In [203]:
```
import pandas as pd
import numpy as np


# Use a csv file created in an earlier week, based on the original data set, but in which some problems have been resolved incl. categorical entries
# the health_NoCats now represents the corrected file with categorical data translated into numerical format
health = pd.read_csv('health_NoCats.csv', header = 0)  #this is the updated file with the more useable Age_Category which is ordered (unlike the original)
```

In [204]:
```
#define the list of features (columns) to be used as predictors
#note this subset is constructed to create greater independence between columns and to eliminate unhelpful features based on bar charts of those features vs. drug persistency

features = ['Gluco_Record_During_Rx', 'Dexa_During_Rx', 'Frag_Frac_During_Rx', 'Risk_Segment_During_Rx', 'Adherent_Flag', 'Idn_Indicator',
            'Injectable_Experience_During_Rx', 'Comorb_Encounter_For_Screening_For_Malignant_Neoplasms', 'Comorb_Encounter_For_Immunization',
            'Comorb_Encntr_For_General_Exam_W_O_Complaint_Susp_Or_Reprtd_Dx', 'Comorb_Vitamin_D_Deficiency',
            'Comorb_Other_Joint_Disorder_Not_Elsewhere_Classified', 'Comorb_Encntr_For_Oth_Sp_Exam_W_O_Complaint_Suspected_Or_Reprtd_Dx',
            'Comorb_Long_Term_Current_Drug_Therapy', 'Comorb_Dorsalgia', 'Comorb_Personal_History_Of_Other_Diseases_And_Conditions',
            'Comorb_Other_Disorders_Of_Bone_Density_And_Structure', 'Comorb_Disorders_of_lipoprotein_metabolism_and_other_lipidemias',
            'Comorb_Osteoporosis_without_current_pathological_fracture', 'Comorb_Personal_history_of_malignant_neoplasm',
            'Comorb_Gastro_esophageal_reflux_disease', 'Concom_Cholesterol_And_Triglyceride_Regulating_Preparations', 'Concom_Narcotics',
            'Concom_Systemic_Corticosteroids_Plain', 'Concom_Anti_Depressants_And_Mood_Stabilisers', 'Concom_Fluoroquinolones',
            'Concom_Cephalosporins', 'Concom_Macrolides_And_Similar_Types', 'Concom_Broad_Spectrum_Penicillins', 'Concom_Anaesthetics_General',
            'Concom_Viral_Vaccines', 'Risk_Rheumatoid_Arthritis', 'Risk_Untreated_Chronic_Hyperthyroidism', 'Risk_Untreated_Chronic_Hypogonadism',
            'Risk_Smoking_Tobacco', 'Risk_Chronic_Malnutrition_Or_Malabsorption', 'Risk_Chronic_Liver_Disease',  'Risk_Low_Calcium_Intake',
            'Risk_Vitamin_D_Insufficiency', 'Risk_Poor_Health_Frailty', 'Risk_Excessive_Thinness', 'Risk_Estrogen_Deficiency', 'Risk_Immobilization',
            'Dexa_Freq_During_Rx_Bucket_Flag','Change_RiskSeg_Worsened','Change_RiskSeg_Improved', 'Change_RiskSeg_Unk' , 'ChangedTScore_Worsened',
            'ChangedTScore_Improved', 'ChangedTScore_Unk', 'AsianRace_Flag',  'Midwest_Flag', 'Ntm_Speciality_MyBuckets1', 'Ntm_Speciality_MyBuckets2' ]

features1 = ['Persistency_Flag','Gluco_Record_During_Rx', 'Dexa_During_Rx', 'Frag_Frac_During_Rx', 'Risk_Segment_During_Rx', 'Adherent_Flag', 'Idn_Indicator',
            'Injectable_Experience_During_Rx', 'Comorb_Encounter_For_Screening_For_Malignant_Neoplasms', 'Comorb_Encounter_For_Immunization',
            'Comorb_Encntr_For_General_Exam_W_O_Complaint_Susp_Or_Reprtd_Dx', 'Comorb_Vitamin_D_Deficiency',
            'Comorb_Other_Joint_Disorder_Not_Elsewhere_Classified', 'Comorb_Encntr_For_Oth_Sp_Exam_W_O_Complaint_Suspected_Or_Reprtd_Dx',
            'Comorb_Long_Term_Current_Drug_Therapy', 'Comorb_Dorsalgia', 'Comorb_Personal_History_Of_Other_Diseases_And_Conditions',
            'Comorb_Other_Disorders_Of_Bone_Density_And_Structure', 'Comorb_Disorders_of_lipoprotein_metabolism_and_other_lipidemias']
features2 = ['Persistency_Flag', 'Comorb_Osteoporosis_without_current_pathological_fracture', 'Comorb_Personal_history_of_malignant_neoplasm',
            'Comorb_Gastro_esophageal_reflux_disease', 'Concom_Cholesterol_And_Triglyceride_Regulating_Preparations', 'Concom_Narcotics',
            'Concom_Systemic_Corticosteroids_Plain', 'Concom_Anti_Depressants_And_Mood_Stabilisers', 'Concom_Fluoroquinolones',
            'Concom_Cephalosporins', 'Concom_Macrolides_And_Similar_Types', 'Concom_Broad_Spectrum_Penicillins', 'Concom_Anaesthetics_General',
            'Concom_Viral_Vaccines', 'Risk_Rheumatoid_Arthritis', 'Risk_Untreated_Chronic_Hyperthyroidism', 'Risk_Untreated_Chronic_Hypogonadism']

features3 =['Persistency_Flag','Risk_Smoking_Tobacco', 'Risk_Chronic_Malnutrition_Or_Malabsorption', 'Risk_Chronic_Liver_Disease',  'Risk_Low_Calcium_Intake',
            'Risk_Vitamin_D_Insufficiency', 'Risk_Poor_Health_Frailty', 'Risk_Excessive_Thinness', 'Risk_Estrogen_Deficiency', 'Risk_Immobilization',
            'Dexa_Freq_During_Rx_Bucket_Flag','Change_RiskSeg_Worsened','Change_RiskSeg_Improved', 'Change_RiskSeg_Unk' , 'ChangedTScore_Worsened',
            'ChangedTScore_Improved', 'ChangedTScore_Unk', 'AsianRace_Flag',  'Midwest_Flag', 'Ntm_Speciality_MyBuckets1', 'Ntm_Speciality_MyBuckets2' ]
```

