Assignment 7
Prostate Cancer Identification

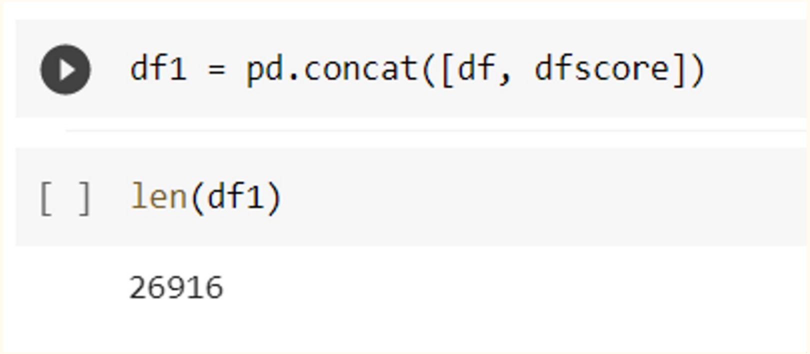
Group 2 : Siraalak, Puthali, Siddartha, Arofat

Initial Handling of Training And Test Data

We started with combining data to make sure any adjustments in training will be applied to test data as well.

The result was 26916 instances and 33 variables.

Initially, we were hesitant to change the variables as it was mentioned in the assignment not to change.



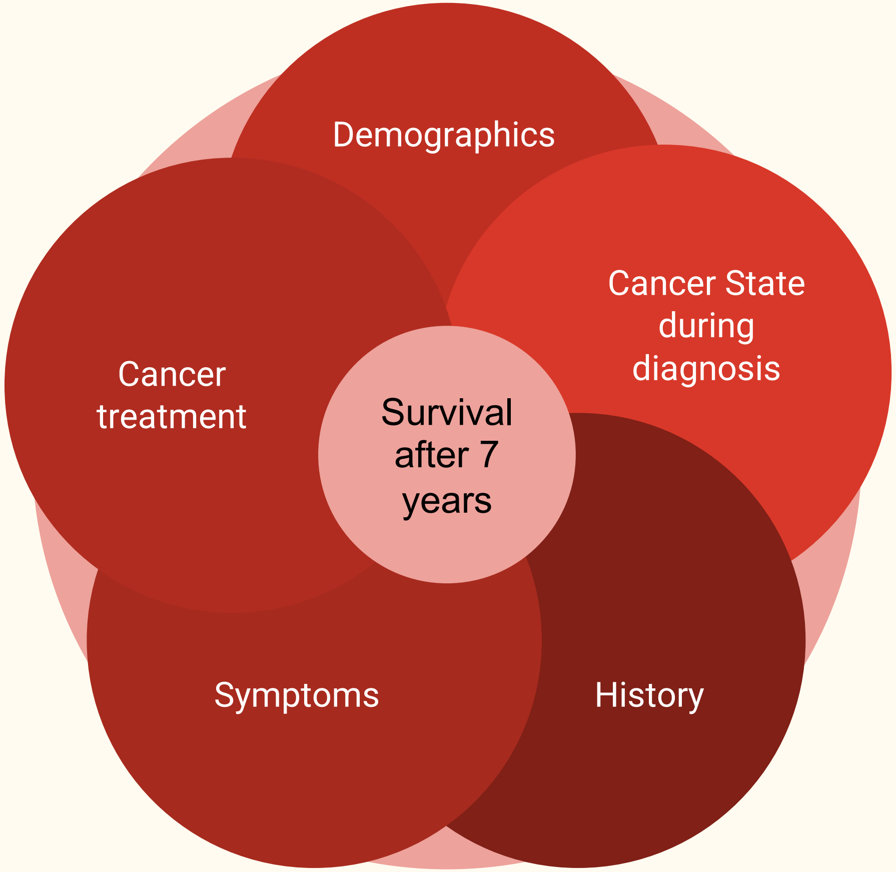
```
df1 = pd.concat([df, dfscore])
```

```
[ ] len(df1)
```

```
26916
```

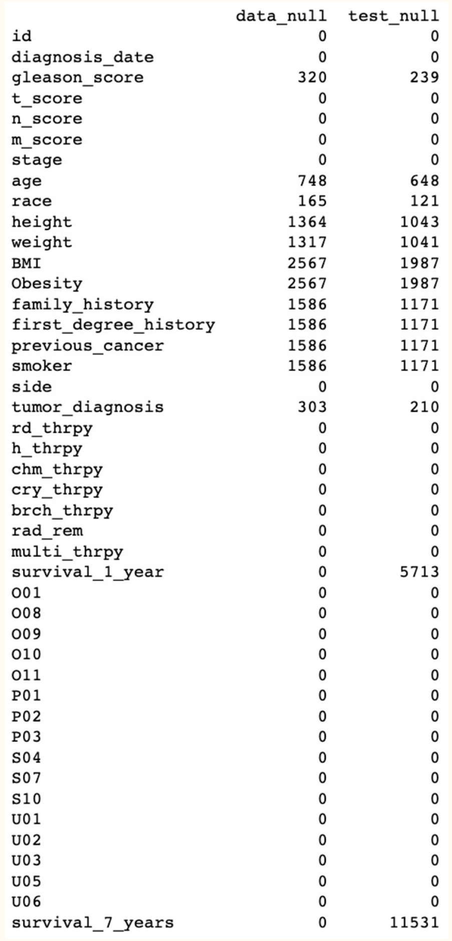
Feature Selection

1. Demographics:
   - Age
   - Race
   - Obesity
1. Cancer treatment
   - Kind of Therapy
   - Surgery
1. Symptoms
   - All Symptoms
1. History
   - Family history
   - Smoker
   - Previous Cancer diagnosis
1. Cancer State during diagnosis
   - Stage
   - Tumor size
   - Gleason score
   - T_score, n_score, m_score
   - side



EDA - Univariate Analysis

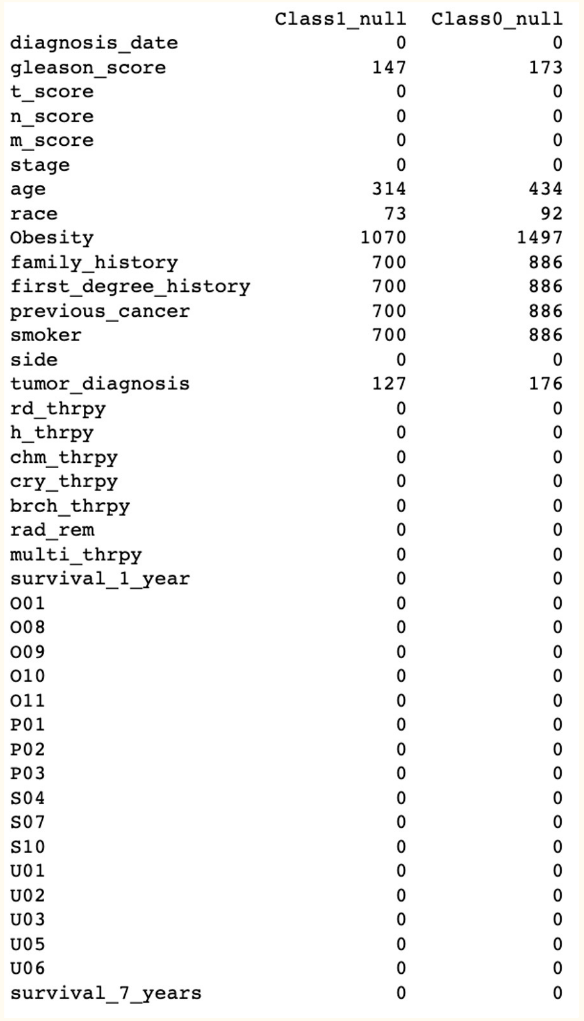
- Checking for null-values in both training and test data
- We can see most of the columns with null values are common in Train and test data.
- We can see that survival_1_year flag has a lot of null values in test data. We chose to exclude it from our data.
- In addition, we created new binary feature called Obesity based on Height (inches) and Weight (lbs). (BMI > 30 - 1 else 0)



```
                        data_null  test_null
id                              0          0
diagnosis_date                  0          0
gleason_score                 320        239
t_score                         0          0
n_score                         0          0
m_score                         0          0
stage                           0          0
age                           748        648
race                          165        121
height                       1364       1043
weight                       1317       1041
BMI                          2567       1987
Obesity                      2567       1987
family_history               1586       1171
first_degree_history         1586       1171
previous_cancer              1586       1171
smoker                       1586       1171
side                            0          0
tumor_diagnosis               303        210
rd_thrpy                        0          0
h_thrpy                         0          0
chm_thrpy                       0          0
cry_thrpy                       0          0
brch_thrpy                      0          0
rad_rem                         0          0
multi_thrpy                     0          0
survival_1_year                 0       5713
O01                             0          0
O08                             0          0
O09                             0          0
O10                             0          0
O11                             0          0
P01                             0          0
P02                             0          0
P03                             0          0
S04                             0          0
S07                             0          0
S10                             0          0
U01                             0          0
U02                             0          0
U03                             0          0
U05                             0          0
U06                             0          0
survival_7_years                0      11531
```

Univariate Analysis

- We divided the data by classes of target variable
- Checking the null values in both classes, We see similar distribution of nulls among variables across both classes
- Therefore, we might have to classify null values as a separate category which can be used in predicting test data.

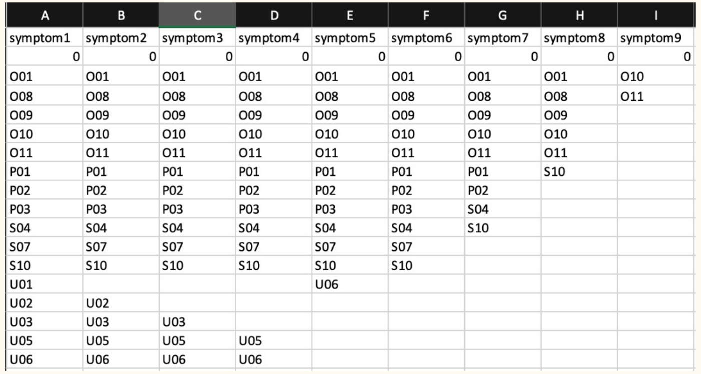


```
                      Class1_null  Class0_null
diagnosis_date                  0            0
gleason_score                 147          173
t_score                         0            0
n_score                         0            0
m_score                         0            0
stage                           0            0
age                           314          434
race                           73           92
Obesity                      1070         1497
family_history                700          886
first_degree_history          700          886
previous_cancer               700          886
smoker                        700          886
side                            0            0
tumor_diagnosis               127          176
rd_thrpy                        0            0
h_thrpy                         0            0
chm_thrpy                       0            0
cry_thrpy                       0            0
brch_thrpy                      0            0
rad_rem                         0            0
multi_thrpy                     0            0
survival_1_year                 0            0
O01                             0            0
O08                             0            0
O09                             0            0
O10                             0            0
O11                             0            0
P01                             0            0
P02                             0            0
P03                             0            0
S04                             0            0
S07                             0            0
S10                             0            0
U01                             0            0
U02                             0            0
U03                             0            0
U05                             0            0
U06                             0            0
survival_7_years                0            0
```

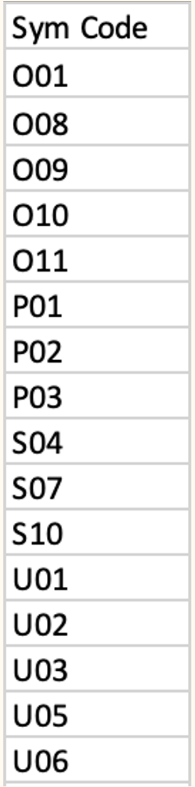
Feature engineering - Symptoms

Create features for each type of symptoms:

- Splitting the symptoms column by the delimiter ‘,’
- There are a maximum of 9 symptoms for a patient
- Finding the unique symptoms present
- Now create new symptom features for all the 16 symptoms.



A	B	C	D	E	F	G	H	I
symptom1	symptom2	symptom3	symptom4	symptom5	symptom6	symptom7	symptom8	symptom9
0	0	0	0	0	0	0	0	0
O01	O01	O01	O01	O01	O01	O01	O01	O10
O08	O08	O08	O08	O08	O08	O08	O08	O11
O09	O09	O09	O09	O09	O09	O09	O09	
O10	O10	O10	O10	O10	O10	O10	O10	
O11	O11	O11	O11	O11	O11	O11	O11	
P01	P01	P01	P01	P01	P01	P01	S10	
P02	P02	P02	P02	P02	P02	P02		
P03	P03	P03	P03	P03	P03	S04		
S04	S04	S04	S04	S04	S04	S10		
S07	S07	S07	S07	S07	S07			
S10	S10	S10	S10	S10	S10			
U01				U06				
U02	U02							
U03	U03	U03						
U05	U05	U05	U05					
U06	U06	U06	U06					

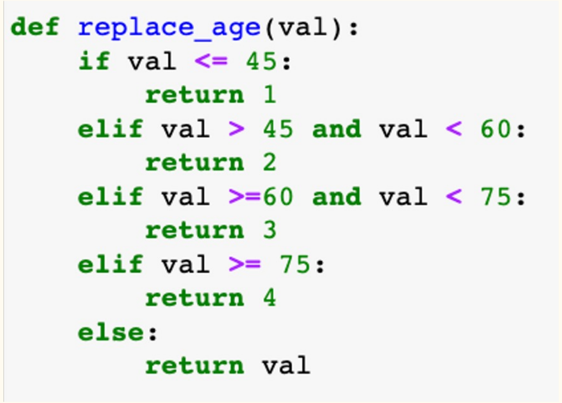


Sym Code
O01
O08
O09
O10
O11
P01
P02
P03
S04
S07
S10
U01
U02
U03
U05
U06

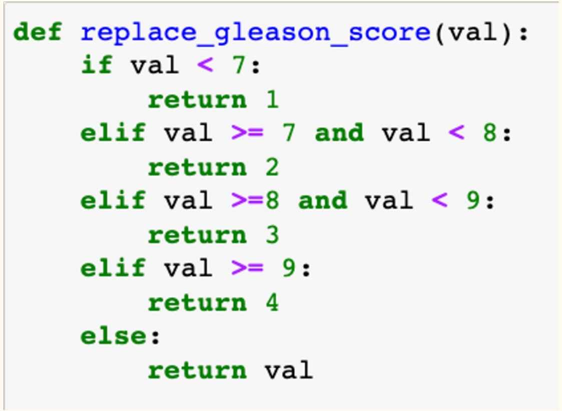
Feature Engineering - Age and Gleason score

- We categorised the age into categories:
  1. Age_group = 1 if age <= 45
  2. Age_group = 2 if 45 <= age < 60
  3. Age_group = 3 if 60 <= age < 75
  4. Age_group = 4 if age >= 75

- We categorised Gleason score into gleason grades:
  1. Gleason grade = 1 if gleason score < 7
  2. Gleason grade = 2 if 7 <= gleason score < 8
  3. Gleason grade = 3 if 8 <= gleason score < 9
  4. Gleason grade = 4 if gleason score >= 9



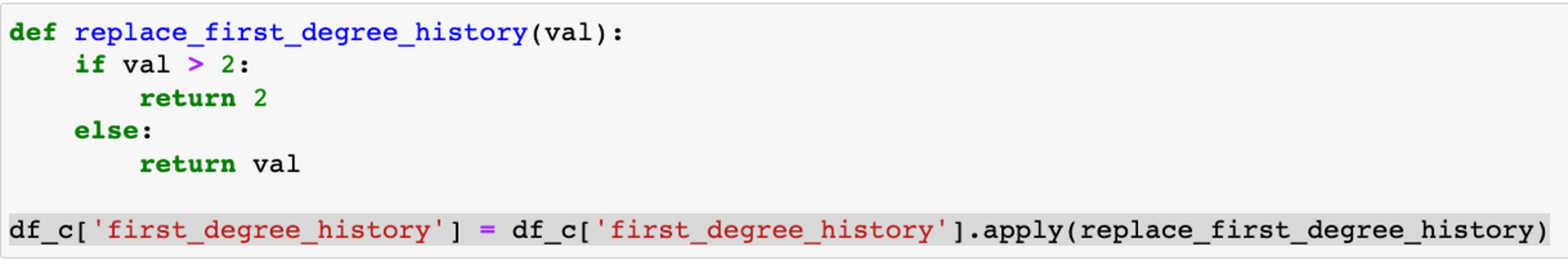
```python
def replace_age(val):
    if val <= 45:
        return 1
    elif val > 45 and val < 60:
        return 2
    elif val >=60 and val < 75:
        return 3
    elif val >= 75:
        return 4
    else:
        return val
```



```python
def replace_gleason_score(val):
    if val < 7:
        return 1
    elif val >= 7 and val < 8:
        return 2
    elif val >=8 and val < 9:
        return 3
    elif val >= 9:
        return 4
    else:
        return val
```

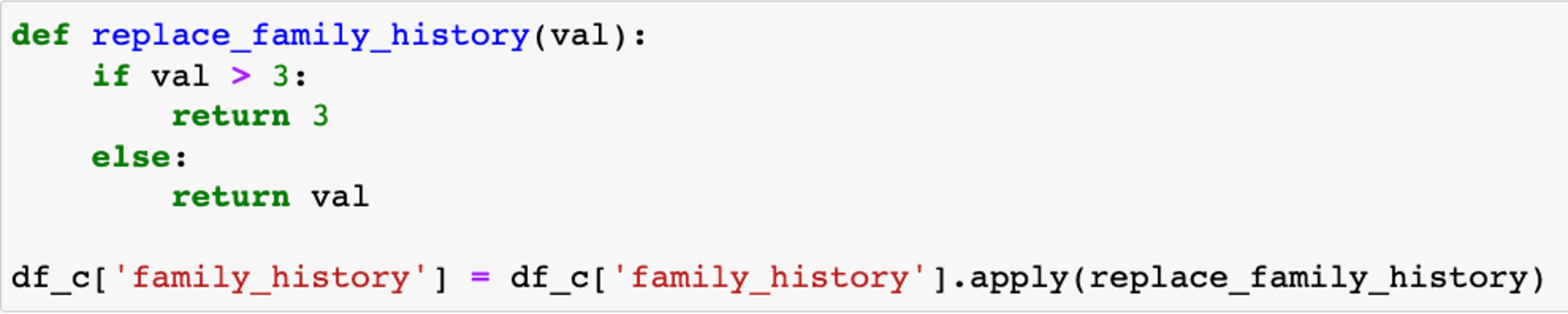
Feature Engineering - Family history and First degree

- Reduced the categories in family history to 5 (0,1,2,3,NA)
- Reduced the categories in First degree history to 4 (0,1,2,3,NA)



```python
def replace_first_degree_history(val):
    if val > 2:
        return 2
    else:
        return val

df_c['first_degree_history'] = df_c['first_degree_history'].apply(replace_first_degree_history)
```

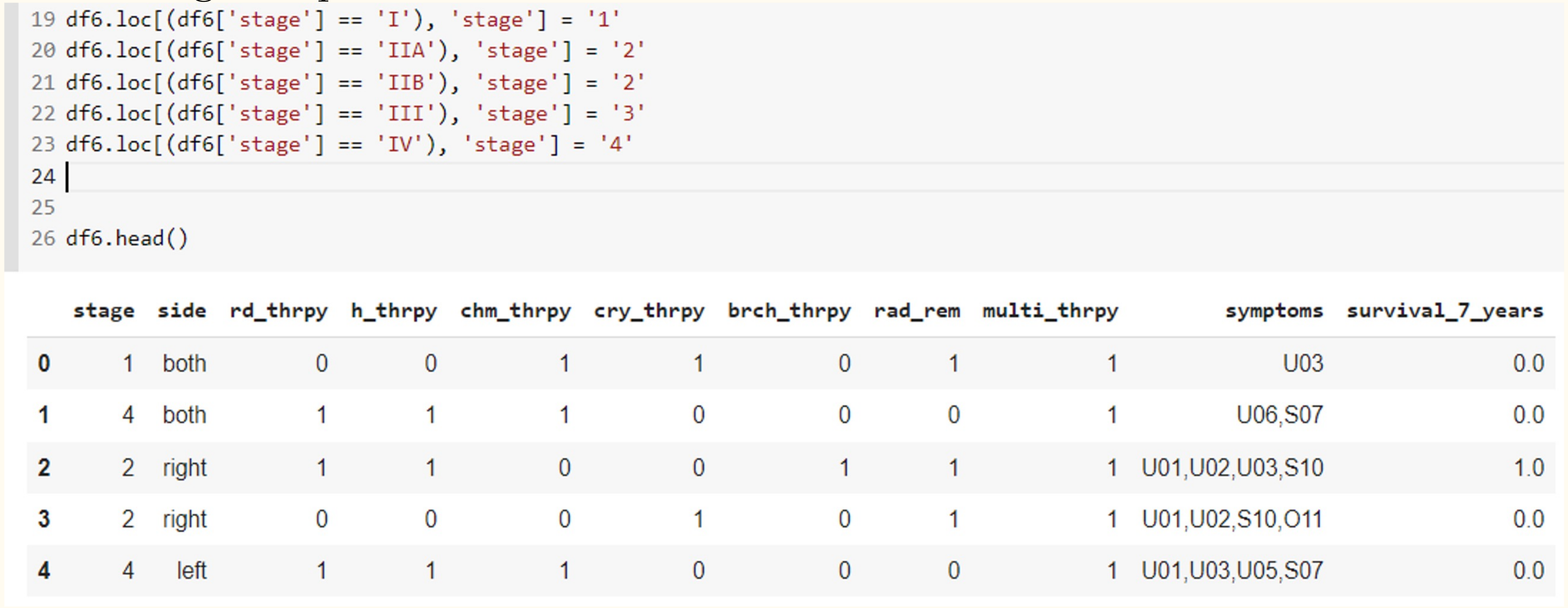


```python
def replace_family_history(val):
    if val > 3:
        return 3
    else:
        return val

df_c['family_history'] = df_c['family_history'].apply(replace_family_history)
```

Feature engineering - stage

Change stage to numbers. Here we changed to IIA and IIB to 2 only as it is one of the four stages of prostate cancer.



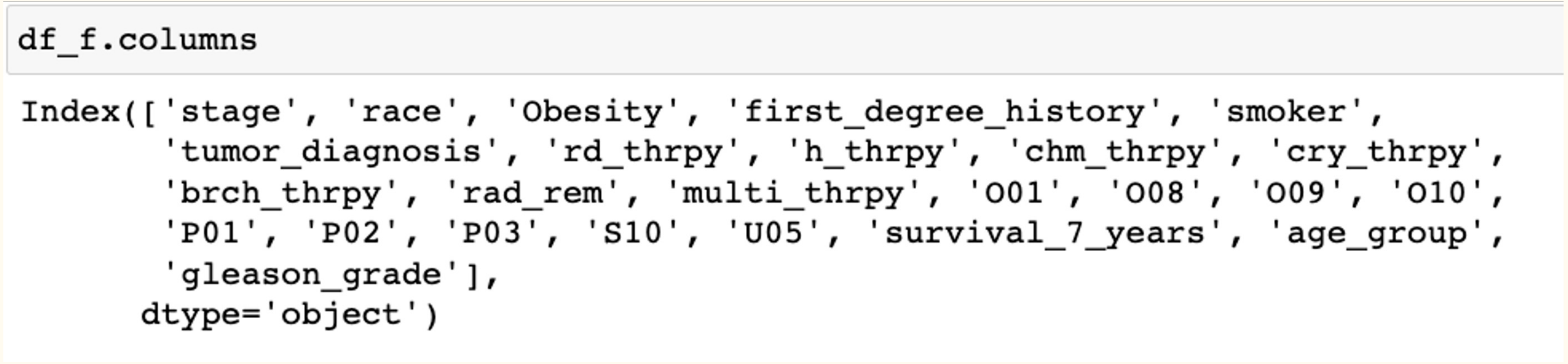
```python
df6.loc[(df6['stage'] == 'I'), 'stage'] = '1'
df6.loc[(df6['stage'] == 'IIA'), 'stage'] = '2'
df6.loc[(df6['stage'] == 'IIB'), 'stage'] = '2'
df6.loc[(df6['stage'] == 'III'), 'stage'] = '3'
df6.loc[(df6['stage'] == 'IV'), 'stage'] = '4'


df6.head()
```

	stage	side	rd_thrpy	h_thrpy	chm_thrpy	cry_thrpy	brch_thrpy	rad_rem	multi_thrpy	symptoms	survival_7_years
0	1	both	0	0	1	1	0	1	1	U03	0.0
1	4	both	1	1	1	0	0	0	1	U06,S07	0.0
2	2	right	1	1	0	0	1	1	1	U01,U02,U03,S10	1.0
3	2	right	0	0	0	1	0	1	1	U01,U02,S10,O11	0.0
4	4	left	1	1	1	0	0	0	1	U01,U03,U05,S07	0.0

Final features

- For building the model, we used 24 features listed below:



```
df_f.columns
```

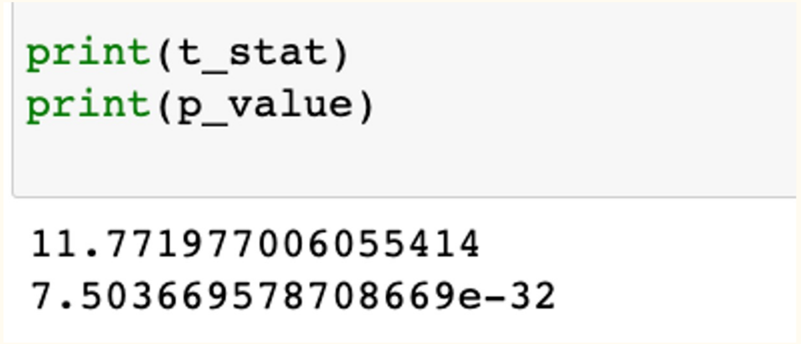
```
Index(['stage', 'race', 'Obesity', 'first_degree_history', 'smoker',
       'tumor_diagnosis', 'rd_thrpy', 'h_thrpy', 'chm_thrpy', 'cry_thrpy',
       'brch_thrpy', 'rad_rem', 'multi_thrpy', 'O01', 'O08', 'O09', 'O10',
       'P01', 'P02', 'P03', 'S10', 'U05', 'survival_7_years', 'age_group',
       'gleason_grade'],
      dtype='object')
```

Bivariate Analysis - T-test

- Applied T-test on tumor_diagnosis variable and target.

- The p_value of the test is smaller than 0.05

- The tumor_diagnosis is significant for predicting target variable.

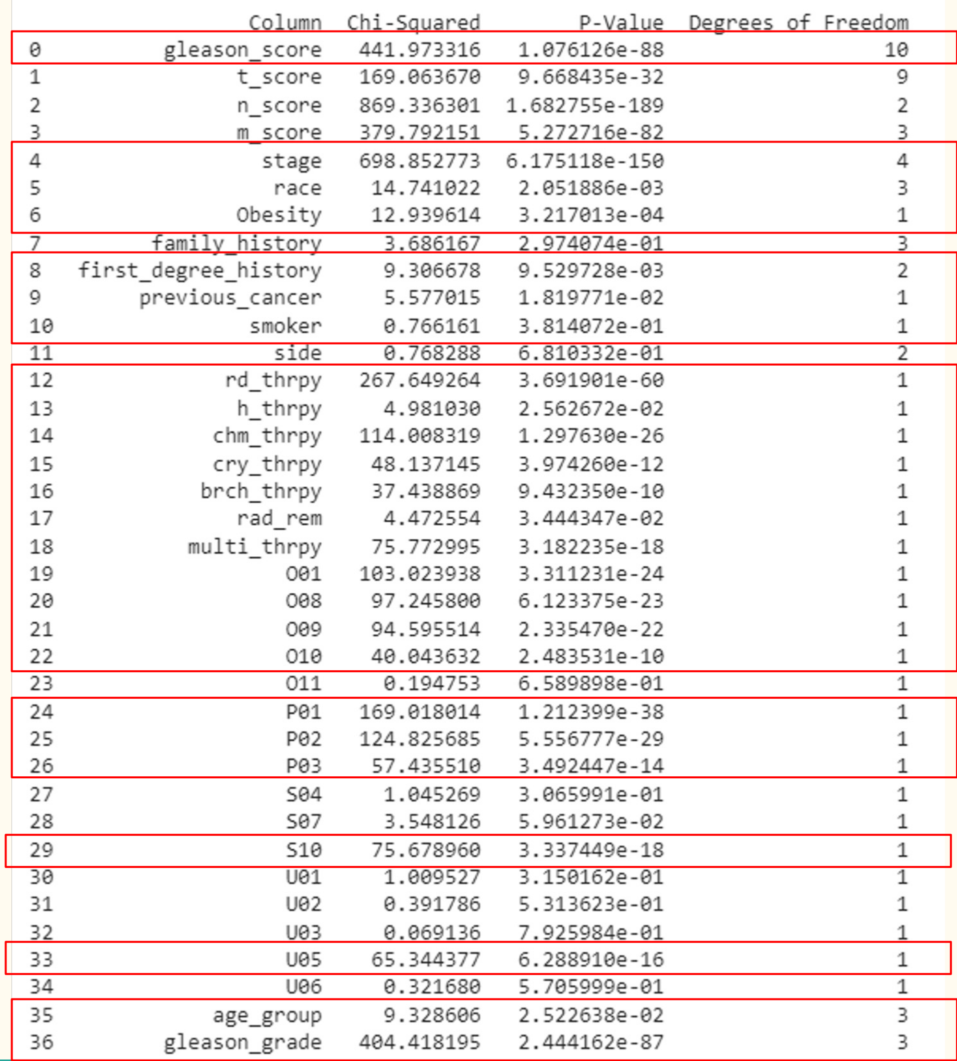


```
print(t_stat)
print(p_value)
```

```
11.771977006055414
7.503669578708669e-32
```

