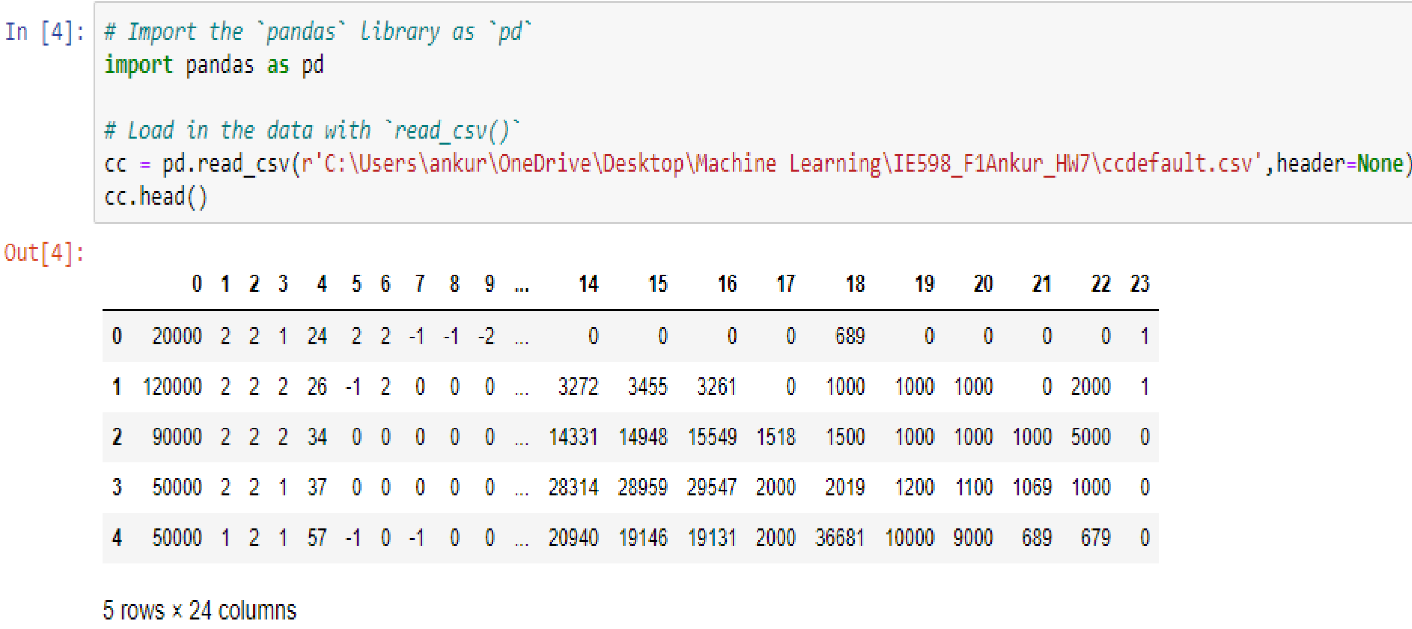
Ankur Mukherjee (ankurm3)

IE598 MLF F18

Module 7 Homework (Random Forest)

Part 1: Random forest estimators

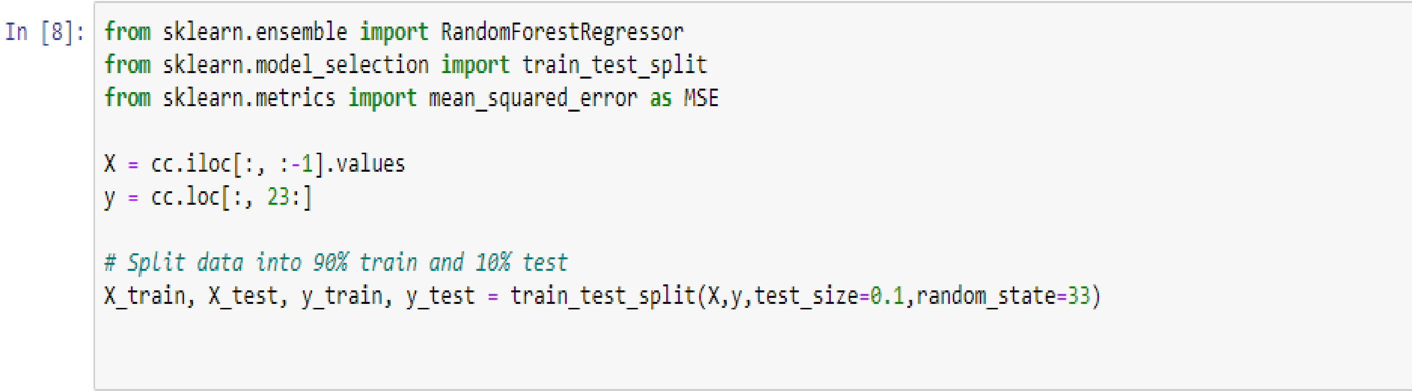


```
In [4]: # Import the `pandas` library as `pd`
        import pandas as pd

        # Load in the data with `read_csv()`
        cc = pd.read_csv(r'C:\Users\ankur\OneDrive\Desktop\Machine Learning\IE598_F1Ankur_HW7\ccdefault.csv',header=None)
        cc.head()
```

