3-Copy1

September 23, 2018

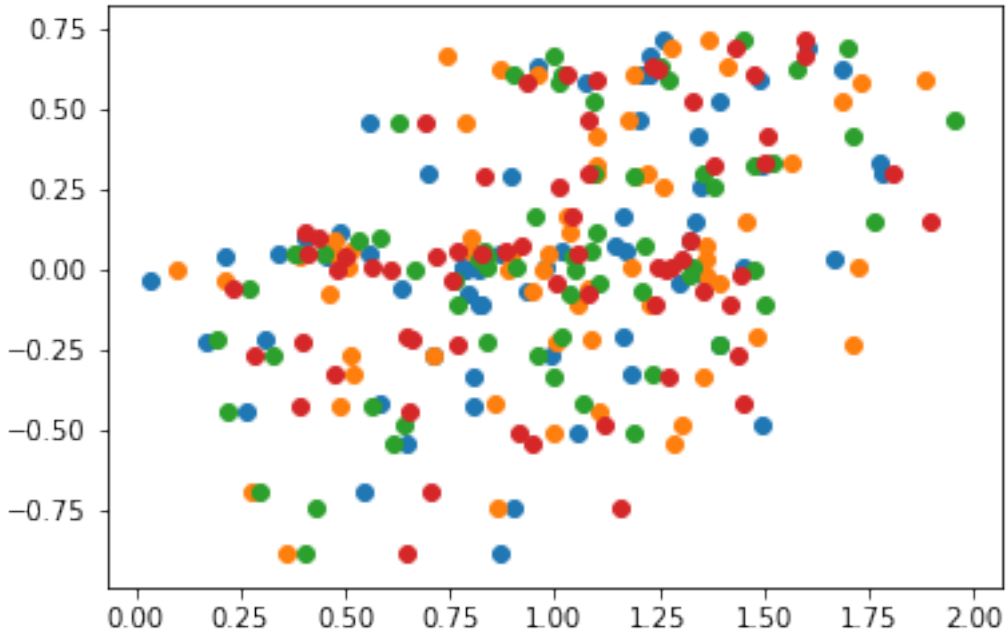
1 Assigment 3

This assigment focuses on getting comfortable with working with multidimensional data and linear regression. Key items include: - Creating random n-dimensional data - Creating a Model that can handle the data - Plot a subset of the data along with the prediction - Using a Dataset to read in and choose certain columns to produce a model - Create several models from various combinations of columns - Plot a few of the results - BONUS: Perform all the plots in 3D instead of 2D

1.1 1. Create a 4 dimensional data set with 64 elements and show 2D plots of the data $x_1 \rightarrow y, x_2 \rightarrow y$, etc.

In [1]:
```
import numpy as np
import matplotlib.pylab as plt
%matplotlib inline

n = 64
x = np.linspace(0, 1, n) + np.random.rand(4, n)
x = np.vstack([x, np.ones(len(x.T))]).T
y = np.linspace(0, 1, n) + np.random.rand(n) - 1

plt.scatter(x.T[0], y) # First dimension
plt.scatter(x.T[1], y) # Second dimension
plt.scatter(x.T[2], y) # Third dimension
plt.scatter(x.T[3], y) # Fourth dimension
```

Out[1]:
```
<matplotlib.collections.PathCollection at 0x1157aaa90>
```



1.2 2. Create a model to fit the data. Hint: follow the example from Lesson 3

In [2]:
```
# $\beta = (X^T X)^{-1} Y^T X$

# The manual method you showed us to calculate Bet

left = np.linalg.inv(np.dot(x.T, x))
right = np.dot(y.T, x)
np.dot(left, right)
```

Out[2]:
```
array([ 0.09470323,  0.10105578,  0.33914364,  0.06129596, -0.55099977])
```

In [3]:
```
# numpy method - it creates an error message for me,
# but the output is correct. I am unsure what this error message is about

beta = np.linalg.lstsq(x, y)[0]
beta
```

```
/anaconda2/envs/mypython3/lib/python3.7/site-packages/ipykernel_launcher.py:4: FutureWarning: `r
To use the future default and silence this warning we advise to pass `rcond=None`, to keep using
  after removing the cwd from sys.path.
```

Out[3]:
```
array([ 0.09470323,  0.10105578,  0.33914364,  0.06129596, -0.55099977])
```