In [205]:
```
#examine correlations between variables in the dataset
health.corr()
```

Out[205]:

	Unnamed: 0	Persistency_Flag	Gluco_Record_During_Rx	Dexa_During_Rx	Frag_Frac_During_Rx	Risk_Segment_During_Rx	Adherent_Flag	Idn_Indicator	Injectable_Experience_
Unnamed: 0	1.000000	-0.020089	-0.000539	0.005562	0.064683	-0.158638	-0.059264	0.202579	
Persistency_Flag	-0.020089	1.000000	0.212704	0.491823	0.106935	-0.180535	-0.112488	0.111440	
Gluco_Record_During_Rx	-0.000539	0.212704	1.000000	0.118155	0.111802	-0.115432	-0.043668	0.143928	
Dexa_During_Rx	0.005562	0.491823	0.118155	1.000000	0.094189	-0.164109	-0.097857	0.037684	
Frag_Frac_During_Rx	0.064683	0.106935	0.111802	0.094189	1.000000	-0.099585	-0.036413	0.060766	
Risk_Segment_During_Rx	-0.158638	-0.180535	-0.115432	-0.164109	-0.099585	1.000000	0.042034	-0.075742	

	Unnamed: 0	Persistency_Flag	Gluco_Record_During_Rx	Dexa_During_Rx	Frag_Frac_During_Rx	Risk_Segment_During_Rx	Adherent_Flag	Idn_Indicator	Injectable_Experience_
Adherent_Flag	-0.059264	-0.112488	-0.043668	-0.097857	-0.036413	0.042034	1.000000	-0.036201	
Idn_Indicator	0.202579	0.111440	0.143928	0.037684	0.060766	-0.075742	-0.036201	1.000000	
Injectable_Experience_During_Rx	0.077701	0.098360	0.126182	0.047813	0.051375	-0.118066	-0.054218	0.275004	
Comorb_Encounter_For_Screening_For_Malignant_Neoplasms	0.004961	0.322320	0.085540	0.274016	0.027477	-0.110399	-0.073861	0.044808	
Comorb_Encounter_For_Immunization	-0.046723	0.314887	0.160775	0.220890	0.073253	-0.121517	-0.082041	0.009617	
Comorb_Encntr_For_General_Exam_W_O_Complaint_Susp_Or_Reprtd_Dx	-0.051591	0.289828	0.035044	0.221744	0.049954	-0.079622	-0.064621	-0.047617	
Comorb_Vitamin_D_Deficiency	0.044411	0.172664	0.078318	0.118377	0.072577	-0.134974	-0.059425	0.044318	
Comorb_Other_Joint_Disorder_Not_Elsewhere_Classified	0.050500	0.233279	0.195195	0.158440	0.175501	-0.111092	-0.063140	0.053129	
Comorb_Encntr_For_Oth_Sp_Exam_W_O_Complaint_Suspected_Or_Reprtd_Dx	0.093555	0.213413	0.060757	0.195537	0.064975	-0.101739	-0.034908	-0.021845	
Comorb_Long_Term_Current_Drug_Therapy	0.039351	0.352760	0.192533	0.239315	0.118376	-0.132883	-0.080464	0.091194	
Comorb_Dorsalgia	0.032122	0.215307	0.183116	0.162640	0.206900	-0.107558	-0.040205	0.048466	
Comorb_Personal_History_Of_Other_Diseases_And_Conditions	0.071528	0.219665	0.130936	0.157838	0.144734	-0.072421	-0.046182	-0.051559	
Comorb_Other_Disorders_Of_Bone_Density_And_Structure	0.007415	0.247283	0.062061	0.245781	0.002278	-0.048429	0.011805	0.062161	
Comorb_Disorders_of_lipoprotein_metabolism_and_other_lipidemias	-0.051369	0.163495	0.114144	0.126548	0.071557	-0.046738	-0.026207	0.029458	
Comorb_Osteoporosis_without_current_pathological_fracture	0.041262	0.139920	0.074010	0.188418	0.170048	-0.073021	-0.005022	0.051865	
Comorb_Personal_history_of_malignant_neoplasm	-0.001703	0.174835	0.089708	0.139750	0.024972	-0.068723	-0.000710	0.072544	
Comorb_Gastro_esophageal_reflux_disease	0.005773	0.220644	0.159194	0.150143	0.076687	-0.056891	-0.041876	0.006108	
Concom_Cholesterol_And_Triglyceride_Regulating_Preparations	-0.019787	0.125552	0.148686	0.077024	0.045169	-0.041832	-0.028828	0.039974	
Concom_Narcotics	0.026094	0.191910	0.304119	0.148717	0.213636	-0.104350	-0.071394	0.187778	
Concom_Systemic_Corticosteroids_Plain	-0.016107	0.242854	0.812460	0.124975	0.103953	-0.112794	-0.055696	0.137559	
Concom_Anti_Depressants_And_Mood_Stabilisers	-0.006682	0.110045	0.185106	0.080035	0.125701	-0.044951	-0.069920	0.086497	
Concom_Fluoroquinolones	-0.044459	0.186190	0.255898	0.121021	0.060634	-0.053714	-0.037059	0.064371	
Concom_Cephalosporins	-0.001783	0.221543	0.257867	0.137903	0.141984	-0.056628	-0.078879	0.107003	
Concom_Macrolides_And_Similar_Types	-0.018338	0.221611	0.266048	0.147506	0.084950	-0.073676	-0.054193	0.065915	
Concom_Broad_Spectrum_Penicillins	-0.009972	0.197854	0.159350	0.125429	0.065537	-0.070327	-0.071070	0.040500	
Concom_Anaesthetics_General	0.026469	0.222293	0.263647	0.173264	0.100068	-0.112476	-0.090429	0.238038	
Concom_Viral_Vaccines	-0.028383	0.222241	0.130839	0.106677	0.035268	-0.052925	-0.044572	0.120117	
Risk_Rheumatoid_Arthritis	0.012866	0.053809	0.130973	0.005015	0.028819	-0.017982	0.010942	-0.010832	
Risk_Untreated_Chronic_Hyperthyroidism	-0.029632	-0.018785	0.040424	-0.014828	-0.009003	-0.021308	0.005577	-0.041517	
Risk_Untreated_Chronic_Hypogonadism	-0.055988	0.067588	0.047516	0.042590	-0.016383	0.017056	0.009996	-0.013654	
Risk_Smoking_Tobacco	-0.082345	0.098045	0.122750	0.040167	0.072143	-0.032487	-0.025460	0.106676	
Risk_Chronic_Malnutrition_Or_Malabsorption	0.008764	0.049158	0.088984	0.048593	0.069443	-0.047029	-0.059102	0.009778	
Risk_Chronic_Liver_Disease	0.002648	0.018537	0.002367	0.018843	0.009976	-0.007081	-0.020109	0.023756	
Risk_Low_Calcium_Intake	0.015736	-0.009920	-0.012434	0.003088	-0.000934	-0.028685	-0.034863	-0.081549	
Risk_Vitamin_D_Insufficiency	0.042898	0.079782	0.055776	0.053493	0.080015	-0.086367	-0.040950	0.054127	
Risk_Poor_Health_Frailty	-0.003615	-0.045277	0.032912	-0.009929	0.021802	-0.068954	0.021450	-0.039069	
Risk_Excessive_Thinness	0.032335	-0.040138	-0.007899	-0.034610	0.056997	-0.043753	0.003709	-0.019564	
Risk_Estrogen_Deficiency	-0.011432	-0.012155	-0.010519	0.011499	-0.005360	0.033197	0.013096	-0.026285	
Risk_Immobilization	0.028778	-0.049787	-0.007149	-0.018763	-0.023861	0.008111	0.014781	0.026786	