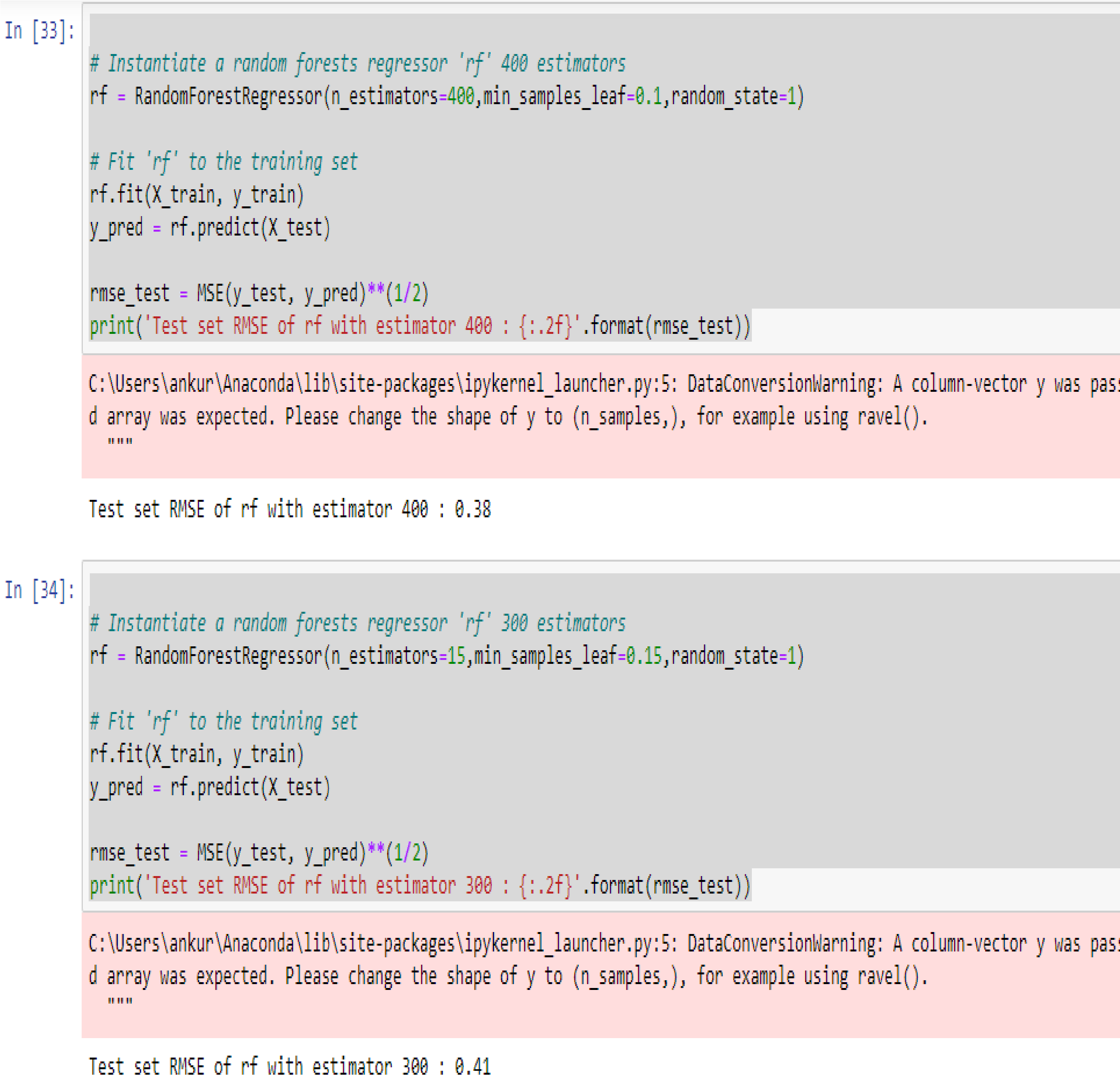
Out[4]:

	0	1	2	3	4	5	6	7	8	9	...	14	15	16	17	18	19	20	21	22	23
0	20000	2	2	1	24	2	2	-1	-1	-2	...	0	0	0	0	689	0	0	0	0	1
1	120000	2	2	2	26	-1	2	0	0	0	...	3272	3455	3261	0	1000	1000	1000	0	2000	1
2	90000	2	2	2	34	0	0	0	0	0	...	14331	14948	15549	1518	1500	1000	1000	1000	5000	0
3	50000	2	2	1	37	0	0	0	0	0	...	28314	28959	29547	2000	2019	1200	1100	1069	1000	0
4	50000	1	2	1	57	-1	0	-1	0	0	...	20940	19146	19131	2000	36681	10000	9000	689	679	0

5 rows × 24 columns



```
In [8]: from sklearn.ensemble import RandomForestRegressor
        from sklearn.model_selection import train_test_split
        from sklearn.metrics import mean_squared_error as MSE

        X = cc.iloc[:, :-1].values
        y = cc.loc[:, 23:]

        # Split data into 90% train and 10% test
        X_train, X_test, y_train, y_test = train_test_split(X,y,test_size=0.1,random_state=33)
```



In [33]:

```
# Instantiate a random forests regressor 'rf' 400 estimators
rf = RandomForestRegressor(n_estimators=400,min_samples_leaf=0.1,random_state=1)

# Fit 'rf' to the training set
rf.fit(X_train, y_train)
y_pred = rf.predict(X_test)

rmse_test = MSE(y_test, y_pred)**(1/2)
print('Test set RMSE of rf with estimator 400 : {:.2f}'.format(rmse_test))
```

```
C:\Users\ankur\Anaconda\lib\site-packages\ipykernel_launcher.py:5: DataConversionWarning: A column-vector y was pas
d array was expected. Please change the shape of y to (n_samples,), for example using ravel().
  """
```

```
Test set RMSE of rf with estimator 400 : 0.38
```

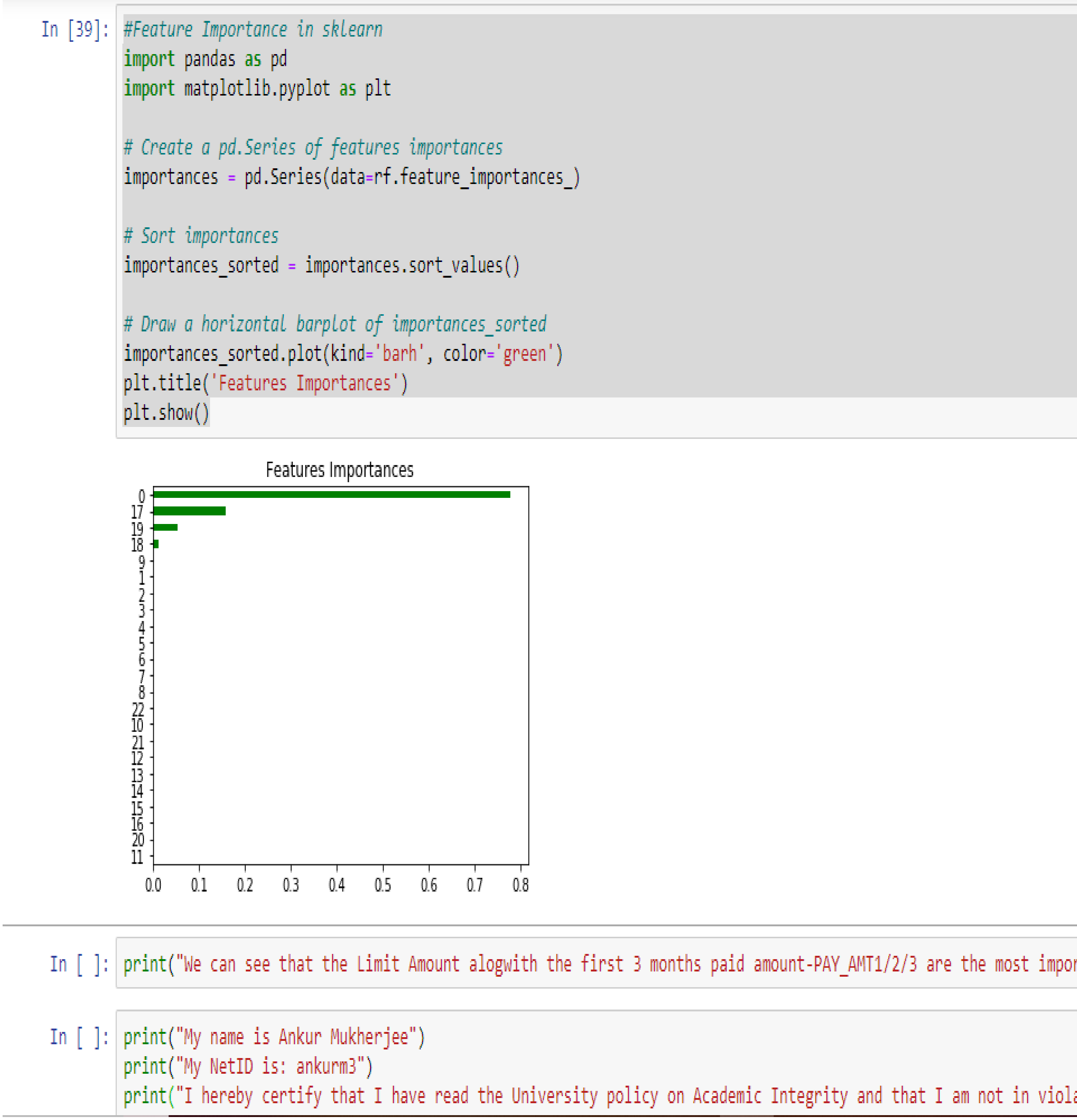
In [34]:

```
# Instantiate a random forests regressor 'rf' 300 estimators
rf = RandomForestRegressor(n_estimators=15,min_samples_leaf=0.15,random_state=1)

# Fit 'rf' to the training set
rf.fit(X_train, y_train)
y_pred = rf.predict(X_test)

rmse_test = MSE(y_test, y_pred)**(1/2)
print('Test set RMSE of rf with estimator 300 : {:.2f}'.format(rmse_test))
```

```
C:\Users\ankur\Anaconda\lib\site-packages\ipykernel_launcher.py:5: DataConversionWarning: A column-vector y was pas
d array was expected. Please change the shape of y to (n_samples,), for example using ravel().
  """
```

```
Test set RMSE of rf with estimator 300 : 0.41
```

Part 2: Random forest feature importance



In [39]:

```
#Feature Importance in sklearn
import pandas as pd
import matplotlib.pyplot as plt

# Create a pd.Series of features importances
importances = pd.Series(data=rf.feature_importances_)

# Sort importances
importances_sorted = importances.sort_values()

# Draw a horizontal barplot of importances_sorted
importances_sorted.plot(kind='barh', color='green')
plt.title('Features Importances')
plt.show()
```

In [ ]:

```
print("We can see that the Limit Amount alogwith the first 3 months paid amount-PAY_AMT1/2/3 are the most impor
```

In [ ]:

```
print("My name is Ankur Mukherjee")
print("My NetID is: ankurm3")
print("I hereby certify that I have read the University policy on Academic Integrity and that I am not in viola
```

Part 3: Conclusions

Short paragraph summarizing my findings:

a) What is the relationship between n_estimators, in-sample CV accuracy and computation time?
As number of estimators increase , the mean squared error of the decision tree decreases upto a certain point(n=400, RMSE = 0.38). But the time complexity of the model increases as we increase n_estimators , although not by any material amount

b) What is the optimal number of estimators for your forest?
In this model the optimum number of estimators is 400 for which the RMSE is minimized after which there is no marked improvement in the error accuracy

c) Which features contribute the most importance in your model according to scikit-learn function?
Feature Limit_Bal is the most important in the model with ~ 78% importance, followed by the paid amounts – PAY_Amt1 ~ 20%, PAY_Amt2 ~ 1.5%, PAY_Amt3 ~ 0.5%. This makes sense since the credit card balance should dictate whether a person defaults or not. Further, his paying history should also contribute to his default probabilities

d) What is feature importance and how is it calculated? (If you are not sure, refer to the Scikit-Learn.org documentation.)

Feature importance in tree based methods enable us to measure the importance of each feature in prediction. It calculates how much a tree node use a particular feature to reduce impurity – also known as the Mean Decrease Impurity.

Mean Decrease Impurity
Random forest consists of a number of decision trees. Every node in the decision trees is a condition on a single feature, designed to split the dataset into two so that similar response values end up in the same set. The measure based on which the (locally) optimal condition is chosen is called impurity. For classification, it is typically either Gini impurity or information gain/entropy and for regression trees it is variance. Thus when training a tree, it can be computed how much each feature decreases the weighted impurity in a tree. For a forest, the impurity decrease from each feature can be averaged and the features are ranked according to this measure.

Limitation of this method is that with correlated features, strong features can end up with low scores and the method can be biased towards variables with many categories

Part 4: Appendix

https://github.com/ankurmukherjeeuiuc?tab=repositories