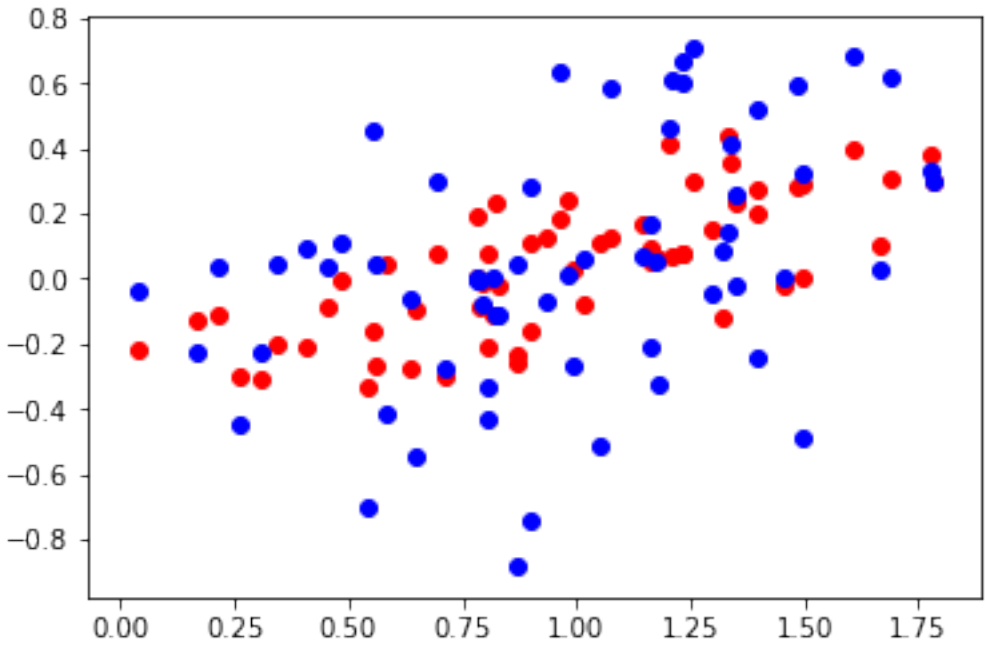
In [4]:
```
pred = np.dot(x, beta)
pred
```

```
Out[4]: array([-0.32819813, -0.30268722, -0.22134168, -0.08282063, -0.12522145,
               -0.10680447, -0.25534966, -0.27586097, -0.29966594, -0.2087093 ,
               -0.21150446, -0.20132915, -0.30585192, -0.16117159, -0.01258595,
               -0.23255654, -0.26953766,  0.05058696,  0.10957476,  0.06032048,
               -0.08867335, -0.09350899, -0.00250673, -0.11692809, -0.1127666 ,
                0.00345573,  0.10834844, -0.07393745,  0.07423638, -0.16354085,
               -0.02021859,  0.06406089,  0.00770329,  0.16400833,  0.2525359 ,
                0.04269851,  0.07976713,  0.02862872,  0.09342059,  0.19164902,
                0.08150634, -0.01798392,  0.28822476,  0.10135314,  0.07728914,
                0.27475077,  0.14895155,  0.23685484,  0.20480525,  0.12588669,
                0.29736231,  0.24487035,  0.12500781,  0.30855994,  0.38235151,
                0.29951657,  0.2340264 ,  0.2802406 ,  0.06976129,  0.35959023,
                0.18321368,  0.43817923,  0.39510909,  0.40994147])
```

1.3 3. Plot the model's prediction in 2D for 2 of the dimensions ($x_1 \rightarrow y_p, x_2 \rightarrow y_p$) along with the original points

