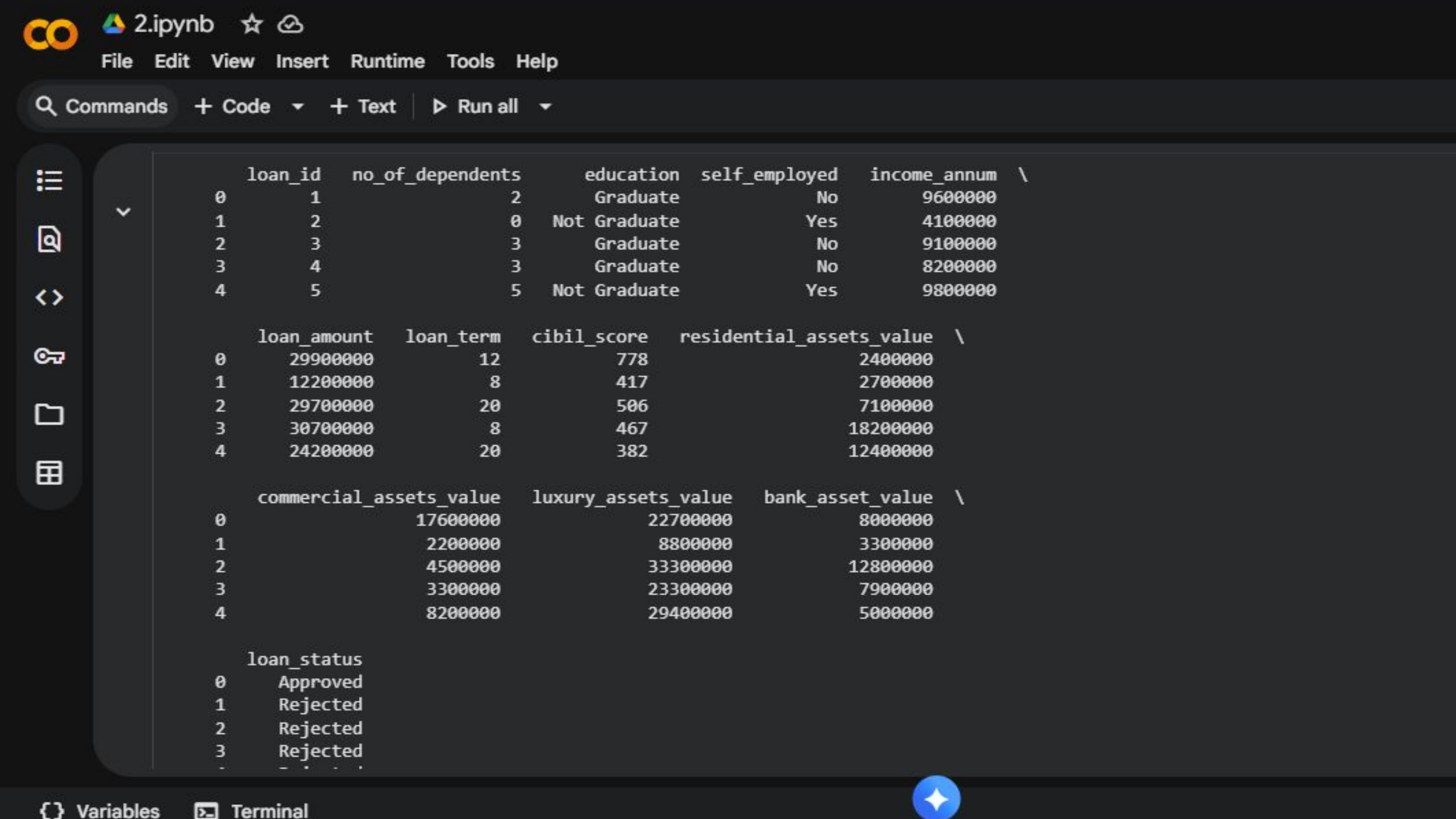
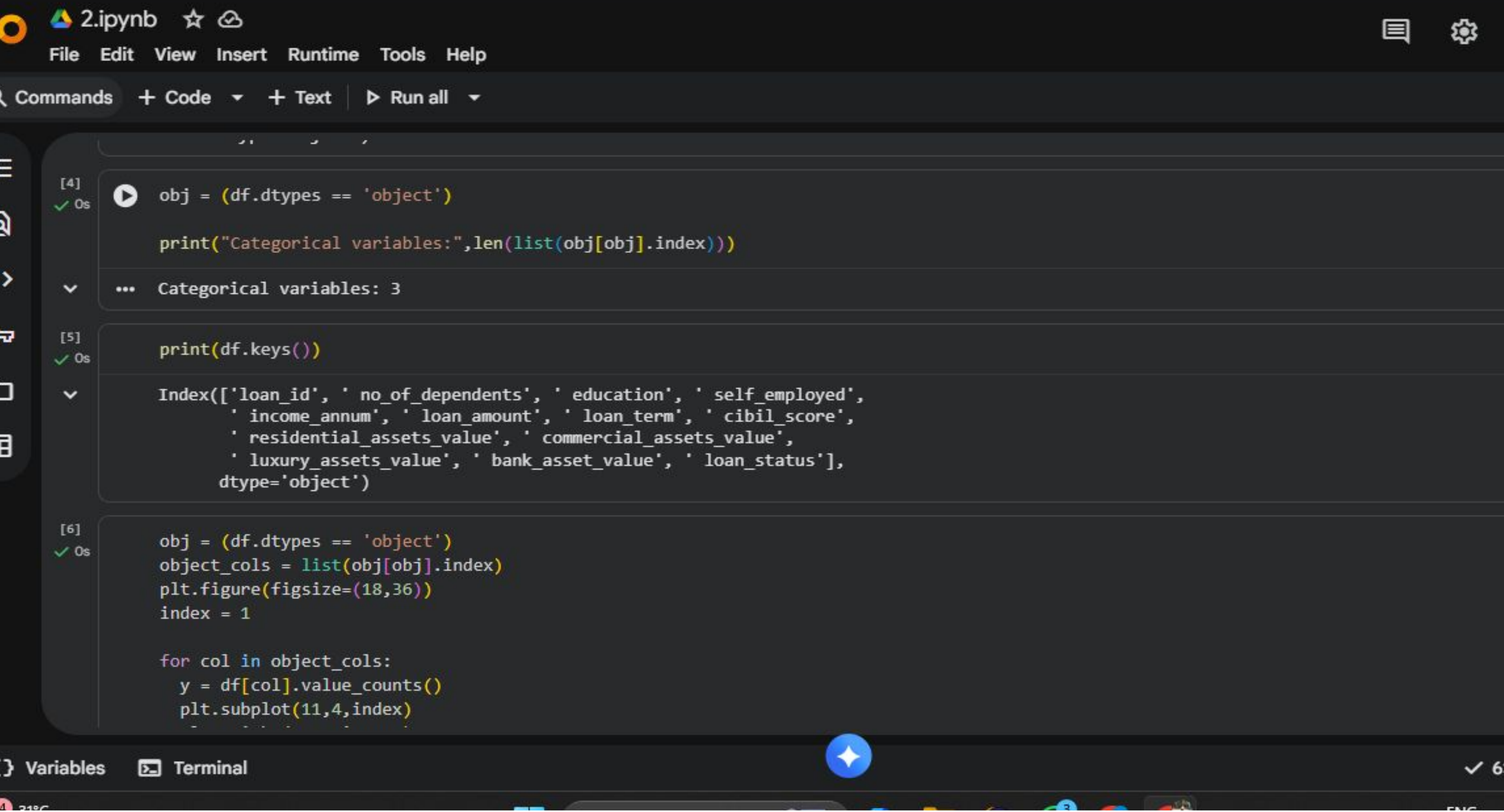


```
   loan_id  no_of_dependents     education self_employed  income_annum  \
0        1                 2      Graduate            No       9600000
1        2                 0  Not Graduate           Yes       4100000
2        3                 3      Graduate            No       9100000
3        4                 3      Graduate            No       8200000
4        5                 5  Not Graduate           Yes       9800000

    loan_amount  loan_term  cibil_score  residential_assets_value  \
0      29900000         12          778                   2400000
1      12200000          8          417                   2700000
2      29700000         20          506                   7100000
3      30700000          8          467                  18200000
4      24200000         20          382                  12400000

    commercial_assets_value  luxury_assets_value  bank_asset_value  \
0                  17600000             22700000           8000000
1                   2200000              8800000           3300000
2                   4500000             33300000          12800000
3                   3300000             23300000           7900000
4                   8200000             29400000           5000000

   loan_status
0     Approved
1     Rejected
2     Rejected
3     Rejected
```



