Класификација на лекови (Drug Classification)

Магдалена Јакимовска 92/2020, Марија Јовановиќ 93/2020, Стефани Павлеска 105/2020, проф. Д-р Христијан Ѓорески

Факултет за електротехника и информациски системи,
Универзитет “Св Кирил и Методиј” во Скопје

Апстракт- Традиционалниот процес на класификација на лекови често се прави преку рачна анализа и стручно знаење, што може да одземе време и да подлежи на човечки предрасуди. Од друга страна пак, алгоритмите за машинско учење можат да научат шеми и врски директно од големи збирки на податоци, овозможувајќи автоматска и објективна класификација на лекови. Овие алгоритми можат да анализираат различни карактеристики поврзани со лекови за да ги класифицираат лековите во значајни категории. Машинското учење има потенцијал да го подобри откривањето на лекови, да ги оптимизира стратегиите за лекување и на крајот да ја подобри грижата за пациентот и резултатите. Во овој труд искористени се пет различни класификатори (Logistic Regression, K Neighbours, Random Forest, Decision Tree, Support Vector Machine) за предвидување на соодветни лекови за идни пациенти. Резултатите покажаа дека Random Forest и Decision Tree се најдобри класификатори со точност од 100%.

Клучни зборови- машинско учење, класификација на лекови, класификатори

I. Вовед

Во последните неколку години, машинското учење претставува моќна алатка во различни области, а полето на класификација на лекови не е исклучок. Класификацијата на лекови со помош на техники за машинско учење има потенцијал да го подобри начинот на кој лековите се идентификуваат, карактеризираат и категоризираат. Со искористување на способностите на вештачката интелигенција и анализа на податоци, алгоритмите за машинско учење можат да помогнат во ефикасна и точна класификација на лековите, помагајќи им на здравствените работници во процесот на донесување на одлука.

Датасетот за класификација на лекови, кој во овој труд се обработува е јавно достапен [2][3][4]. Тој содржи собрани податоци од група пациенти, од кои сите страдале од истата болест. Во текот на индивидуалниот третман, секој пациент реагирал на еден од петте лекови, Лек А, Лек Б, Лек в, Лек х и у.

За секој пациент се складира возраст, пол, крвен притисок (BP), ниво на холестерол, однос на натриум и калиум (Na_to_K) и видот на лекот на кој реагирал пациентот.

Најголемиот предизвик во овој труд беше да се избере класификатор со најголема точност при класифицирањето на лековите за идни пациенти. Ова е од исклучителна важност за изведување на правилен и успешен третман.

II. Карактеристики за датасетот

На Слика 2.1 можеме да ги видиме првите пет примероци од датасетот. Во однос на типот на променливите може да забележиме дека два features се нумерички, а тоа се Возраст и Na_to_K.

	Age	Sex	BP	Cholesterol	Na_to_K	Drug
0	23	F	HIGH	HIGH	25.355	DrugY
1	47	M	LOW	HIGH	13.093	drugC
2	47	M	LOW	HIGH	10.114	drugC
3	28	F	NORMAL	HIGH	7.798	drugX
4	61	F	LOW	HIGH	18.043	DrugY

Слика 2.1 Табела на првите пет примероци од датасетот

Во однос на лековите, на Слика 2.2 може да видиме за секој лек посебно колку примероци се јавуваат во датасетот, односно колку пациенти реагирале на секој лек посебно.

```
DrugY    91
drugX    54
drugA    23
drugC    16
drugB    16
Name: Drug, dtype: int64
```

Слика 2.2 Преглед на бројот на примероци во датасетот за секој лек

Од горенаведените резултати, можеме да заклучиме дека повеќето од пациентите (91 пациент) реагирале на лекот DrugY. Додека пак најмал број од пациентите имале реакција на лекот drugC (16 пациенти) и лекот drugB (16 пациенти).