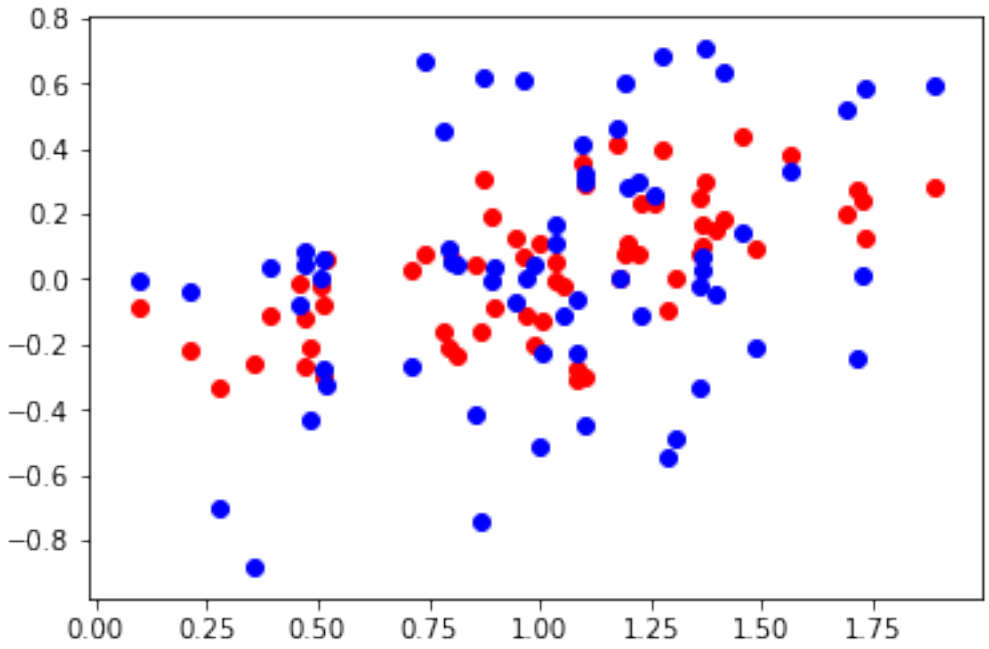
```
In [5]: plt.scatter(x.T[0], pred, c='red')
        plt.scatter(x.T[0], y, c='b')
```

```
Out[5]: <matplotlib.collections.PathCollection at 0x115cc25f8>
```



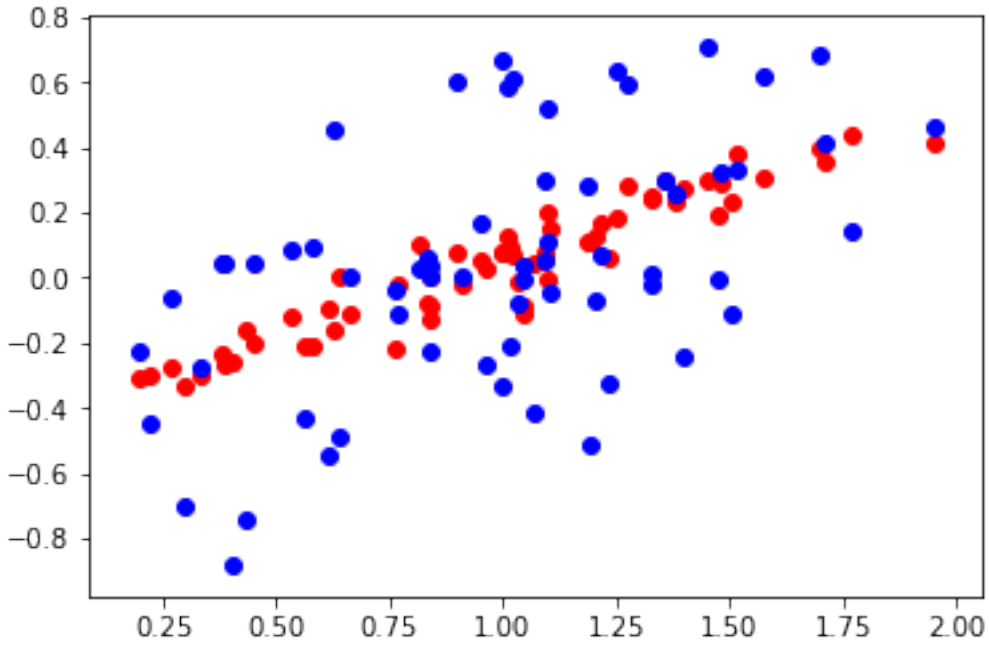

```
In [6]: plt.scatter(x.T[1], pred, c='red')
        plt.scatter(x.T[1], y, c='b')
```

```
Out[6]: <matplotlib.collections.PathCollection at 0x115d87860>
```