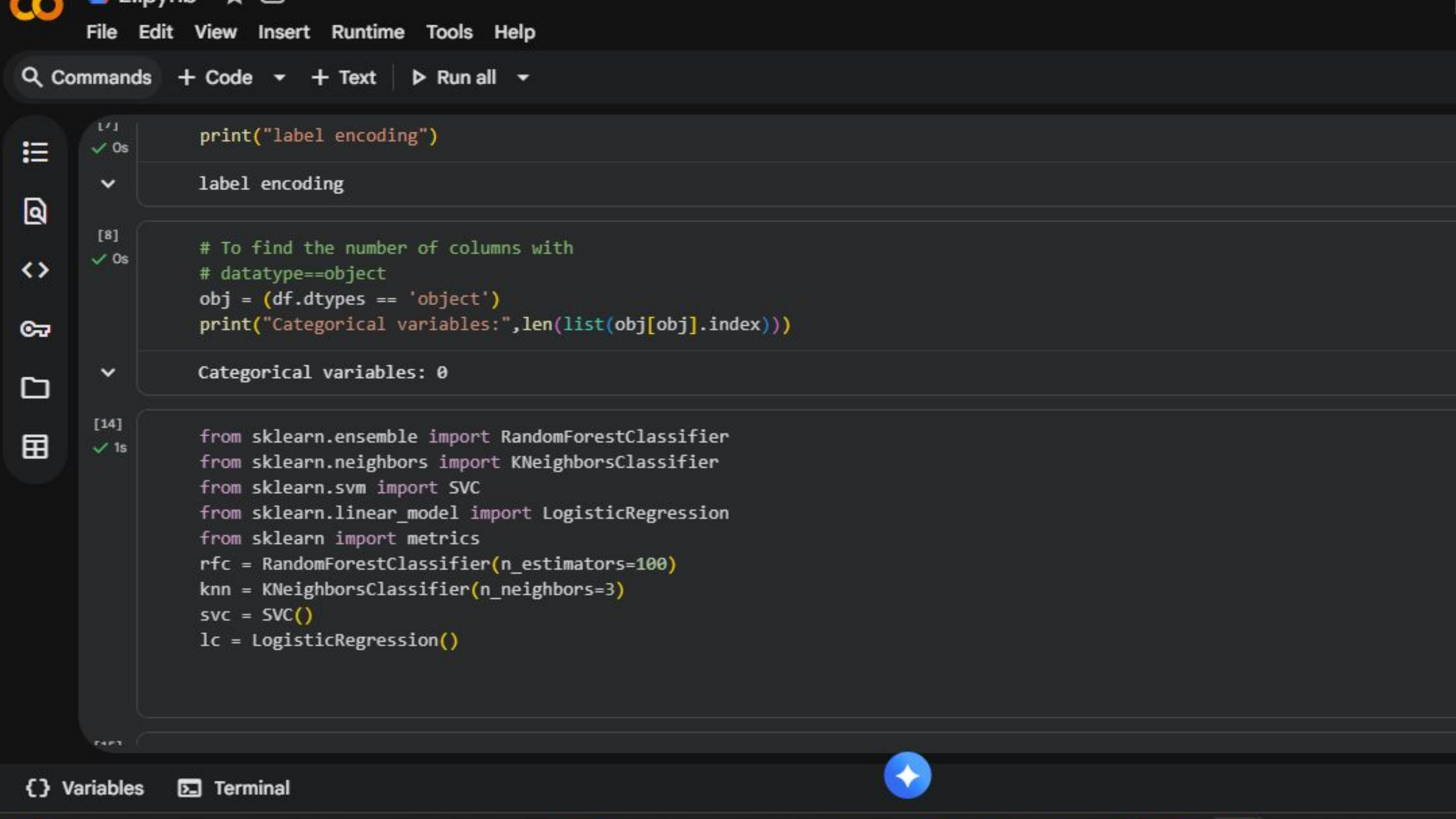
```
obj = (df.dtypes == 'object')

print("Categorical variables:",len(list(obj[obj].index)))
```

```
Categorical variables: 3
```

```
print(df.keys())
```

```
Index(['loan_id', ' no_of_dependents', ' education', ' self_employed',
       ' income_annum', ' loan_amount', ' loan_term', ' cibil_score',
       ' residential_assets_value', ' commercial_assets_value',
       ' luxury_assets_value', ' bank_asset_value', ' loan_status'],
      dtype='object')
```

```
obj = (df.dtypes == 'object')
object_cols = list(obj[obj].index)
plt.figure(figsize=(18,36))
index = 1

for col in object_cols:
  y = df[col].value_counts()
  plt.subplot(11,4,index)
```