	Unnamed: 0	Persistency_Flag	Gluco_Record_During_Rx	Dexa_During_Rx	Frag_Frac_During_Rx	Risk_Segment_During_Rx	Adherent_Flag	Idn_Indicator	Injectable_Experience_
Dexa_Freq_During_Rx_Bucket_Flag	-0.012554	-0.524993	-0.124154	-0.964760	-0.093429	0.173620	0.100805	-0.041150	
Change_RiskSeg_Worsened	0.063219	0.089614	0.057902	0.060062	0.146378	-0.168697	-0.020845	0.013236	
Change_RiskSeg_Improved	-0.003620	0.035594	0.009995	0.057295	-0.007593	-0.070878	0.018551	0.021609	
Change_RiskSeg_Unk	-0.092610	-0.084896	-0.060084	-0.085689	-0.096419	0.645355	0.012929	-0.003648	
ChangedTScore_Worsened	0.054493	0.115240	0.064858	0.124784	0.040490	-0.203322	-0.074632	0.051533	
ChangedTScore_Improved	-0.017119	0.112934	0.017193	0.137571	0.008482	-0.148085	-0.018366	0.027958	
ChangedTScore_Unk	-0.158638	-0.180535	-0.115432	-0.164109	-0.099585	1.000000	0.042034	-0.075742	
AsianRace_Flag	0.054213	0.036541	-0.064840	-0.008314	0.021763	-0.021791	-0.015135	-0.059610	
Midwest_Flag	-0.082046	-0.088015	-0.030173	0.003922	-0.024446	0.059205	0.002383	-0.013423	

In [206]:
```
#examine correlations between variables in the features list (subset)
#strongly correlated features should not be included together in classification models
health[features].corr()
```

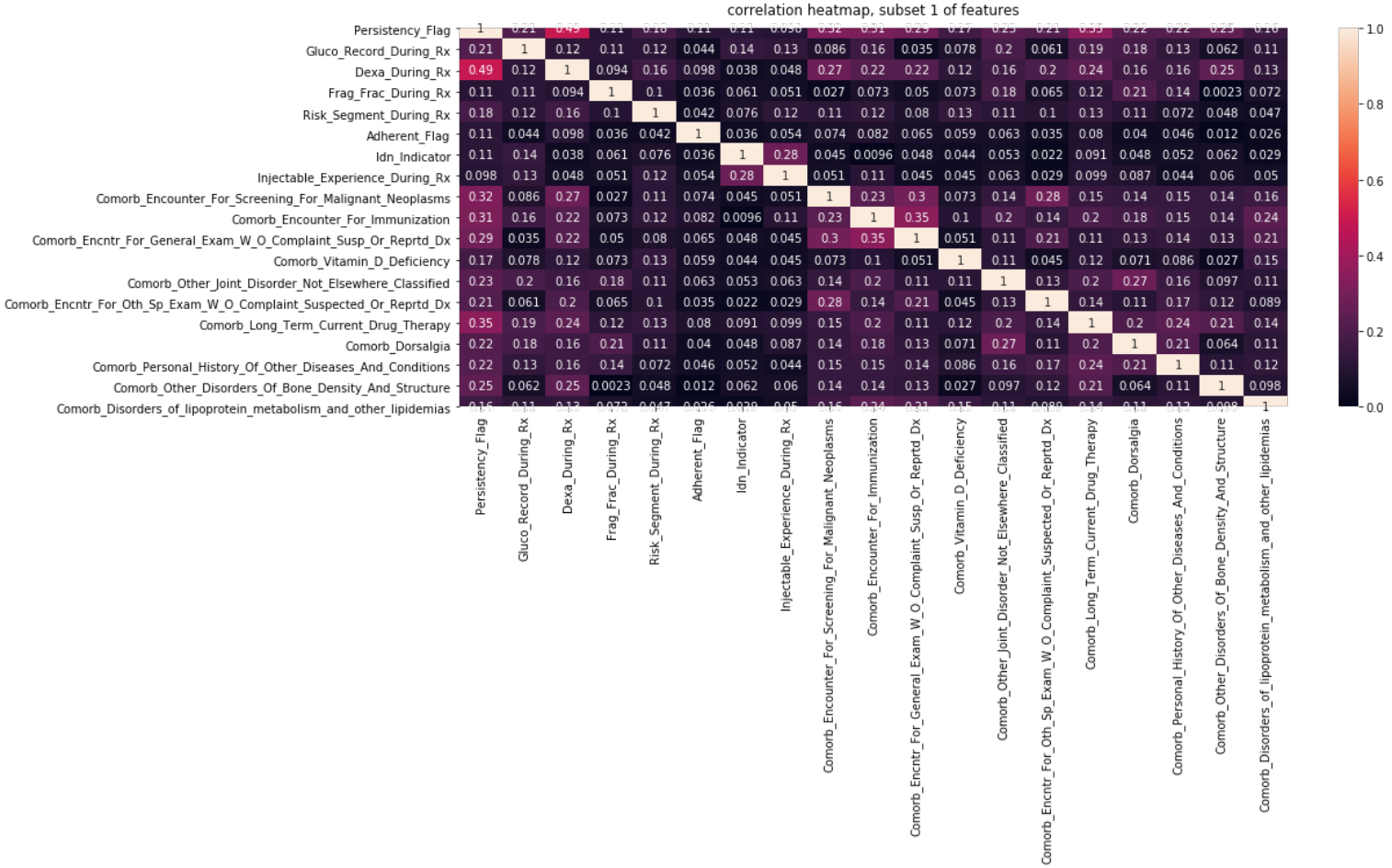
Out[206]:

	Gluco_Record_During_Rx	Dexa_During_Rx	Frag_Frac_During_Rx	Risk_Segment_During_Rx	Adherent_Flag	Idn_Indicator	Injectable_Experience_During_Rx	Comorb_Encounte
Gluco_Record_During_Rx	1.000000	0.118155	0.111802	-0.115432	-0.043668	0.143928	0.126182	
Dexa_During_Rx	0.118155	1.000000	0.094189	-0.164109	-0.097857	0.037684	0.047813	
Frag_Frac_During_Rx	0.111802	0.094189	1.000000	-0.099585	-0.036413	0.060766	0.051375	
Risk_Segment_During_Rx	-0.115432	-0.164109	-0.099585	1.000000	0.042034	-0.075742	-0.118066	
Adherent_Flag	-0.043668	-0.097857	-0.036413	0.042034	1.000000	-0.036201	-0.054218	
Idn_Indicator	0.143928	0.037684	0.060766	-0.075742	-0.036201	1.000000	0.275004	
Injectable_Experience_During_Rx	0.126182	0.047813	0.051375	-0.118066	-0.054218	0.275004	1.000000	
Comorb_Encounter_For_Screening_For_Malignant_Neoplasms	0.085540	0.274016	0.027477	-0.110399	-0.073861	0.044808	0.050749	
Comorb_Encounter_For_Immunization	0.160775	0.220890	0.073253	-0.121517	-0.082041	0.009617	0.105597	
Comorb_Encntr_For_General_Exam_W_O_Complaint_Susp_Or_Reprtd_Dx	0.035044	0.221744	0.049954	-0.079622	-0.064621	-0.047617	0.044964	
Comorb_Vitamin_D_Deficiency	0.078318	0.118377	0.072577	-0.134974	-0.059425	0.044318	0.045456	
Comorb_Other_Joint_Disorder_Not_Elsewhere_Classified	0.195195	0.158440	0.175501	-0.111092	-0.063140	0.053129	0.062998	
Comorb_Encntr_For_Oth_Sp_Exam_W_O_Complaint_Suspected_Or_Reprtd_Dx	0.060757	0.195537	0.064975	-0.101739	-0.034908	-0.021845	0.029116	
Comorb_Long_Term_Current_Drug_Therapy	0.192533	0.239315	0.118376	-0.132883	-0.080464	0.091194	0.099131	
Comorb_Dorsalgia	0.183116	0.162640	0.206900	-0.107558	-0.040205	0.048466	0.087103	
Comorb_Personal_History_Of_Other_Diseases_And_Conditions	0.130936	0.157838	0.144734	-0.072421	-0.046182	-0.051559	0.044420	
Comorb_Other_Disorders_Of_Bone_Density_And_Structure	0.062061	0.245781	0.002278	-0.048429	0.011805	0.062161	0.059666	
Comorb_Disorders_of_lipoprotein_metabolism_and_other_lipidemias	0.114144	0.126548	0.071557	-0.046738	-0.026207	0.029458	0.050372	
Comorb_Osteoporosis_without_current_pathological_fracture	0.074010	0.188418	0.170048	-0.073021	-0.005022	0.051865	0.035256	
Comorb_Personal_history_of_malignant_neoplasm	0.089708	0.139750	0.024972	-0.068723	-0.000710	0.072544	-0.000596	
Comorb_Gastro_esophageal_reflux_disease	0.159194	0.150143	0.076687	-0.056891	-0.041876	0.006108	0.060135	
Concom_Cholesterol_And_Triglyceride_Regulating_Preparations	0.148686	0.077024	0.045169	-0.041832	-0.028828	0.039974	0.065526	
Concom_Narcotics	0.304119	0.148717	0.213636	-0.104350	-0.071394	0.187778	0.118883	
Concom_Systemic_Corticosteroids_Plain	0.812460	0.124975	0.103953	-0.112794	-0.055696	0.137559	0.126648	
Concom_Anti_Depressants_And_Mood_Stabilisers	0.185106	0.080035	0.125701	-0.044951	-0.069920	0.086497	0.077846	
Concom_Fluoroquinolones	0.255898	0.121021	0.060634	-0.053714	-0.037059	0.064371	0.032623	