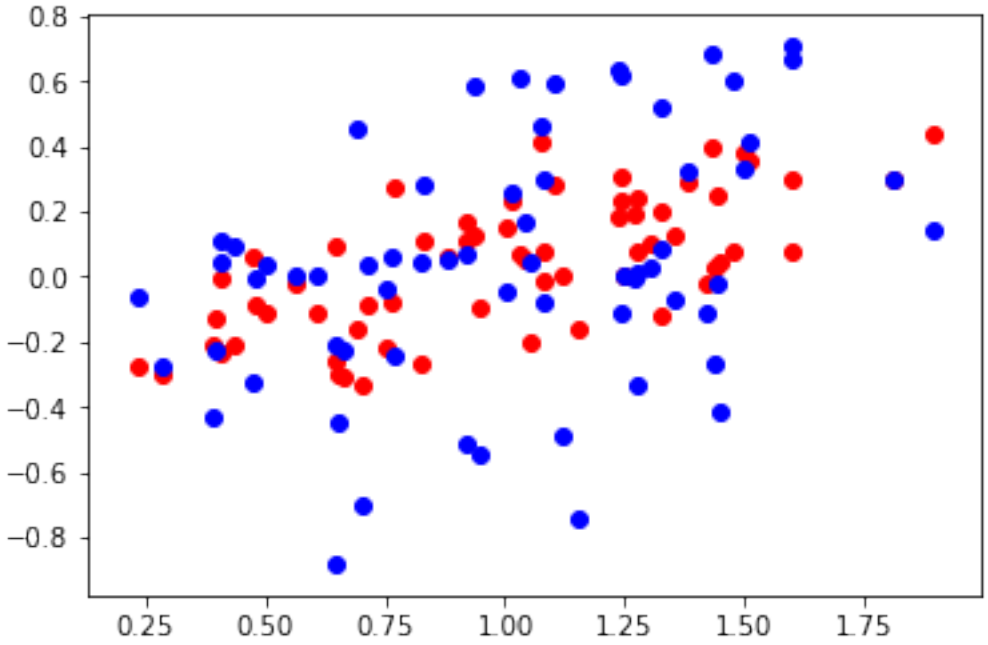


In [7]:
```
plt.scatter(x.T[2], pred, c='red')
plt.scatter(x.T[2], y, c='b')
```

Out[7]: <matplotlib.collections.PathCollection at 0x115ded470>



```
In [8]: plt.scatter(x.T[3], pred, c='red')
        plt.scatter(x.T[3], y, c='b')

Out[8]: <matplotlib.collections.PathCollection at 0x115f17048>
```

