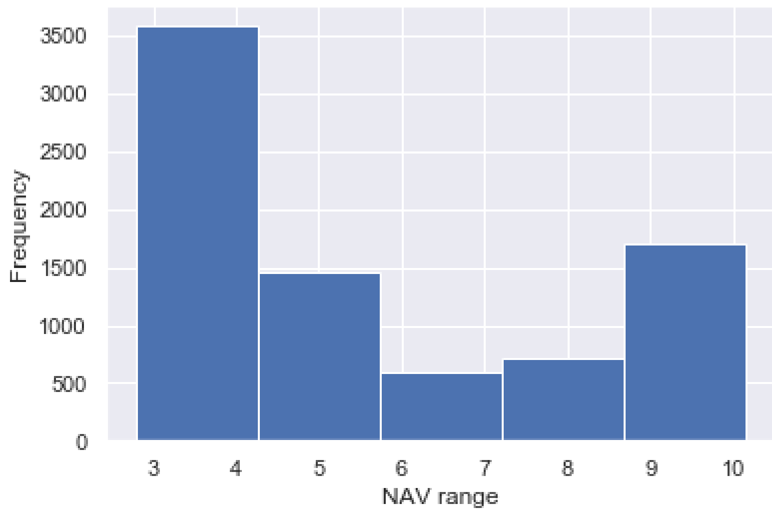
Ankur Mukherjee (ankurm3)

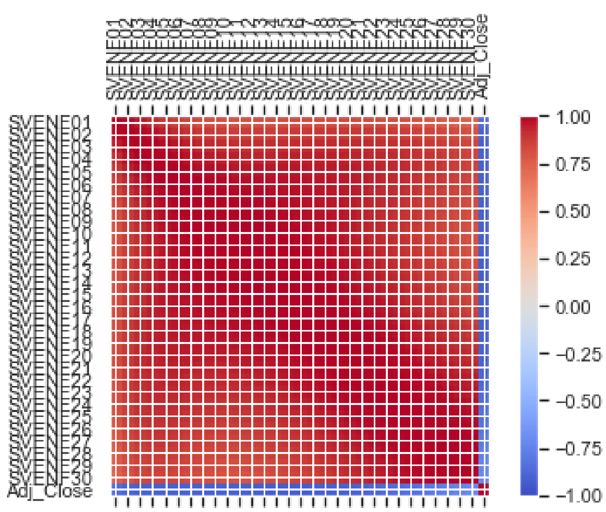
IE598 MLF F18

Module 5 Homework (Dimensionality Reduction)