	Gluco_Record_During_Rx	Dexa_During_Rx	Frag_Frac_During_Rx	Risk_Segment_During_Rx	Adherent_Flag	Idn_Indicator	Injectable_Experience_During_Rx	Comorb_Encounte
Concom_Cephalosporins	0.257867	0.137903	0.141984	-0.056628	-0.078879	0.107003	0.101024	
Concom_Macrolides_And_Similar_Types	0.266048	0.147506	0.084950	-0.073676	-0.054193	0.065915	0.074294	
Concom_Broad_Spectrum_Penicillins	0.159350	0.125429	0.065537	-0.070327	-0.071070	0.040500	0.028719	
Concom_Anaesthetics_General	0.263647	0.173264	0.100068	-0.112476	-0.090429	0.238038	0.121577	
Concom_Viral_Vaccines	0.130839	0.106677	0.035268	-0.052925	-0.044572	0.120117	0.102146	
Risk_Rheumatoid_Arthritis	0.130973	0.005015	0.028819	-0.017982	0.010942	-0.010832	0.039334	
Risk_Untreated_Chronic_Hyperthyroidism	0.040424	-0.014828	-0.009003	-0.021308	0.005577	-0.041517	-0.030639	
Risk_Untreated_Chronic_Hypogonadism	0.047516	0.042590	-0.016383	0.017056	0.009996	-0.013654	-0.031687	
Risk_Smoking_Tobacco	0.122750	0.040167	0.072143	-0.032487	-0.025460	0.106676	0.024649	
Risk_Chronic_Malnutrition_Or_Malabsorption	0.088984	0.048593	0.069443	-0.047029	-0.059102	0.009778	0.028810	
Risk_Chronic_Liver_Disease	0.002367	0.018843	0.009976	-0.007081	-0.020109	0.023756	0.012187	
Risk_Low_Calcium_Intake	-0.012434	0.003088	-0.000934	-0.028685	-0.034863	-0.081549	-0.004163	
Risk_Vitamin_D_Insufficiency	0.055776	0.053493	0.080015	-0.086367	-0.040950	0.054127	0.041214	
Risk_Poor_Health_Frailty	0.032912	-0.009929	0.021802	-0.068954	0.021450	-0.039069	0.019000	
Risk_Excessive_Thinness	-0.007899	-0.034610	0.056997	-0.043753	0.003709	-0.019564	0.021792	
Risk_Estrogen_Deficiency	-0.010519	0.011499	-0.005360	0.033197	0.013096	-0.026285	-0.030290	
Risk_Immobilization	-0.007149	-0.018763	-0.023861	0.008111	0.014781	0.026786	0.022235	
Dexa_Freq_During_Rx_Bucket_Flag	-0.124154	-0.964760	-0.093429	0.173620	0.100805	-0.041150	-0.050431	
Change_RiskSeg_Worsened	0.057902	0.060062	0.146378	-0.168697	-0.020845	0.013236	0.020453	
Change_RiskSeg_Improved	0.009995	0.057295	-0.007593	-0.070878	0.018551	0.021609	0.027906	
Change_RiskSeg_Unk	-0.060084	-0.085689	-0.096419	0.645355	0.012929	-0.003648	-0.056252	
ChangedTScore_Worsened	0.064858	0.124784	0.040490	-0.203322	-0.074632	0.051533	0.028387	
ChangedTScore_Improved	0.017193	0.137571	0.008482	-0.148085	-0.018366	0.027958	0.023677	
ChangedTScore_Unk	-0.115432	-0.164109	-0.099585	1.000000	0.042034	-0.075742	-0.118066	
AsianRace_Flag	-0.064840	-0.008314	0.021763	-0.021791	-0.015135	-0.059610	-0.005925	
Midwest_Flag	-0.030173	0.003922	-0.024446	0.059205	0.002383	-0.013423	-0.008379	
Ntm_Speciality_MyBuckets1	0.013319	0.156939	0.005105	-0.086753	-0.007226	0.094120	0.053567	
Ntm_Speciality_MyBuckets2	-0.014899	-0.015208	0.005556	-0.038862	-0.001784	0.044423	0.043375	