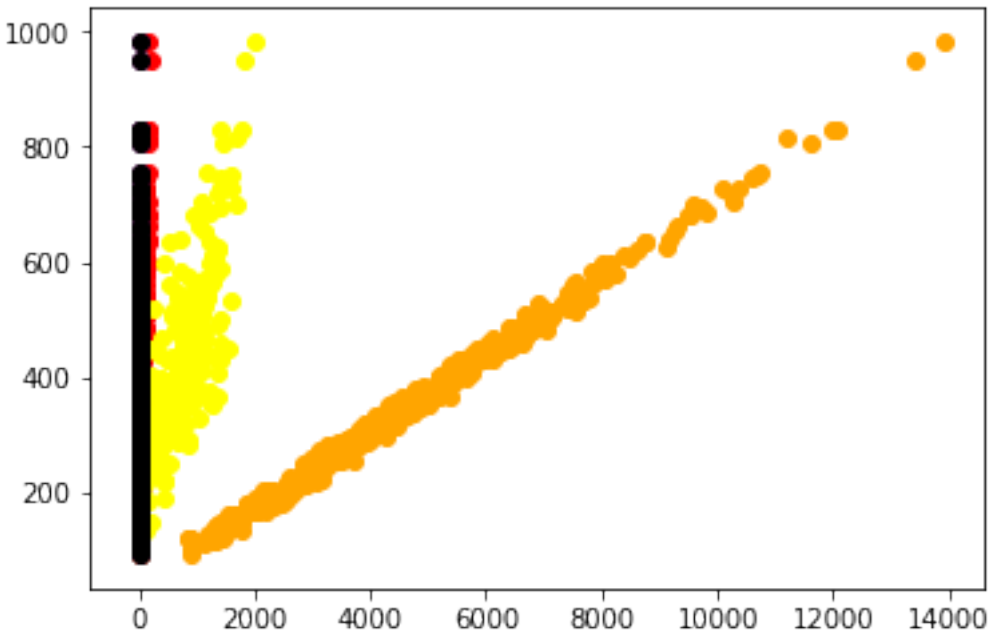


1.4 4. Read in `mlnn/data/Credit.csv` with Pandas and create a model to predict Credit Rating (`Rating`). Use only the numeric columns in your model, but feel free to experiment which which columns you believe are better predicters of Credit Rating

```
In [9]: import pandas as pd
        credit = pd.read_csv('../data/Credit.csv')
        credit.head()

Out[9]:    Unnamed: 0   Income  Limit  Rating  Cards  Age  Education  Gender Student  \
        0           1   14.891   3606     283      2   34         11    Male      No
        1           2  106.025   6645     483      3   82         15  Female     Yes
        2           3  104.593   7075     514      4   71         11    Male      No
        3           4  148.924   9504     681      3   36         11  Female      No
        4           5   55.882   4897     357      2   68         16    Male      No

          Married  Ethnicity  Balance
        0     Yes  Caucasian      333
        1     Yes      Asian      903
        2      No      Asian      580
        3      No      Asian      964
        4     Yes  Caucasian      331
```

In [10]:
```
X = credit[['Income', 'Limit', 'Balance', 'Cards']].values
X = np.vstack([X.T, np.ones(len(X))]).T

y = credit['Rating']

plt.scatter(X.T[0], y, c='red') # Should be income?
plt.scatter(X.T[1], y, c='orange') # Limit?
plt.scatter(X.T[2], y, c='yellow') # Balance?
plt.scatter(X.T[3], y, c='purple') # Cards ?
plt.scatter(X.T[4], y, c='black') # Cards ?
```

Out[10]: <matplotlib.collections.PathCollection at 0x1179f4860>



In [11]:
```
left = np.linalg.inv(np.dot(X.T, X))
right = np.dot(y.T, X)
np.dot(left, right)

beta = np.linalg.lstsq(X, y)[0]

pred = np.dot(X, beta)
pred
```

```
/anaconda2/envs/mypython3/lib/python3.7/site-packages/ipykernel_launcher.py:5: FutureWarning: `r
To use the future default and silence this warning we advise to pass `rcond=None`, to keep using
  """
```

```
Out[11]: array([273.44963788, 486.69596644, 516.33685502, 675.02344951,
                360.30520124, 581.0542427 , 258.97911467, 508.59822189,
                267.51221177, 498.12935688, 585.97509754, 128.00275184,
                383.34330186, 490.58048784, 252.12974209, 206.51509225,
                286.1205154 , 330.11444577, 455.0417298 , 475.85225245,
                233.24505158, 454.11092975, 213.32992323, 393.98267069,
                156.26272708, 336.29752855, 298.70115261, 335.86685974,
                931.9553508 , 417.52767881, 421.34340334, 230.15148871,
                558.33020749, 167.56057114, 209.58513168, 212.28772168,
                464.63673323, 465.74574703, 302.99570029, 273.52587071,
                259.87477123, 549.89494971, 351.01055476, 446.35731906,
                464.03689298, 544.12181296, 373.5003208 , 333.00272637,
                205.42496867, 352.02681467, 384.12704938, 298.16327385,
                391.6530631 , 407.54550211, 135.40097143, 155.51668362,
                374.40758505, 362.50205856, 264.92021218, 396.46546559,
                380.11352556, 243.73851283, 147.91277009, 242.12798215,
                229.42971099, 320.35811607, 692.6456174 , 382.92758597,
                407.97132817, 498.67607788, 306.38513841, 527.7552046 ,
                356.78638644, 344.82038736, 402.54556607, 245.36157276,
                251.69656527, 245.18406966, 484.17363429, 174.09851821,
                267.8909934 , 312.76094283, 333.5101541 , 136.24120107,
                237.94299994, 851.23878477, 452.15714787, 194.10997343,
                329.96125577, 540.57824102, 427.78842422, 441.62501145,
                231.97159848, 395.98309555, 243.56797136, 108.02752715,
                391.19092648, 268.50410403, 234.48395294, 616.12452126,
                277.40720174, 203.838898  , 544.12799253, 691.90371927,
                360.94478146, 244.94324062, 125.12883612, 251.00013611,
                446.86499819, 265.00057973, 256.21208043, 230.50340049,
                486.34280011, 470.1962966 , 264.53493019, 369.97349337,
                181.72070225, 638.010856  , 183.77702067, 135.12201448,
                130.29550398, 579.57907849, 499.57660865, 126.98864532,
                198.24410516, 198.18428025, 421.09125283, 266.98603791,
                597.72646095, 268.52920074, 301.80811411, 153.53945433,
                406.3701527 , 431.85369499, 427.3191907 , 271.56321437,
                307.94623189, 284.19783988, 172.86975067, 732.74054031,
                446.01384688, 485.26980425, 534.18584468, 364.53930767,
                221.13573367, 344.93926881, 378.8078108 , 141.05149482,
                194.58718026, 109.95944834, 421.19151129, 360.49666814,
                188.46336807, 347.49569255, 245.61380235, 136.42045898,
                333.54129956, 405.94611357, 403.56444983, 240.1993231 ,
                372.27682307, 154.87530554, 531.61073671, 202.79936357,
                442.86599673, 344.96245897, 226.10039893, 185.24461082,
                216.75047338, 455.55447298, 175.51156636, 313.1090868 ,
                357.03893889, 354.49448449, 752.45284628, 184.52035479,
                216.87505975, 297.64262877, 326.13872372, 549.1277026 ,
                286.48996522, 383.75735284, 462.14332454, 312.87519314,
                804.42273748, 324.87913904, 287.19578271, 179.87210953,
                545.7887566 , 327.66496151, 400.5638873 , 682.05334073,
```