Според резултатот добиен на Слика 2.3 може да забележиме колку примероци се јавуваат за секој пол во датасетот, каде за машки имаме 104 примероци, додека пак за женски имаме 96 примероци.

```
M    104
F     96
Name: Sex, dtype: int64
```

Слика 2.3 Преглед на бројот на примероци во датасетот за секој пол посебно

Следното истражување кое го направивме беше да добиеме колку примероци се јавуваат за секое ниво на крвен притисок (високо (high), нормално (normal) и ниско (low)). Според резултатот добиен на Слика 2.4, може да заклучиме дека распределбата на нивото на крвниот притисок е избалансирана.

```
HIGH      77
LOW       64
NORMAL    59
Name: BP, dtype: int64
```

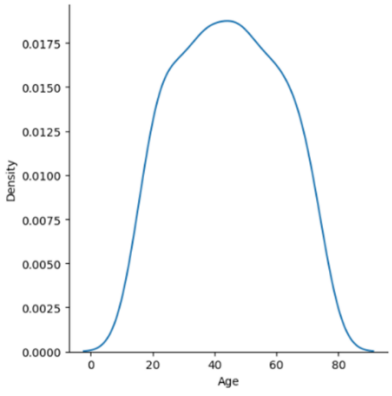
Слика 2.4 Преглед на бројот на примероци во датасетот за секое ниво на крвен притисок

Понатаму го истражувавме нивото на холестерол кај пациентите. Според Слика 2.5 може да воочиме дека 103 пациенти имаат висок холестерол додека пак 97 имаат нормален.

```
HIGH      103
NORMAL     97
Name: Cholesterol, dtype: int64
```

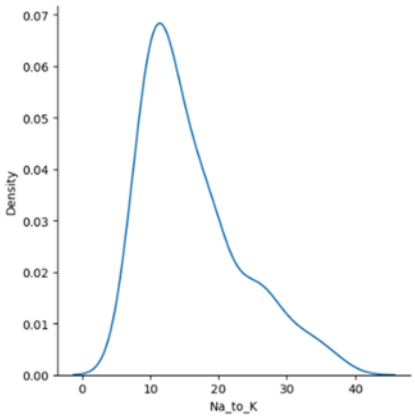
Слика 2.5 Преглед на нивото на холестерол

Следно од голема важност ни беше да ја откриеме возраста на пациентите во датасетот. Возраста е важен фактор при класификација на лековите, каде добивме дека најмладиот пациент има 15 години, а највозрасниот пациент има 74 години.
Користејќи ги библиотеките за визуелизација на податоци во Python, на Слика 2.6 е прикажана дистрибуцијата на возраста на соодветен график.



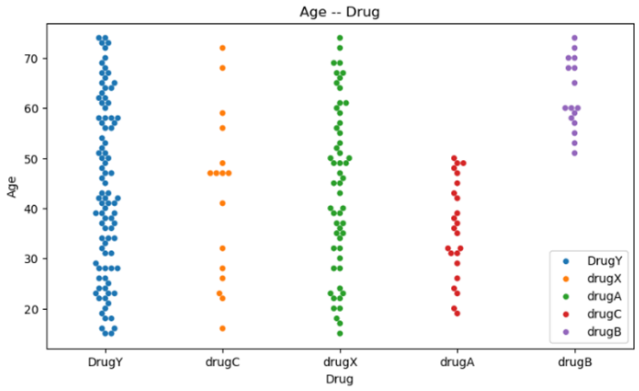
Слика 2.6 Дистрибуција на возраст

На истиот начин на Слика 2.7 се претставува и дистрибуцијата на односот натриум - калиум во крвта. Овој начин на претставување на податоците е за подобра и полесна визуелизација.



