (quantinsti.com)

2.https://blog.quantinsti.com/machine-learning-k-nearest-neighbors-knn-algorithm-python/

3. Discrete vs. Continuous Data: All You Need to Know (yummysoftware.com)

4. How Naive Bayes Algorithm Works? (with example and full code) | ML+ (machinelearningplus.com)