Chi-squared & P value

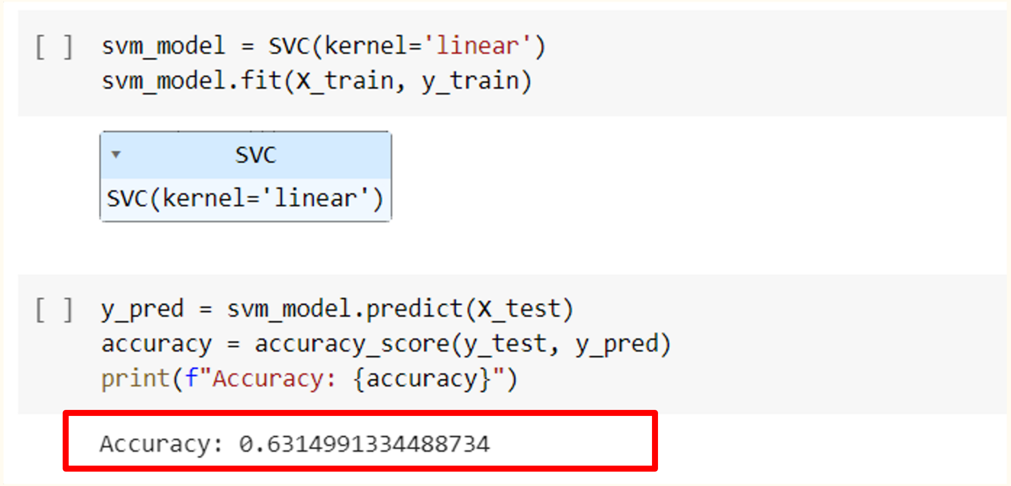
- Performed Chi-Squared test between categorical variables and target variable.
- Dropping the the symptoms which are insignificant
- Dropping Family history as it turned out to be insignificant and similar information is present in first degree history
- Dropping t_score, m_score and n_score as stage would have similar information.
- Dropping side as it is insignificant
- Keeping smoker in as it might affect target
- Dropping gleason score as we have categorised it into gleason grades
- Replaced all the null values with ‘-1’, i.e. classifying as a separate class.



```
                Column  Chi-Squared        P-Value  Degrees of Freedom
0        gleason_score   441.973316   1.076126e-88                  10
1              t_score   169.063670   9.668435e-32                   9
2              n_score   869.336301  1.682755e-189                   2
3              m_score   379.792151   5.272716e-82                   3
4                stage   698.852773  6.175118e-150                   4
5                 race    14.741022   2.051886e-03                   3
6              Obesity    12.939614   3.217013e-04                   1
7       family_history     3.686167   2.974074e-01                   3
8  first_degree_history     9.306678   9.529728e-03                   2
9      previous_cancer     5.577015   1.819771e-02                   1
10              smoker     0.766161   3.814072e-01                   1
11                side     0.768288   6.810332e-01                   2
12            rd_thrpy   267.649264   3.691901e-60                   1
13             h_thrpy     4.981030   2.562672e-02                   1
14           chm_thrpy   114.008319   1.297630e-26                   1
15           cry_thrpy    48.137145   3.974260e-12                   1
16          brch_thrpy    37.438869   9.432350e-10                   1
17             rad_rem     4.472554   3.444347e-02                   1
18         multi_thrpy    75.772995   3.182235e-18                   1
19                 O01   103.023938   3.311231e-24                   1
20                 O08    97.245800   6.123375e-23                   1
21                 O09    94.595514   2.335470e-22                   1
22                 O10    40.043632   2.483531e-10                   1
23                 O11     0.194753   6.589898e-01                   1
24                 P01   169.018014   1.212399e-38                   1
25                 P02   124.825685   5.556777e-29                   1
26                 P03    57.435510   3.492447e-14                   1
27                 S04     1.045269   3.065991e-01                   1
28                 S07     3.548126   5.961273e-02                   1
29                 S10    75.678960   3.337449e-18                   1
30                 U01     1.009527   3.150162e-01                   1
31                 U02     0.391786   5.313623e-01                   1
32                 U03     0.069136   7.925984e-01                   1
33                 U05    65.344377   6.288910e-16                   1
34                 U06     0.321680   5.705999e-01                   1
35           age_group     9.328606   2.522638e-02                   3
36       gleason_grade   404.418195   2.444162e-87                   3
```

Models and Accuracy

- Built a Support Vector Machine model on the train data and predicted on validation data
- Accuracy on validation data os 63.15%



```
[ ] svm_model = SVC(kernel='linear')
    svm_model.fit(X_train, y_train)
```

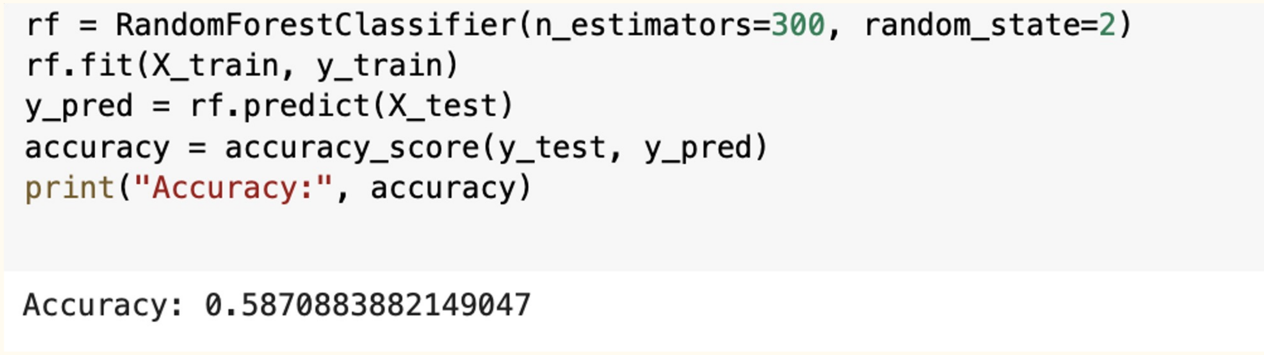
```
▾         SVC
SVC(kernel='linear')
```

```
[ ] y_pred = svm_model.predict(X_test)
    accuracy = accuracy_score(y_test, y_pred)
    print(f"Accuracy: {accuracy}")
```

```
Accuracy: 0.6314991334488734
```

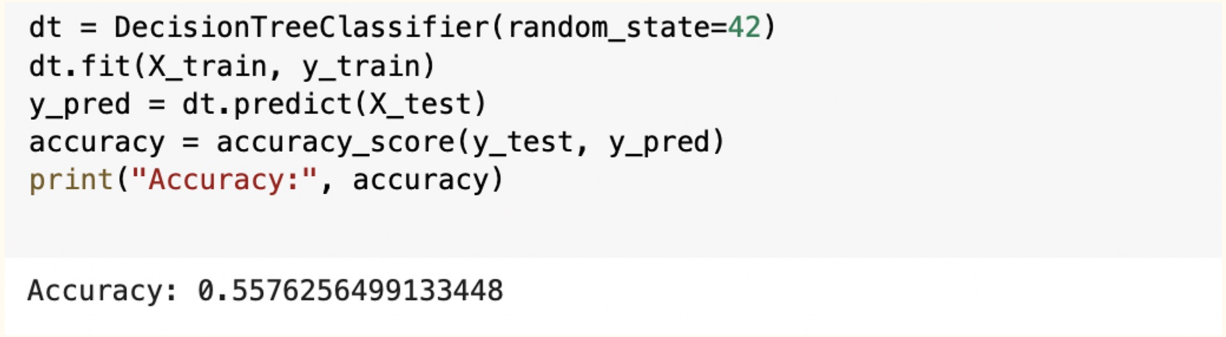
Alternate models:

We used Random forest classifier and Decision classifier we received less accuracy compared to SVM



```
rf = RandomForestClassifier(n_estimators=300, random_state=2)
rf.fit(X_train, y_train)
y_pred = rf.predict(X_test)
accuracy = accuracy_score(y_test, y_pred)
print("Accuracy:", accuracy)
```

```
Accuracy: 0.5870883882149047
```

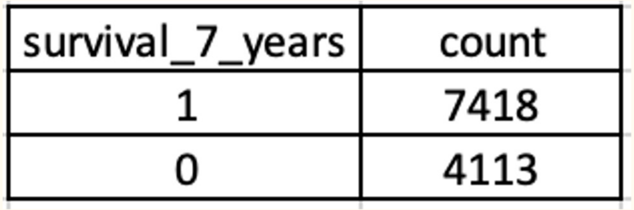


```
dt = DecisionTreeClassifier(random_state=42)
dt.fit(X_train, y_train)
y_pred = dt.predict(X_test)
accuracy = accuracy_score(y_test, y_pred)
print("Accuracy:", accuracy)
```

```
Accuracy: 0.557625649913448
```

Test Data Prediction

- Finally, we used the SVM to predict on test data.
- Generated a csv file with the predictions and concatenated with score.csv
- On the test data:



survival_7_years	count
1	7418
0	4113

Thank you