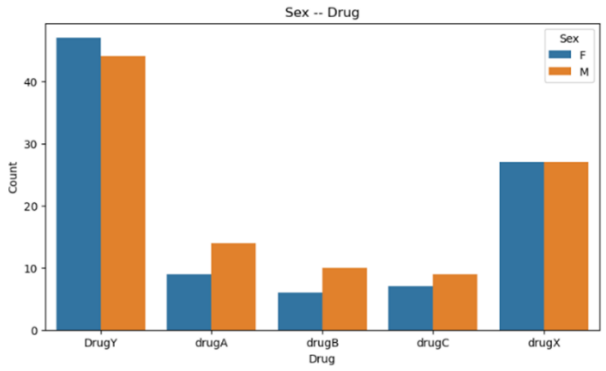
Слика 2.7 Дистрибуција на односот на натриум - калиум во крвта

На следниот график на Слика 2.8, колоната за „лекови'' е прикажана на x-оската, додека пак колоната „возраст'' на y-оската. Колоната во која се распределени лековите исто така се користи како параметар за нијанси, се доделуваат различни бои на точките врз основа на категоријата „лекови'' на која припаѓаат. Целта е да се направи визуелизација на влијанието на различните лекови кај возраста.



Слика 2.8 Анализа Возраст-Лек

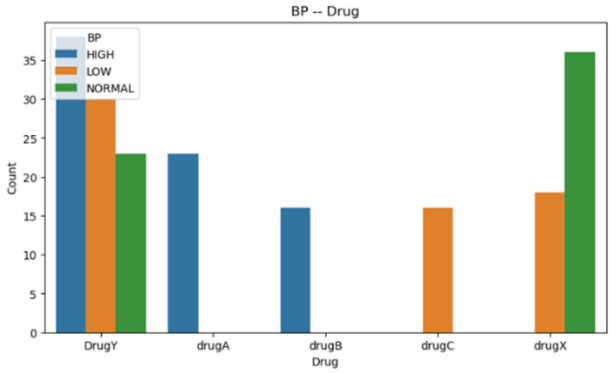
Воочуваме дека на лекот B реагирале само пациенти постари од 51 година, наспроти лекот A на кој реагирале пациенти помлади од 50 години.
Во класификацијата на лековите од особена важност ни е и реакцијата кај мажи и жени соодветно. Тие се разликуваат во нивните реакции на лековите, бидејќи лекот може да резултира со несакани ефекти. Врз основа на истражувања на Системот за известување за несакани настани (AERS) и Администрацијата за храна и лекови на САД [1] се сугерира дека жените доживуваат повеќе несакани ефекти од мажите и дека тие се посериозни кај жените.
Нашето истражување, за соодветната класификација на лекови ни го прикажува следното:



Слика 2.9 Анализа Пол-Лек

Со помош на Слика 2.9, можеме да потврдиме дека на лекот Y реагирале повеќе жени од мажи, реакцијата кај лекот X е еднаква, наспроти лековите А, Б, Ц, на кои реагирале повеќе мажите.

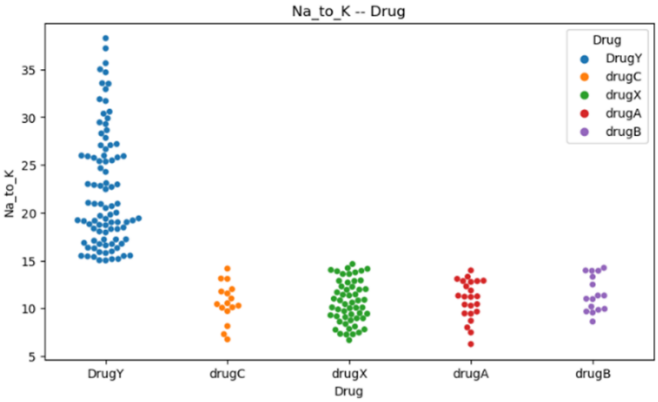
Соодветно, резултатите од класификацијата на Слика 2.10 според нивото на крвен притисок (високо (high), нормално (normal) и ниско (low)) покажуваат дека лековите А и Б пројавиле реакција кај пациенти исклучиво со високо ниво на крвен притисок, лекот Ц кај пациенти со низок крвен притисок, додека пак лекот X пројавил реакција повеќе кај луѓе кои што имаат нормално ниво на крвен притисок.