Use the Treasury Yield Curve dataset

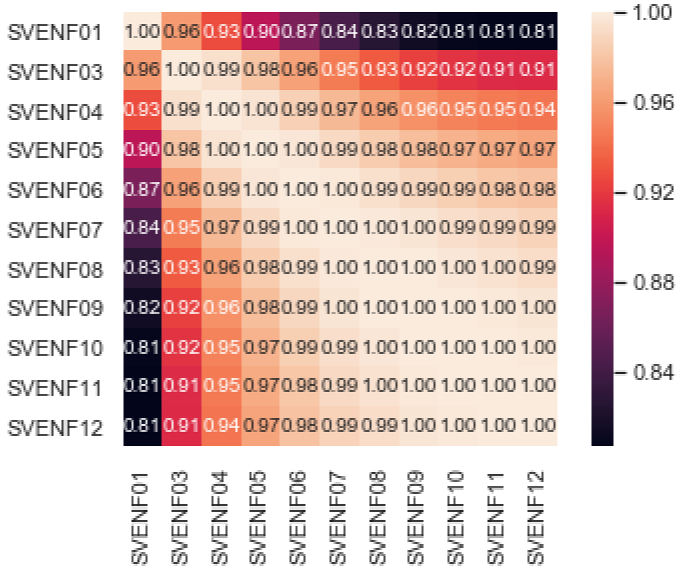
Part 1: Exploratory Data Analysis





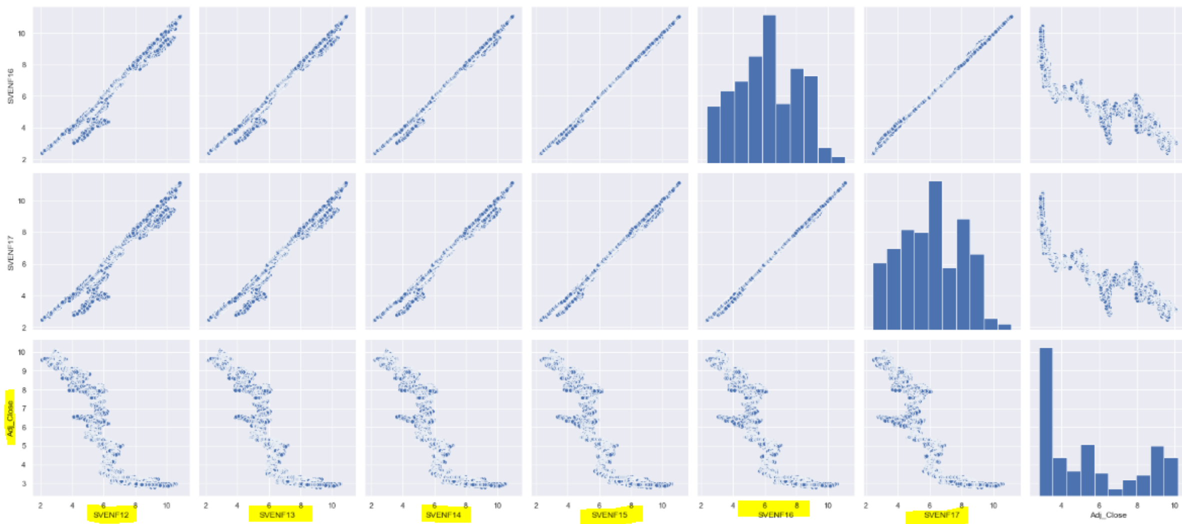
Correlations between the factors- dark red means ρX,Y is +1, dark blue ρX,Y is -1

Heat Map:




```
NAV Correlation with SVENF1 -0.8495622302124518
NAV Correlation with SVENF2 -0.8841940419624281
NAV Correlation with SVENF3 -0.8989522955811705
NAV Correlation with SVENF4 -0.9037070653128738
NAV Correlation with SVENF5 -0.9037790839277182
NAV Correlation with SVENF6 -0.9023432595330945
NAV Correlation with SVENF7 -0.9012419854036915
NAV Correlation with SVENF8 -0.9013166715609904
NAV Correlation with SVENF9 -0.9027058966887004
NAV Correlation with SVENF10 -0.9051340222439511
NAV Correlation with SVENF11 -0.9081358118523702
NAV Correlation with SVENF12 -0.9111990836903389
NAV Correlation with SVENF13 -0.9138433538543793
NAV Correlation with SVENF14 -0.9156507293510996
NAV Correlation with SVENF15 -0.9162734270068686
NAV Correlation with SVENF16 -0.9154282260087608
NAV Correlation with SVENF17 -0.9128898979548294
NAV Correlation with SVENF18 -0.9084830683034524
NAV Correlation with SVENF19 -0.9020801436295983
NAV Correlation with SVENF20 -0.893598697766412 5
NAV Correlation with SVENF21 -0.8830028668014344
NAV Correlation with SVENF22 -0.8703053141348676
NAV Correlation with SVENF23 -0.8555660899999161
NAV Correlation with SVENF24 -0.838893886173855
NAV Correlation with SVENF25 -0.8204399044152119
NAV Correlation with SVENF26 -0.8003948276812026
NAV Correlation with SVENF27 -0.7789795403317409
NAV Correlation with SVENF28 -0.7564348284313134
NAV Correlation with SVENF29 -0.733014305628527 9
NAV Correlation with SVENF30 -0.7089703629455018
```

Printing Scatter plot for factors – 12 to 17(The 6 most correlated ones as above)



Part 2: Perform a PCA on the Treasury Yield dataset

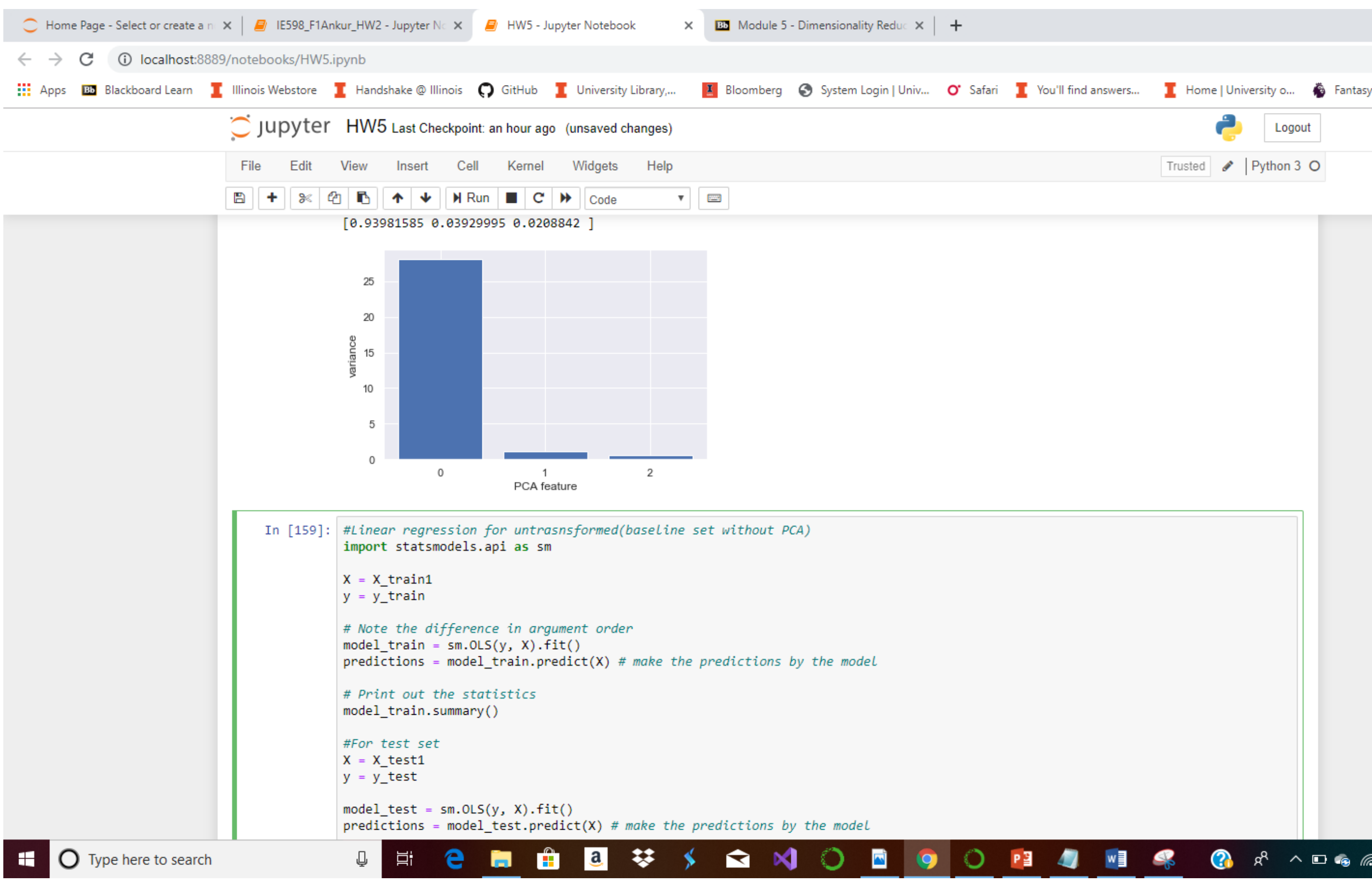
In [129]:
```
#Train Test split
from sklearn.model_selection import train_test_split
from sklearn import preprocessing
from sklearn.preprocessing import StandardScaler

X = data.iloc[:, :-1].values
y = data.iloc[:, :1].values

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.15,random_state=42)
print( X_train.shape, y_train.shape)
# performing preprocessing part
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
scaler.fit(X_train)
X_train = scaler.transform(X_train)
X_test = scaler.transform(X_test)
```

```
(7100, 30) (7100, 1)
```

```python
In [148]: #Explained for all components
from sklearn.decomposition import PCA
from sklearn.preprocessing import StandardScaler
from sklearn.pipeline import make_pipeline
import matplotlib.pyplot as plt

pca = PCA()
X_train1 = pca.fit_transform(X_train)
X_test1 = pca.transform(X_test)
explained_variance = pca.explained_variance_ratio_
print(explained_variance)
# Plot the explained variances
features = range(pca.n_components_)
plt.bar(features, pca.explained_variance_)
plt.xlabel('PCA feature')
plt.ylabel('variance')
plt.xticks(features)
plt.show()
```

```
[0.93981585 0.03929995 0.0208842 ]
```

Part 3: Logistic regression classifier v. SVM classifier - baseline



Home Page - Select or create a n | IE598_F1Ankur_HW2 - Jupyter N | HW5 - Jupyter Notebook | Module 5 - Dimensionality Redu

localhost:8889/notebooks/HW5.ipynb

Apps | Blackboard Learn | Illinois Webstore | Handshake @ Illinois | GitHub | University Library,... | Bloomberg | System Login | Univ... | Safari | You'll find answers... | Home

jupyter HW5 Last Checkpoint: an hour ago (unsaved changes)

File Edit View Insert Cell Kernel Widgets Help | Trusted

Run | Code

```
model_test.summary()
```

Out[159]:

OLS Regression Results

Dep. Variable:	y	**R-squared (uncentered):**	0.309
Model:	OLS	**Adj. R-squared (uncentered):**	0.307
Method:	Least Squares	**F-statistic:**	186.4
Date:	Sun, 29 Sep 2019	**Prob (F-statistic):**	6.76e-100
Time:	21:33:44	**Log-Likelihood:**	-3489.6
No. Observations:	1253	**AIC:**	6985.
Df Residuals:	1250	**BIC:**	7001.
Df Model:	3		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
x1	0.4136	0.021	19.991	0.000	0.373	0.454
x2	0.3852	0.103	3.752	0.000	0.184	0.587
x3	-1.8139	0.141	-12.833	0.000	-2.091	-1.537

Omnibus:	9.870	**Durbin-Watson:**	0.027
Prob(Omnibus):	0.007	**Jarque-Bera (JB):**	14.076
Skew:	-0.013	**Prob(JB):**	0.000878
Kurtosis:	3.519	**Cond. No.**	6.84

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

Type here to search

jupyter HW5 Last Checkpoint: an hour ago (autosaved)

In [155]:
```
#Explained for 3 components
pca3 = PCA(n_components=3)
X_train2 = pca3.fit_transform(X_train)
X_test2 = pca3.transform(X_test)
explained_variance = pca3.explained_variance_ratio_
cummulative_exp_var = explained_variance[0]+explained_variance[1]+explained_variance[2]
print("Individual PCA feature: " + str(explained_variance))
print("Cummulative explained Variation is: " + str(cummulative_exp_var))
# Plot the explained variances
features = range(pca3.n_components_)
plt.bar(features, pca3.explained_variance_)
plt.xlabel('PCA feature')
plt.ylabel('variance')
plt.xticks(features)
plt.show()
```

```
Individual PCA feature: [0.93981585 0.03929995 0.0208842 ]
Cummulative explained Variation is: 1.0
```

jupyter HW5 Last Checkpoint: an hour ago (autosaved)

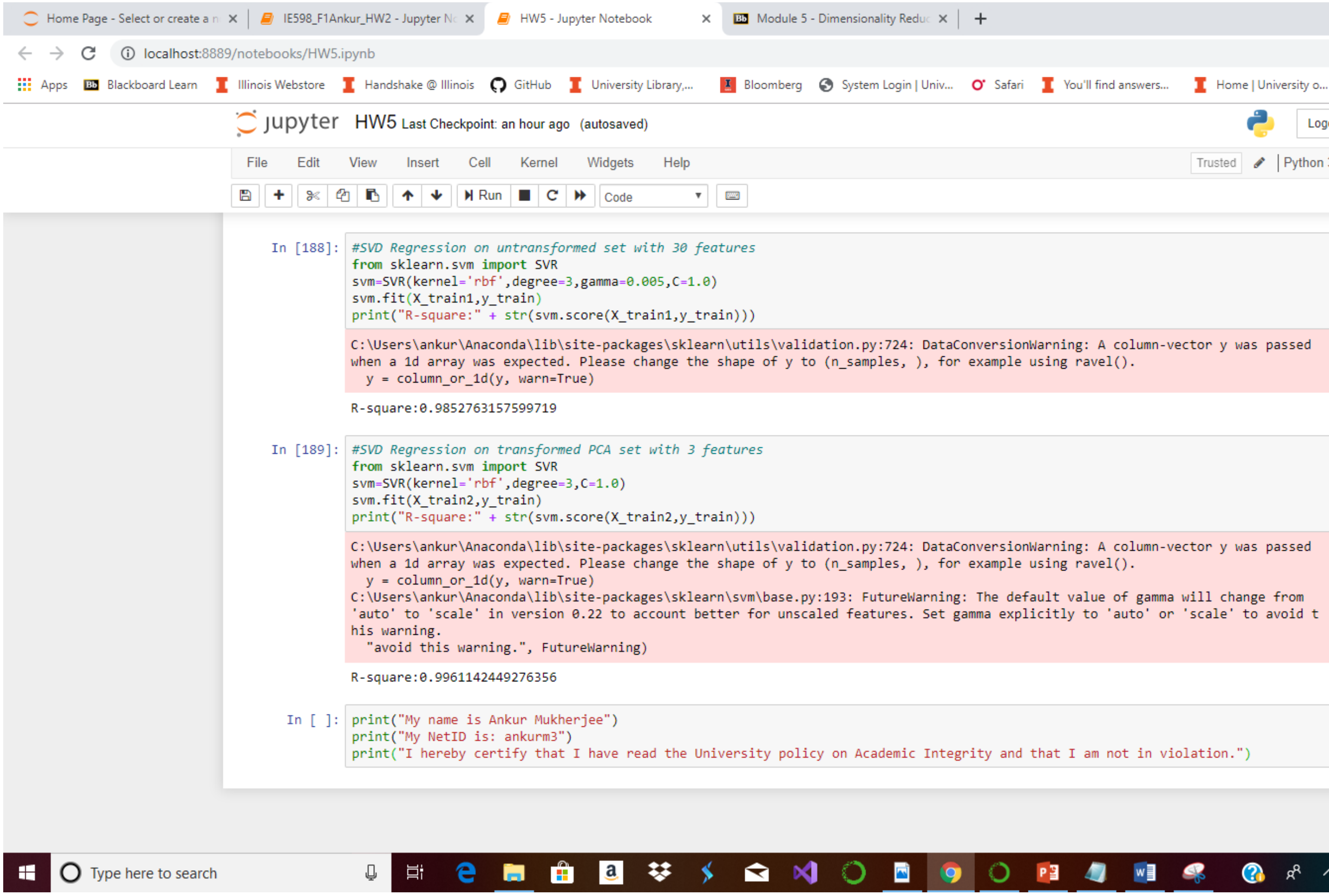
In [162]:

```
#Linear regression for trasnsformed PCA set with 3 features
import statsmodels.api as sm

X = X_train2
y = y_train

# Note the difference in argument order
model = sm.OLS(y, X).fit()
predictions = model.predict(X) # make the predictions by the model

# Print out the statistics
model.summary()

#For test set
X = X_test2
y = y_test

# Note the difference in argument order
model_test = sm.OLS(y, X).fit()
predictions = model_test.predict(X) # make the predictions by the model

# Print out the statistics
model_test.summary()
#print("Conclusion: There is no discernible performance or training time difference when doing Linear regressio
```

Out[162]:

OLS Regression Results

Dep. Variable:	y	**R-squared (uncentered):**	0.309
Model:	OLS	**Adj. R-squared (uncentered):**	0.307
Method:	Least Squares	**F-statistic:**	186.4
Date:	Sun, 29 Sep 2019	**Prob (F-statistic):**	6.76e-100
Time:	21:35:12	**Log-Likelihood:**	-3489.6
No. Observations:	1253	**AIC:**	6985.



jupyter HW5 Last Checkpoint: an hour ago (autosaved)

In [188]:
```
#SVD Regression on untransformed set with 30 features
from sklearn.svm import SVR
svm=SVR(kernel='rbf',degree=3,gamma=0.005,C=1.0)
svm.fit(X_train1,y_train)
print("R-square:" + str(svm.score(X_train1,y_train)))
```

```
C:\Users\ankur\Anaconda\lib\site-packages\sklearn\utils\validation.py:724: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please change the shape of y to (n_samples, ), for example using ravel().
  y = column_or_1d(y, warn=True)
```

```
R-square:0.9852763157599719
```

In [189]:
```
#SVD Regression on transformed PCA set with 3 features
from sklearn.svm import SVR
svm=SVR(kernel='rbf',degree=3,C=1.0)
svm.fit(X_train2,y_train)
print("R-square:" + str(svm.score(X_train2,y_train)))
```

```
C:\Users\ankur\Anaconda\lib\site-packages\sklearn\utils\validation.py:724: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please change the shape of y to (n_samples, ), for example using ravel().
  y = column_or_1d(y, warn=True)
C:\Users\ankur\Anaconda\lib\site-packages\sklearn\svm\base.py:193: FutureWarning: The default value of gamma will change from 'auto' to 'scale' in version 0.22 to account better for unscaled features. Set gamma explicitly to 'auto' or 'scale' to avoid t
his warning.
  "avoid this warning.", FutureWarning)
```

```
R-square:0.9961142449276356
```

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 violation.")
```

Part 4: Conclusions

	Experiment 1 (Treasury Yields)			
	Logistic		SVM	
Baseline (all attributes)	Train Acc:	0.314	Train Acc:	0.314
	Test Acc:	0.309	Test Acc:	0.309
PCA transform (3 PCs)	Train Acc:	0.985	Train Acc:	0.996
	Test Acc:	0.77	Test Acc:	0.81

Conclusion:

There is no change noticed in the performance or training time in the Regression models on the untransformed and PCA three feature transformed set. However, on the SVM – performance and training time both improved on the PCA set

Part 5: Appendix

https://github.com/ankurmukherjeeuiuc/