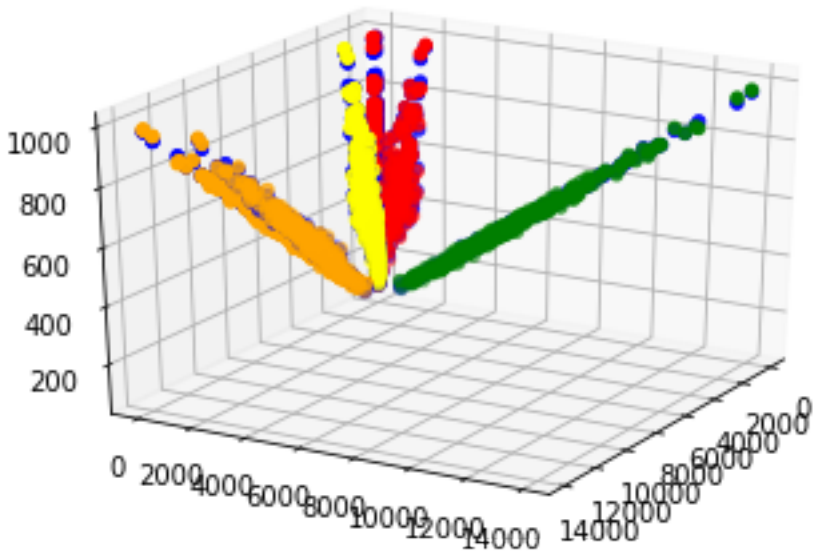
```
305.2110679 , 733.32559561, 176.74756046, 392.3303877 ,
523.21054593, 298.58768111, 178.17103915, 315.18166075,
389.91660012, 543.17043221, 132.66392304, 505.52221638,
387.98721877, 321.25798732, 197.29959969, 344.70751971,
327.14400239, 646.37738068, 246.75545425, 392.73816283,
323.43819081, 383.55353751, 386.30466728, 319.99344765,
221.72626571, 397.88736198, 146.02318114, 391.58521607,
427.0639254 , 633.96741682, 461.52293668, 355.42969514,
566.68433637, 421.67613783, 514.26467567, 413.3641812 ,
345.44954468, 543.0030435 , 380.92747837, 361.63714502,
347.63343435, 189.83340711, 589.4252016 , 232.71208755,
366.01965246, 391.98250722, 246.42844694, 265.46948048,
271.12847714,  97.30943751, 123.67374698, 470.55468253,
164.02559017, 183.6279023 , 256.86323834, 188.45396288,
 92.54264281, 144.92540256, 194.77266872, 261.30661687,
613.02048662, 399.37918219, 469.32386661, 320.07437384,
154.08325737, 194.60828407, 206.13713744, 466.61836508,
389.4350756 , 666.22308295, 297.85624189, 268.7619449 ,
368.99784721, 372.79123646, 373.87208258, 427.56737172,
136.1967488 , 414.59145856, 233.76833031, 353.40237682,
283.47204221, 370.54019593, 439.59415919, 623.29721551,
264.88360986, 367.15998883, 500.52395964, 246.48937168,
393.58379639, 164.77847958, 574.62772722, 465.26580584,
170.56649676, 143.90902161, 139.06411538, 251.88833864,
389.77945972, 293.62671387, 263.44359088, 287.34985377,
373.88043485, 807.79764763, 213.84362019, 143.60310021,
371.85818223, 330.63183861, 206.6037455 , 375.01154303,
337.3099813 , 273.29638312, 359.9881313 , 359.82485994,
535.13021864, 163.58500128, 288.96004645, 298.47364104,
342.69539373, 507.09548978, 358.72485037, 400.22316097,
401.63381102, 547.05969542, 653.63277476, 298.50869539,
521.26933439, 348.05157976, 139.55768404, 220.57442537,
118.45365689, 239.62391524, 285.08930135, 974.60210546,
227.8756718 , 366.42345258, 722.63238382, 483.0414414 ,
283.27348333, 540.24868927, 339.95246679, 298.13818185,
387.29703202, 259.71292914, 350.33741397, 260.57189988,
435.73661104,  98.44227034, 382.50528955, 716.52584   ,
296.91664914, 294.34622366, 244.68768274, 282.66857262,
404.96629452, 137.56468455, 406.82987197, 753.16341604,
116.4725815 , 381.63868327, 150.20450186, 371.52437109,
511.1015924 , 348.09131292, 286.26886658, 835.09327626,
444.70571794, 199.83970461, 317.21475388, 335.73881631,
436.44404613, 363.41601428, 378.3353876 , 437.41074129,
695.89665514, 478.19439366, 564.11639872, 273.93929799,
423.67961683, 574.76811029, 452.65814619, 188.92894476,
289.27060744, 394.99597573, 360.89275898, 426.3504993 ,
512.70570829, 145.70050701, 358.03359486, 221.77246411,
557.29899793, 570.11112413, 399.22815116, 285.4372319 ,
```

```
       164.41269107, 414.55938642, 288.54277941, 131.10031951,
       489.01980258, 508.36135985, 746.1706987 , 471.64422101,
       188.41149022, 125.04742227, 427.71271742, 311.46208768,
       303.53017615, 326.42871796, 198.76138617, 416.28622467])
```