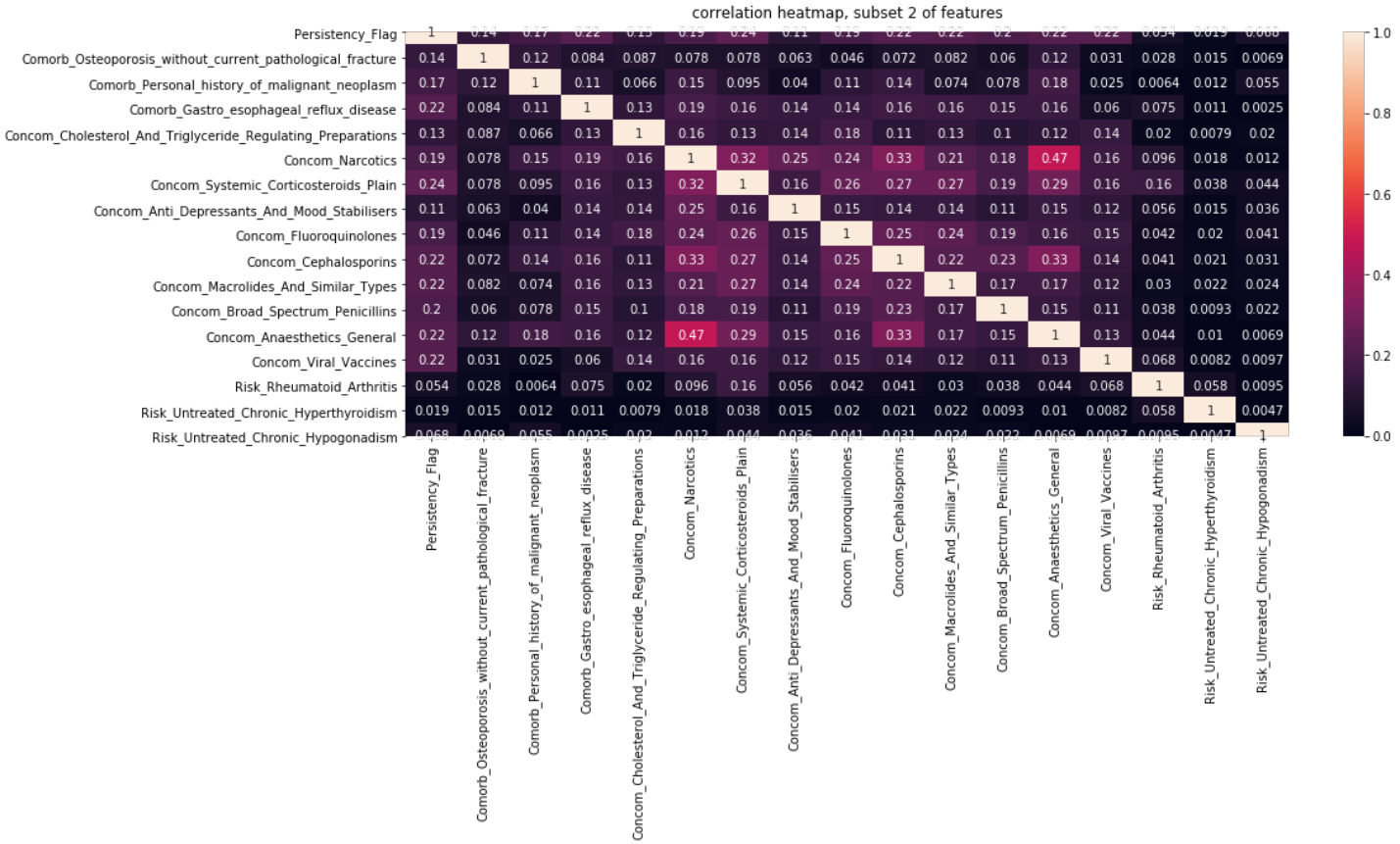
In [207]:
```
import seaborn as sns
import matplotlib.pyplot as plt
plt.figure(figsize=(16,6))
heatmap = sns.heatmap(abs(health[features1].corr()), vmin = 0, vmax = 1, annot = True)
heatmap.set_title("correlation heatmap, subset 1 of features", fontdict= {'fontsize' :12}, pad = 12)
```

Out[207]: Text(0.5, 1, 'correlation heatmap, subset 1 of features')



In [208]:
```
plt.figure(figsize=(16,6))
heatmap = sns.heatmap(abs(health[features2].corr()), vmin = 0, vmax = 1, annot = True)
heatmap.set_title("correlation heatmap, subset 2 of features", fontdict= {'fontsize' :12}, pad = 12)
```

Out[208]:
```
Text(0.5, 1, 'correlation heatmap, subset 2 of features')
```



In [209]:
```
plt.figure(figsize=(16,6))
heatmap = sns.heatmap(abs(health[features3].corr()), vmin = 0, vmax = 1, annot = True)
heatmap.set_title("correlation heatmap, subset 3 of features", fontdict= {'fontsize' :12}, pad = 12)
```

Out[209]:
```
Text(0.5, 1, 'correlation heatmap, subset 3 of features')
```

In [210]:
```
# import the sklearn package for use in log regression, import confusion matrix
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import confusion_matrix

# instantiate the model
logreg = LogisticRegression()
```

In [211]:
```
X = health[features]
y = health.Persistency_Flag


# split X and y into training and testing sets
from sklearn.model_selection import train_test_split
X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.25,random_state=27)
```

In [212]:
```
# fit the model with data - cannot run it yet, until variables are transformed
logreg.fit(X_train,y_train)

#
y_pred=logreg.predict(X_test)

confusion_matrix = confusion_matrix(y_test, y_pred)
```

```
print(confusion_matrix)
print("Accuracy:",metrics.accuracy_score(y_test, y_pred))
```

```
[[481  73]
 [ 87 215]]
Accuracy: 0.8130841121495327

/Users/jen/opt/anaconda3/lib/python3.7/site-packages/sklearn/linear_model/logistic.py:432: FutureWarning: Default solver will be changed to 'lbfgs' in 0.22. Specify a solver to
silence this warning.
  FutureWarning)
```

In [213]:
```
#statsmodels provides more information by way of summary()
#we can use this to refine the columns further (eliminating some)
#we are looking to keep those columns (predictor variables) with low values in the 'P>[t]'' column of the summary, or the ones where 'coef' has larger absolute value

import statsmodels.api as sm
X_train = sm.add_constant(X_train)
lm_2 = sm.OLS(y_train, X_train).fit()
lm_2.summary()
```

```
/Users/jen/opt/anaconda3/lib/python3.7/site-packages/numpy/core/fromnumeric.py:2495: FutureWarning: Method .ptp is deprecated and will be removed in a future version. Use nump
y.ptp instead.
  return ptp(axis=axis, out=out, **kwargs)
```

Out[213]:

OLS Regression Results

Dep. Variable:	Persistency_Flag	**R-squared:**	0.459
Model:	OLS	**Adj. R-squared:**	0.448
Method:	Least Squares	**F-statistic:**	40.28
Date:	Sat, 11 Dec 2021	**Prob (F-statistic):**	1.19e-291
Time:	16:52:03	**Log-Likelihood:**	-1004.0
No. Observations:	2568	**AIC:**	2116.
Df Residuals:	2514	**BIC:**	2432.
Df Model:	53		
Covariance Type:	nonrobust		