Слика 2.10 Анализа Крвен притисок (BP)-Лек

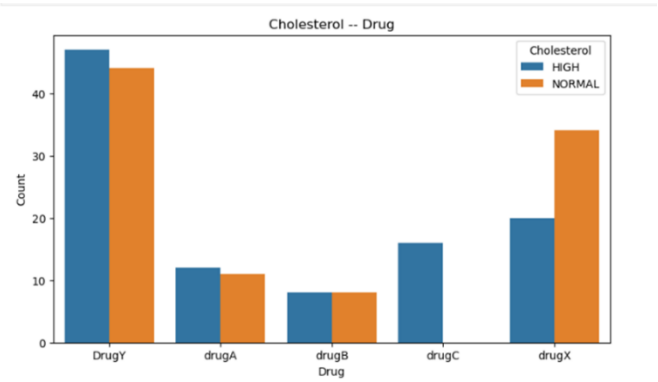
Поделбата на пациентите според крвниот притисок е важна од неколку причини:
Медицинска важност: Крвниот притисок е важно физиолошко мерење кое може да влијае на изборот на лекови. Оваа поделба е релевантна за полесно разбирање како се препишуваат или препорачуваат различни лекови.
Ефикасност на третманот: Одредени лекови може да бидат поефикасни за пациенти со специфични состојби на крвен притисок.
Несакани ефекти или контраиндикации: Крвниот притисок може да биде еден од потенцијалните несакани ефекти кај одредени лекови.
Всушност, класификацијата на лековите врз основа на поделбата на пациентите според крвен притисок може да обезбеди увид во потенцијалните ризици или мерки на претпазливост поврзани со различни лекови за пациенти со различни нивоа на крвен притисок.

Следниот график на Слика 2.11 ни покажува дека пациентите кои имаат сооднос Na_to_K поголем од 15, реагирале на лекот Y.



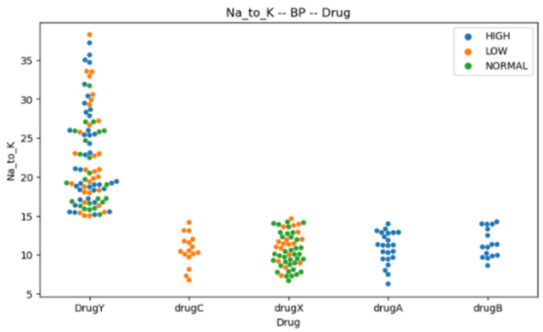
Слика 2.11 Анализа Na:K-Лек

Врската холестерол-лек може да е важна од неколку аспекти: ефективност на лекот, упатства за третман, несакани ефекти и контраиндикации, како и долгорочни исходи.



Слика 2.12 Анализа Холестерол-Лек

Од Слика 2.12 се забележува дека на лекот Ц реагирале пациенти кои имаат високо ниво на холестерол.
Последниот график на Слика 2.13 ни покажува дека луѓето со висок крвен притисок и Na:K помал од 15, реагирале на лековите А и Б, наспроти луѓето со низок крвен притисок и Na:K помал од 15, реагирале само на лекот Ц.



Слика 2.13 Анализа Na:K-BP-Лек

III. КРЕИРАЊЕ НА МОДЕЛИ

Во овој дел од научниот труд, најпрво ќе го подготвиме датасетот пред да ги креираме и тренираме моделите.

Поделба на датасетот (Splitting the Dataset)

Датасетот го делиме на тој начин што 70% од вкупните инстанци ќе ни бидат за тренирање, додека останатите 30% ќе бидат за тестирање. Инстанците се направени случајно да се избираат (shuffle = True), наместо секвенцијално.

```
X = df.drop(["Drug"], axis=1)
y = df["Drug"]
```

Во X променливата го сместуваме целиот датасет со исклучок на колоната Drug. Колоната Drug мора да ја изоставиме бидејќи во X ќе чуваме само влезни (input) променливи. Drug е излезна (output) променлива која ќе се чува во y. Drug е истовремено и самата класа. Врз база на влезните (input) променливите треба да добиеме соодветен излез (output) за типот на лекот.

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size =
0.3, random_state = 42, shuffle = True)
```

Користење на pandas.get_dummies функцијата

Оваа функција се користи за манупулација со податоците. Со други зборови, ги претвора категоричните податоци во dummy или индикатор променливи кои може да примаат вредност 0 или 1.

```
X_train = pd.get_dummies(X_train)
X_test = pd.get_dummies(X_test)
```

```
X_test.head()
```

Следно ги креираме класификаторите.

Logistic Regression

```
from sklearn.linear_model import LogisticRegression
LRclassifier = LogisticRegression(solver='liblinear',
max_iter=5000)
LRclassifier.fit(X_train, y_train)
y_pred = LRclassifier.predict(X_test)
print(classification_report(y_test, y_pred))
print(confusion_matrix(y_test, y_pred))
from sklearn.metrics import accuracy_score
LRAcc = accuracy_score(y_pred,y_test)
print('Logistic Regression accuracy is:
{:.2f}%'.format(LRAcc*100))
```

Прво ја вклучуваме библиотеката за имплементирање на Logistic Regression. Следно го дефинираме Logistic Regression класификаторот. Потоа го тренираме моделот користејќи ги инстанците за тренирање. Истренираниот модел го користиме да врши предвидување врз нови или невидени инстанци. На крајот ја пресметуваме точноста на класификаторот.

На Слика 3.1 прикажани се сите поважни параметри добиени од тренирањето и предвидувањето (precision, recall, f1-score, support, accuracy, macro avg и weighted avg).

Precision: Процент на точни позитивни предвидувања во однос на вкупните позитивни предвидувања.

Recall: Процент на точни позитивни предвидувања во однос на вкупните реални позитивни.

F1 Score: Тежинска средна вредност за precision и recall. Колку е поблиску до 1, толку е подобар моделот.

Support: Овие вредности едноставно ни кажуваат колку пациенти припаѓаат на секоја класа во тест датасетот.

Accuracy: Процентот на правилно класифицирани податоци, кој се движи од 0 до 1.

Macro average: Ја претставува аритметичката средина помеѓу сите f1_scores, така што секој score има иста важност.

Micro (weighted) average: Зема предвид колку примероци има по категорија (колку е поголем support, толку е поважен f1_score на таа категорија).

Исто така прикажана е и матрицата на конфузност.

Овој класификатор ни дава точност од 93.33%.

```
              precision    recall  f1-score   support

       DrugY       0.96      0.96      0.96        26
       drugA       1.00      0.71      0.83         7
       drugB       0.50      1.00      0.67         3
       drugC       1.00      0.83      0.91         6
       drugX       1.00      1.00      1.00        18

    accuracy                           0.93        60
   macro avg       0.89      0.90      0.87        60
weighted avg       0.96      0.93      0.94        60

[[25  0  1  0  0]
 [ 0  5  2  0  0]
 [ 0  0  3  0  0]
 [ 1  0  0  5  0]
 [ 0  0  0  0 18]]
Logistic Regression accuracy is: 93.33%
```

Слика 3.1 Параметри добиени од тренирање и предикција и приказ на матрицата на конфузност кај Logistic Regression

K Neighbours

```
from sklearn.neighbors import KNeighborsClassifier
KNclassifier = KNeighborsClassifier(n_neighbors=20)
KNclassifier.fit(X_train, y_train)
y_pred = KNclassifier.predict(X_test)
print(classification_report(y_test, y_pred))
print(confusion_matrix(y_test, y_pred))
from sklearn.metrics import accuracy_score
KNAcc = accuracy_score(y_pred,y_test)
print('K Neighbours accuracy is: {:.2f}%'.format(KNAcc*100))
```

Прво ја вклучуваме библиотеката за имплементирање на K Neighbours. Следно го дефинираме K Neighbours класификаторот. Го тренираме моделот користејќи ги инстанците за тренирање. Истренираниот модел го користиме

да врши предвидување врз нови или невидени инстанци. На крајот ја пресметуваме точноста на класификаторот.
На Слика 3.2 ги печатиме сите поважни параметри добиени од тренирањето и предикцијата (precision, recall, f1-score, support, accuracy, macro avg и weighted avg).
Исто така прикажана е и матрицата на конфузност.
Овој класификатор ни дава точност од 68.33%.

```
              precision    recall  f1-score   support

       DrugY       0.90      1.00      0.95        26
       drugA       0.33      0.14      0.20         7
       drugB       0.00      0.00      0.00         3
       drugC       0.00      0.00      0.00         6
       drugX       0.54      0.78      0.64        18

    accuracy                           0.68        60
   macro avg       0.35      0.38      0.36        60
weighted avg       0.59      0.68      0.62        60

[[26  0  0  0  0]
 [ 1  1  0  0  5]
 [ 0  0  0  0  3]
 [ 1  1  0  0  4]
 [ 1  1  2  0 14]]
K Neighbours accuracy is: 68.33%
```

Слика 3.2 Параметри добиени од тренирање и предикција и приказ на матрицата на конфузност кај K Neighbours

Random Forest

```
from sklearn.ensemble import RandomForestClassifier
rfc = RandomForestClassifier(random_state = 42)
rfc.fit(X_train,y_train)
y_pred = rfc.predict(X_test)
print(classification_report(y_test, y_pred))
print(confusion_matrix(y_test, y_pred))
from sklearn.metrics import accuracy_score
rfAcc = accuracy_score(y_pred,y_test)
print('Random Forest accuracy is: {:.2f}%'.format(rfAcc*100))
```

Прво ја вклучуваме библиотеката за имплементирање на Random Forest. Следно го дефинираме Random Forest класификаторот. Го трениране моделот користејќи ги инстанците за тренирање. Истрениранот модел го користиме да врши предвидување врз нови или невидени инстанци. На крајот ја пресметуваме точноста на класификаторот.
На Слика 3.3 ги прикажуваме сите поважни параметри добиени од тренирањето и предикцијата (precision, recall, f1-score, support, accuracy, macro avg и weighted avg).
Исто така прикажана е и матрицата на конфузност.
Овој класификатор ни дава точност од 100%.

```
              precision    recall  f1-score   support

       DrugY       1.00      1.00      1.00        26
       drugA       1.00      1.00      1.00         7
       drugB       1.00      1.00      1.00         3
       drugC       1.00      1.00      1.00         6
       drugX       1.00      1.00      1.00        18

    accuracy                           1.00        60
   macro avg       1.00      1.00      1.00        60
weighted avg       1.00      1.00      1.00        60

[[26  0  0  0  0]
 [ 0  7  0  0  0]
 [ 0  0  3  0  0]
 [ 0  0  0  6  0]
 [ 0  0  0  0 18]]
Random Forest accuracy is: 100.00%
```

Слика 3.3 Параметри добиени од тренирање и предикција и приказ на матрицата на конфузност кај Random Forest

Decision Tree

```
from sklearn.tree import DecisionTreeClassifier
DTclassifier = DecisionTreeClassifier(max_leaf_nodes=10
DTclassifier.fit(X_train, y_train)
y_pred = DTclassifier.predict(X_test)
print(classification_report(y_test, y_pred))
print(confusion_matrix(y_test, y_pred))
from sklearn.metrics import accuracy_score
DTAcc = accuracy_score(y_pred,y_test)
print('Decision Tree accuracy is: {:.2f}%'.format(DTAcc*100))
```

Прво ја вклучуваме библиотеката за имплементирање на Decision Tree. Следно го дефинираме Decision Tree класификаторот. Го трениране моделот користејќи ги инстанците за тренирање. Истрениранот модел го користиме да врши предвидување врз нови или невидени инстанци. На крајот ја пресметуваме точноста на класификаторот.
На Слика 3.4 ги печатиме сите поважни параметри добиени од тренирањето и предикцијата (precision, recall, f1-score, support, accuracy, macro avg и weighted avg).
Исто така прикажана е и матрицата на конфузност.
Овој класификатор ни дава точност од 100%.

```
              precision    recall  f1-score   support

       DrugY       1.00      1.00      1.00        26
       drugA       1.00      1.00      1.00         7
       drugB       1.00      1.00      1.00         3
       drugC       1.00      1.00      1.00         6
       drugX       1.00      1.00      1.00        18

    accuracy                           1.00        60
   macro avg       1.00      1.00      1.00        60
weighted avg       1.00      1.00      1.00        60

[[26  0  0  0  0]
 [ 0  7  0  0  0]
 [ 0  0  3  0  0]
 [ 0  0  0  6  0]
 [ 0  0  0  0 18]]
Decision Tree accuracy is: 100.00%
```

Слика 3.4 Параметри добиени од тренирање и предикција и приказ на матрицата на конфузност кај Decision Tree

Support Vector Machine (SVM)

```
from sklearn.svm import SVC
SVCclassifier = SVC(kernel='linear', max_iter=251)
SVCclassifier.fit(X_train, y_train)
y_pred = SVCclassifier.predict(X_test)
print(classification_report(y_test, y_pred))
print(confusion_matrix(y_test, y_pred))
from sklearn.metrics import accuracy_score
SVCAcc = accuracy_score(y_pred,y_test)
print('SVC accuracy is: {:.2f}%'.format(SVCAcc*100))
```

Прво ја вклучуваме библиотеката за имплементирање на Support Vector Machine (SVM). Следно го дефинираме Support Vector Machine (SVM) класификаторот. Го тренираме моделот користејќи ги инстанците за тренирање. Истрениранот модел го користиме да врши предвидување врз нови или невидени инстанци. На крајот ја пресметуваме точноста на класификаторот.
На Слика 3.5 ги печатиме сите поважни параметри добиени од тренирањето и предикцијата (precision, recall, f1-score и support).
Исто така прикажана е и матрицата на конфузност.
Овој класификатор ни дава точност од 98.33%.

```
              precision    recall  f1-score   support

       DrugY       1.00      0.96      0.98        26
       drugA       1.00      1.00      1.00         7
       drugB       0.75      1.00      0.86         3
       drugC       1.00      1.00      1.00         6
       drugX       1.00      1.00      1.00        18

    accuracy                           0.98        60
   macro avg       0.95      0.99      0.97        60
weighted avg       0.99      0.98      0.98        60

[[25  0  1  0  0]
 [ 0  7  0  0  0]
 [ 0  0  3  0  0]
 [ 0  0  0  6  0]
 [ 0  0  0  0 18]]
SVC accuracy is: 98.33%
```

Слика 3.5 Параметри добиени од тренирање и предикција и приказ на матрицата на конфузност кај Support Vector Machine

IV. Заклучок

Во овој труд ги споредивме перформансите на пет различни класификатори (Logistic Regression, K Neighbours, Random Forest, Decision Tree, Support Vector Machine). Од самите резултати можеме да заклучиме дека класификаторот Random Forest и класификаторот Decision Tree достигнуваат 100% точност.
Останатите класификатори Logistic Regression и Support Vector Machine достигнуваат повеќе од 90% точност. Класификаторот со најлоши перформанси се покажа дека е K Neighbors, со точност од само 68,33%.

	Model	Accuracy
2	Random Forest	100.000000
3	Decision Tree	100.000000
4	SVM	98.333333
0	Logistic Regression	93.333333
1	K Neighbours	68.333333

Слика 4.1 Приказ на точноста на секој модел

Со оглед на тоа што овие класификатори беа изградени користејќи податоци од релативно мал број на пациенти, останува отворено прашањето за идни истражувања за тоа дали искористените модели ќе соодветствуваат на различни профили на пациенти.

Литература

[1] Guideline for the Study and Evaluation of Gender Differences in the Clinical Evaluation of Drugs , Food and Drug Administration – Department of Health and Human Services, 22 јули, 1993.

[2] drug-classification-plotly-eda-ml jupyter notebook, Bilal Bora

[3] drug-classification-w-various-ml-models jupyter notebook, Mario Caesar

[4] drug-classification-with-different-algorithms jupyter notebook, Görkem Günay