1.4.1 5. Plot your results (Bonus if you use 3D plots). Show as many of your columns vs. credit rating that you can.

In [12]:
```
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D

fig = plt.figure()
ax = fig.add_subplot(1,1,1, projection='3d')
ax.view_init(25, 30)
ax.scatter(X.T[0], X.T[1], y, zdir='z', c='green')
ax.scatter(X.T[0], X.T[2], y, zdir='z', c='red')
ax.scatter(X.T[0], X.T[3], y, zdir='z', c='red')
ax.scatter(X.T[0], X.T[4], y, zdir='z', c='red')
ax.scatter(X.T[1], X.T[2], y, zdir='z', c='orange')
ax.scatter(X.T[1], X.T[3], y, zdir='z', c='orange')
ax.scatter(X.T[1], X.T[4], y, zdir='z', c='orange')
ax.scatter(X.T[2], X.T[3], y, zdir='z', c='yellow')
ax.scatter(X.T[2], X.T[4], y, zdir='z', c='yellow')
ax.scatter(X.T[3], X.T[4], y, zdir='z', c='purple')
ax.scatter(X.T[0], X.T[1], pred, zdir='z', c='blue')
ax.scatter(X.T[0], X.T[2], pred, zdir='z', c='blue')
ax.scatter(X.T[0], X.T[3], pred, zdir='z', c='blue')
ax.scatter(X.T[0], X.T[4], pred, zdir='z', c='blue')
ax.scatter(X.T[1], X.T[2], pred, zdir='z', c='blue')
ax.scatter(X.T[1], X.T[3], pred, zdir='z', c='blue')
ax.scatter(X.T[1], X.T[4], pred, zdir='z', c='blue')
ax.scatter(X.T[2], X.T[3], pred, zdir='z', c='blue')
ax.scatter(X.T[2], X.T[4], pred, zdir='z', c='blue')
ax.scatter(X.T[3], X.T[4], pred, zdir='z', c='blue')
```

Out[12]:
```
<mpl_toolkits.mplot3d.art3d.Path3DCollection at 0x117a8da58>
```


1000
800
600
400
200
0
2000
4000
6000
8000
10000
12000
14000
0
2000
4000
6000
8000
10000
12000
14000

In [13]: *# Cool.*