In [214]:
```
#now redefine features eliminating all those with importance vals from the list above less than 0.01
features = ['Gluco_Record_During_Rx', 'Dexa_During_Rx', 'Frag_Frac_During_Rx', 'Risk_Segment_During_Rx', 'Adherent_Flag', 'Idn_Indicator',
            'Injectable_Experience_During_Rx', 'Comorb_Encounter_For_Screening_For_Malignant_Neoplasms', 'Comorb_Encounter_For_Immunization',
            'Comorb_Encntr_For_General_Exam_W_O_Complaint_Susp_Or_Reprtd_Dx', 'Comorb_Vitamin_D_Deficiency',
            'Comorb_Other_Joint_Disorder_Not_Elsewhere_Classified', 'Comorb_Encntr_For_Oth_Sp_Exam_W_O_Complaint_Suspected_Or_Reprtd_Dx',
            'Comorb_Long_Term_Current_Drug_Therapy', 'Comorb_Dorsalgia', 'Comorb_Personal_History_Of_Other_Diseases_And_Conditions',
            'Comorb_Other_Disorders_Of_Bone_Density_And_Structure', 'Comorb_Disorders_of_lipoprotein_metabolism_and_other_lipidemias',
            'Comorb_Osteoporosis_without_current_pathological_fracture', 'Comorb_Personal_history_of_malignant_neoplasm',
            'Comorb_Gastro_esophageal_reflux_disease', 'Concom_Cholesterol_And_Triglyceride_Regulating_Preparations',
            'Concom_Narcotics', 'Concom_Systemic_Corticosteroids_Plain', 'Concom_Anti_Depressants_And_Mood_Stabilisers', 'Concom_Fluoroquinolones',
            'Concom_Cephalosporins', 'Concom_Macrolides_And_Similar_Types', 'Concom_Broad_Spectrum_Penicillins', 'Concom_Anaesthetics_General',
            'Concom_Viral_Vaccines', 'Risk_Smoking_Tobacco', 'Risk_Chronic_Malnutrition_Or_Malabsorption','Risk_Vitamin_D_Insufficiency',
            'Dexa_Freq_During_Rx_Bucket_Flag', 'Change_RiskSeg_Unk',
            'ChangedTScore_Unk', 'Midwest_Flag', 'Ntm_Speciality_MyBuckets1' ]

#rebuild the test and training sets with this new reduced set of features and without the extra column used by stats models
X = health[features]
y = health.Persistency_Flag

# split X and y into training and testing sets
from sklearn.model_selection import train_test_split
X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.25,random_state=27)
```

In [215]:
```
from sklearn.ensemble import RandomForestClassifier



#Create a Gaussian Classifier
clf=RandomForestClassifier(n_estimators=100)

#Train the model using the training sets y_pred=clf.predict(X_test)
clf.fit(X_train,y_train)

# prediction on test set
y_pred=clf.predict(X_test)

# Import scikit-learn metrics module for accuracy calculation
from sklearn import metrics
# Model Accuracy, how often is the classifier correct?
print("Accuracy:",metrics.accuracy_score(y_test, y_pred))
```

```
Accuracy: 0.7990654205607477
```

In [216]:
```
# the above classifier also affords the opportunity to rank the predictor variables by importance (according to this classification method)

feature_imp = pd.Series(clf.feature_importances_,index=features).sort_values(ascending=False)
feature_imp
```

Out[216]:

```
Dexa_During_Rx                                                  0.102123
Dexa_Freq_During_Rx_Bucket_Flag                                 0.090829
Comorb_Encounter_For_Screening_For_Malignant_Neoplasms          0.050117
Comorb_Long_Term_Current_Drug_Therapy                           0.049187
```

In [217]:
```
#now redefine features eliminating all those with importance vals from the list above less than 0.01
features = ['Gluco_Record_During_Rx', 'Dexa_During_Rx', 'Frag_Frac_During_Rx', 'Risk_Segment_During_Rx', 'Adherent_Flag', 'Idn_Indicator',
            'Injectable_Experience_During_Rx', 'Comorb_Encounter_For_Screening_For_Malignant_Neoplasms', 'Comorb_Encounter_For_Immunization',
            'Comorb_Encntr_For_General_Exam_W_O_Complaint_Susp_Or_Reprtd_Dx', 'Comorb_Vitamin_D_Deficiency', 'Comorb_Other_Joint_Disorder_Not_Elsewhere_Classified',
            'Comorb_Encntr_For_Oth_Sp_Exam_W_O_Complaint_Suspected_Or_Reprtd_Dx', 'Comorb_Long_Term_Current_Drug_Therapy',
            'Comorb_Dorsalgia', 'Comorb_Personal_History_Of_Other_Diseases_And_Conditions', 'Comorb_Other_Disorders_Of_Bone_Density_And_Structure',
            'Comorb_Disorders_of_lipoprotein_metabolism_and_other_lipidemias', 'Comorb_Osteoporosis_without_current_pathological_fracture',
            'Comorb_Personal_history_of_malignant_neoplasm', 'Comorb_Gastro_esophageal_reflux_disease', 'Concom_Cholesterol_And_Triglyceride_Regulating_Preparations',
            'Concom_Narcotics', 'Concom_Systemic_Corticosteroids_Plain', 'Concom_Anti_Depressants_And_Mood_Stabilisers', 'Concom_Fluoroquinolones',
            'Concom_Cephalosporins', 'Concom_Macrolides_And_Similar_Types', 'Concom_Broad_Spectrum_Penicillins', 'Concom_Anaesthetics_General', 'Concom_Viral_Vaccines',
            'Risk_Smoking_Tobacco', 'Risk_Chronic_Malnutrition_Or_Malabsorption','Risk_Vitamin_D_Insufficiency', 'Dexa_Freq_During_Rx_Bucket_Flag', 'Change_RiskSeg_Unk',
            'ChangedTScore_Unk', 'Midwest_Flag', 'Ntm_Speciality_MyBuckets1' ]
```

In [218]:
```
#redefine the predictor variables with now smaller set of features

X = health[features]
y = health.Persistency_Flag

#set training and test sets accordingly
X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.25,random_state=27)

#Create a Gaussian Classifier
clf=RandomForestClassifier(n_estimators=100)

#Train the model using the training sets y_pred=clf.predict(X_test)
clf.fit(X_train,y_train)

# prediction on test set
y_pred=clf.predict(X_test)

#Import scikit-learn metrics module for accuracy calculation
from sklearn import metrics
# Model Accuracy, how often is the classifier correct?
print("Accuracy:",metrics.accuracy_score(y_test, y_pred))
```

```
Accuracy: 0.7990654205607477
```