[7] 0s

```
print("label encoding")
```

```
label encoding
```

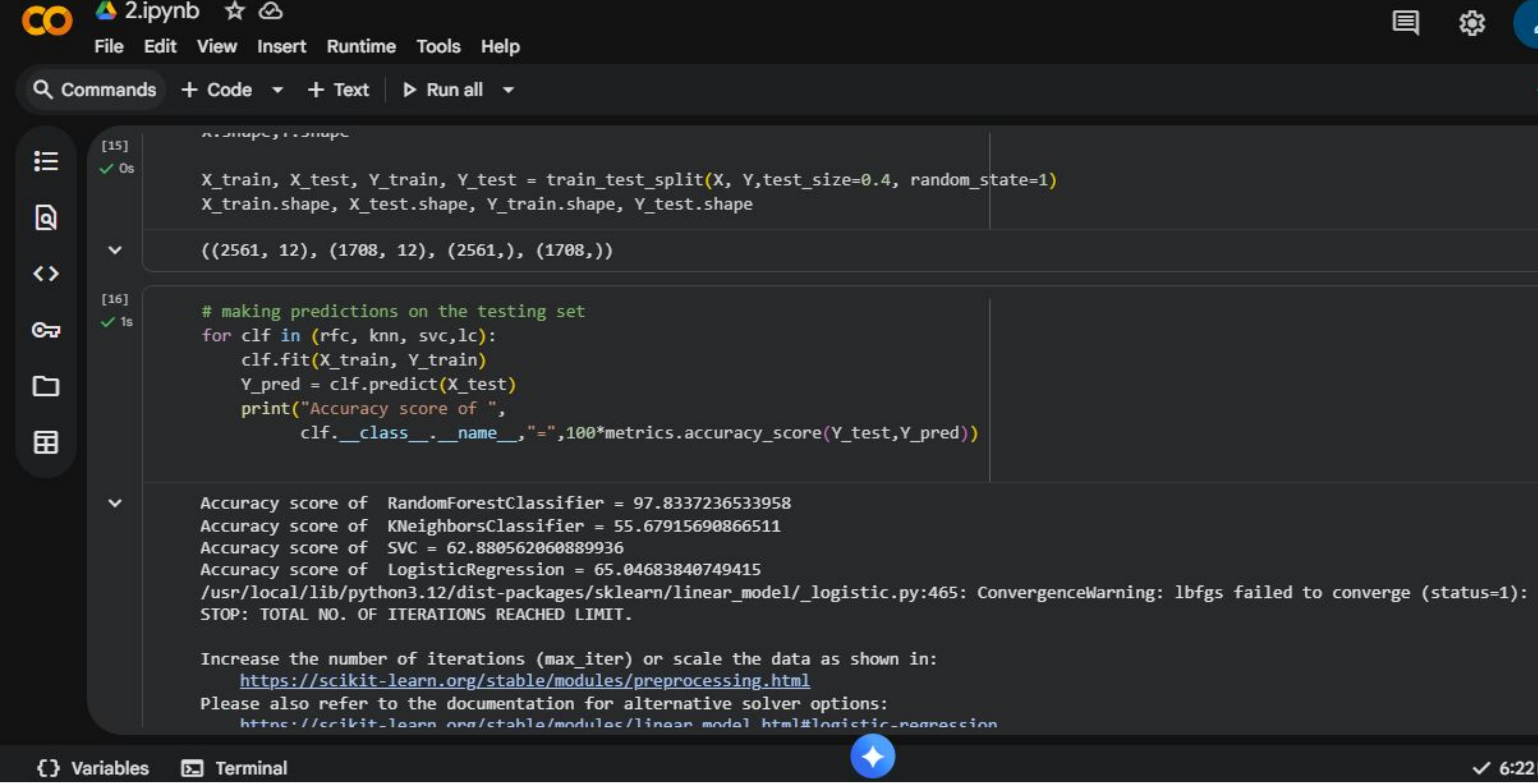
[8] 0s

```
# To find the number of columns with
# datatype==object
obj = (df.dtypes == 'object')
print("Categorical variables:",len(list(obj[obj].index)))
```

```
Categorical variables: 0
```

[14] 1s

```
from sklearn.ensemble import RandomForestClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.svm import SVC
from sklearn.linear_model import LogisticRegression
from sklearn import metrics
rfc = RandomForestClassifier(n_estimators=100)
knn = KNeighborsClassifier(n_neighbors=3)
svc = SVC()
lc = LogisticRegression()
```

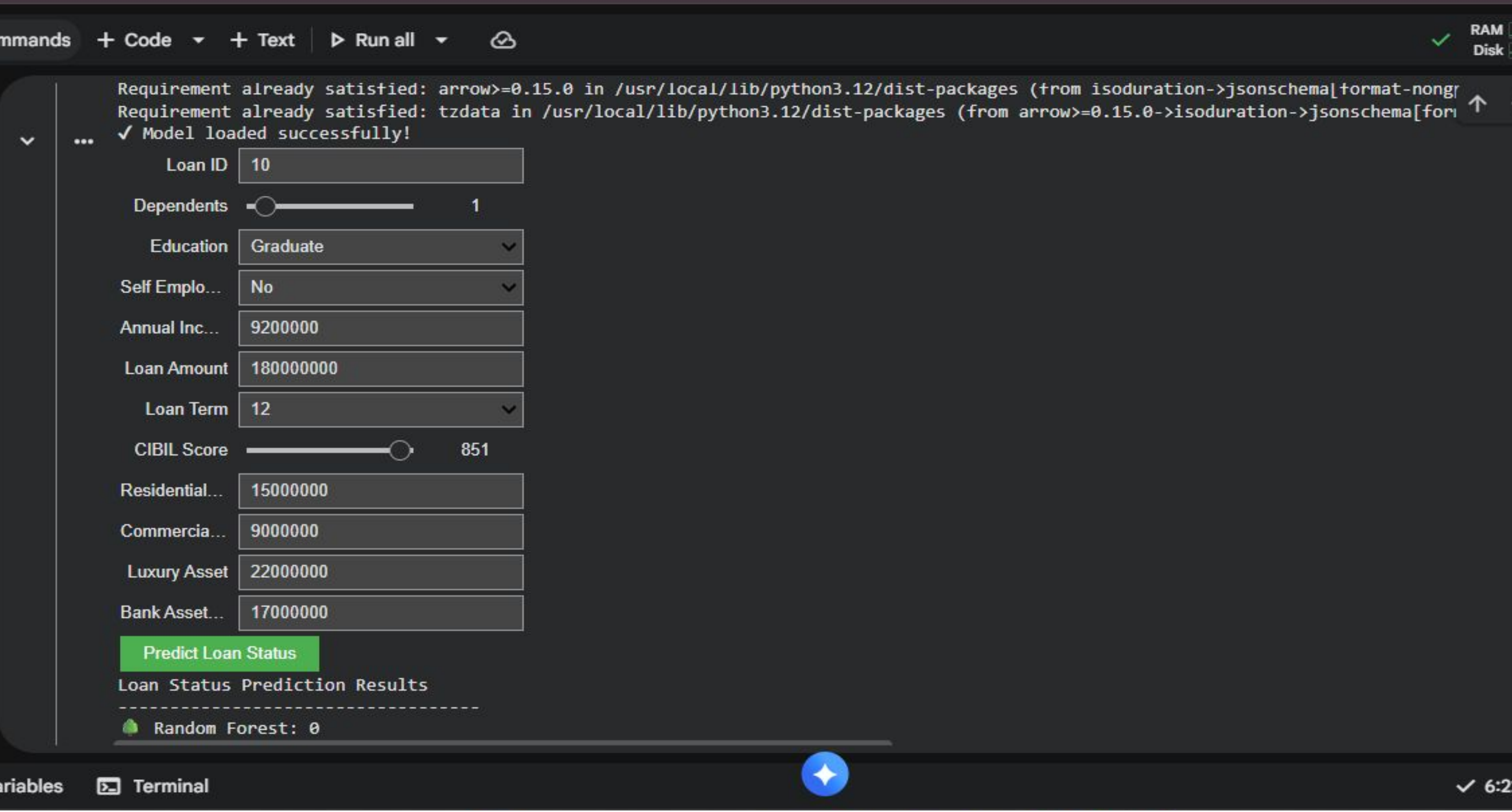


```
X.shape,Y.shape

X_train, X_test, Y_train, Y_test = train_test_split(X, Y,test_size=0.4, random_state=1)
X_train.shape, X_test.shape, Y_train.shape, Y_test.shape
```

```
((2561, 12), (1708, 12), (2561,), (1708,))
```

```
# making predictions on the testing set
for clf in (rfc, knn, svc,lc):
    clf.fit(X_train, Y_train)
    Y_pred = clf.predict(X_test)
    print("Accuracy score of ",
          clf.__class__.__name__,"=",100*metrics.accuracy_score(Y_test,Y_pred))
```

```
Accuracy score of  RandomForestClassifier = 97.8337236533958
Accuracy score of  KNeighborsClassifier = 55.67915690866511
Accuracy score of  SVC = 62.880562060889936
Accuracy score of  LogisticRegression = 65.04683840749415
/usr/local/lib/python3.12/dist-packages/sklearn/linear_model/_logistic.py:465: ConvergenceWarning: lbfgs failed to converge (status=1):
STOP: TOTAL NO. OF ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
```



```
Requirement already satisfied: arrow>=0.15.0 in /usr/local/lib/python3.12/dist-packages (from isoduration->jsonschema[format-nongr
Requirement already satisfied: tzdata in /usr/local/lib/python3.12/dist-packages (from arrow>=0.15.0->isoduration->jsonschema[for
✓ Model loaded successfully!
```

Loan ID: 10

Dependents: 1

Education: Graduate

Self Emplo...: No

Annual Inc...: 9200000

Loan Amount: 180000000

Loan Term: 12

CIBIL Score: 851

Residential...: 15000000

Commercia...: 9000000

Luxury Asset: 22000000

Bank Asset...: 17000000

Predict Loan Status

```
Loan Status Prediction Results
----------------------------------------
🌳 Random Forest: 0
```