In [219]:
```
# Note - the accuracy with random forest on fewer predictors is very close to what it was with the larger set of predictors (within 2 %)
```

In [220]:
```
#now with log reg on smaller set of features (predictors)
# fit the model with data
# instantiate the model
logreg = LogisticRegression()
logreg.fit(X_train,y_train)

#
y_pred=logreg.predict(X_test)

#confusion matrix
from sklearn.metrics import confusion_matrix

confusion_matrix = confusion_matrix(y_test, y_pred)
print(confusion_matrix)

print("Accuracy:",metrics.accuracy_score(y_test, y_pred))
```

```
[[480  74]
 [ 87 215]]
Accuracy: 0.8119158878504673

/Users/jen/opt/anaconda3/lib/python3.7/site-packages/sklearn/linear_model/logistic.py:432: FutureWarning: Default solver will be changed to 'lbfgs' in 0.22. Specify a solver to
silence this warning.
  FutureWarning)
```

In [221]:
```
from sklearn.ensemble import AdaBoostClassifier #For Classification
from sklearn.ensemble import AdaBoostRegressor #For Regression
from sklearn.tree import DecisionTreeClassifier
dtree = DecisionTreeClassifier()
cl = AdaBoostClassifier(n_estimators=100, base_estimator=dtree,learning_rate=1)
cl.fit(X_train,y_train)

# prediction on test set
y_pred=cl.predict(X_test)


# Model Accuracy, how often is the classifier correct?
print("Accuracy:",metrics.accuracy_score(y_test, y_pred))
```

```
Accuracy: 0.7920560747663551
```

In [222]:
```
from sklearn.ensemble import GradientBoostingClassifier #For Classification
from sklearn.ensemble import GradientBoostingRegressor #For Regression
cl = GradientBoostingClassifier(n_estimators=100, learning_rate=1.0, max_depth=1)
cl.fit(X_train, y_train)

# prediction on test set
y_pred=cl.predict(X_test)


# Model Accuracy, how often is the classifier correct?
print("Accuracy:",metrics.accuracy_score(y_test, y_pred))
```

```
Accuracy: 0.8130841121495327
```

In [223]:
```
# to run the next one I had to install a new package
# install -c anaconda py-xgboost
```

In [224]:
```

from xgboost import XGBClassifier
xgbc = XGBClassifier()

xgbc.fit(X_train, y_train)
```

```
# prediction on test set
y_pred=xgbc.predict(X_test)


# Model Accuracy, how often is the classifier correct?
print("Accuracy:",metrics.accuracy_score(y_test, y_pred))
```

```
Accuracy: 0.8165887850467289
```

In [225]:
```
# in order to use the next classifiers, that take into account imbalanced classes, I had to install imbalanced-learn

#conda install -c conda-forge imbalanced-learn
```

In [226]:
```
# This classifier addresses the imbalanced classes (unequal persistent vs. non-persistent) producing substantially superior results to earlier efforts
# that ignored the class imbalance

# bagged decision trees with random undersampling for imbalanced classification
from numpy import mean
from sklearn.datasets import make_classification
from sklearn.model_selection import cross_val_score
from sklearn.model_selection import RepeatedStratifiedKFold
from imblearn.ensemble import BalancedBaggingClassifier

# define model
model = BalancedBaggingClassifier()



# generate dataset
X, y = make_classification(n_samples=10000, n_features=2, n_redundant=0,
    n_clusters_per_class=1, weights=[0.99], flip_y=0, random_state=27)
# define model
model = BalancedBaggingClassifier()
# define evaluation procedure
cv = RepeatedStratifiedKFold(n_splits=10, n_repeats=3, random_state=27)
# evaluate model
scores = cross_val_score(model, X, y, scoring='roc_auc', cv=cv, n_jobs=-1)
# summarize performance
print('Mean ROC AUC: %.2f' % mean(scores))

scores = cross_val_score(model, X, y, scoring='accuracy', cv=cv, n_jobs=-1)
# summarize performance
print('Mean Accuracy: %.2f' % mean(scores))
```

```
Mean ROC AUC: 1.00
Mean Accuracy: 1.00
```

In [227]:
```
# Random  Forest with random undersampling for Imbalanced Classif.
#This is another classifier that addresses the imbalanced classes (unequal persistent vs. non-persistent)
from numpy import mean
from sklearn.datasets import make_classification
from sklearn.model_selection import cross_val_score
from sklearn.model_selection import RepeatedStratifiedKFold
from imblearn.ensemble import BalancedRandomForestClassifier


# define model
model = BalancedRandomForestClassifier(n_estimators=10)


# generate dataset
X, y = make_classification(n_samples=10000, n_features=2, n_redundant=0,
```

```
    n_clusters_per_class=1, weights=[0.99], flip_y=0, random_state=27)
# define model
model = BalancedRandomForestClassifier(n_estimators=10)
# define evaluation procedure
cv = RepeatedStratifiedKFold(n_splits=10, n_repeats=3, random_state=1)
# evaluate model
scores = cross_val_score(model, X, y, scoring='roc_auc', cv=cv, n_jobs=-1)
# summarize performance
print('Mean ROC AUC: %.2f' % mean(scores))


# evaluate model
scores = cross_val_score(model, X, y, scoring='accuracy', cv=cv, n_jobs=-1)
# summarize performance
print('Mean accuracy: %.2f' % mean(scores))
```

```
Mean ROC AUC: 1.00
Mean accuracy: 1.00
```

In [228]:
```
print("Best classifier on this data appears to be the BalancedBaggingClassifier for imbalanced classes from imblearn.ensemble ")
```

```
Best classifier on this data appears to be the BalancedBaggingClassifier for imbalanced classes from imblearn.ensemble
```

In [ ]:

In [ ]: