

---

## PROIECT FUNDAMENTE DE BIG DATA

---

### UTILIZAREA MODELELOR DE PREDICTIE IN PROCESUL DE DIAGNOSTICARE AI CANCERULUI MAMAR MALIGN

STUDENTI:                      CRET                      ALIN  
   CHELARU ROBERT-ADRIAN

AN: IE 3

## CUPRINS

1. INTRODUCERE.....	3
2. SETUL DE DATE.....	4
3. REZULTATE SI DISCUTII.....	6
4. CONCLUZIA.....	15
5. BIBLIOGRAFIE.....	15

# 1.Introducere

Cancerul la san este una dintre cele mai frecvente tipuri de cancer la femei, fiind o problema majora de sanatate publica la nivel global. Potrivit Organizatiei Mondiale a Sanatatii, cancerul la san reprezinta aproximativ 25% din toate cazurile de cancer la femei. Diagnosticarea corecta si timpurie este importanta pentru imbunatatirea prognosticului si a calitatii vietii pacientelor. In acest context, dezvoltarea unor metode precise de clasificare a tumorilor mamare ca fiind benigne sau maligne este importanta.

Scopul acestui proiect este de a dezvolta un model de predictie care sa clasifice tumorile de san ca fiind fie benigne sau maligne, pe baza caracteristicilor lor fizice. Utilizarea modelelor de predictie antrenate pe date disponibile de la o lista de pacienti ce au trecut deja prin aceasta boala, poate oferi instrumente valoroase pentru specialistii in domeniul medical, ajutandu-i sa ia decizii informate si rapide.

## Intrebari relevante proiectului

1. Care sunt cei mai importanti factori a unei tumori? Ce caracteristici ajung sa determine un cancer malign?
2. Poate un model de predictie sa fie folosit in procesul de diagnosticare? De ce procentaj de acuratete am avea nevoie pentru a putea justifica utilizarea acestuia?
3. Ce alte caracteristici, inexistente in fisierul de date utilizat - ale tumorii - ar putea fi analizate pentru realizarea unui model de predictie mai bun?

## 2.Setul de date

Setul de date utilizat in acest proiect provine de pe platforma Kaggle si include urmatoarele caracteristici:

- **Diagnosis:** Diagnosticul tumorii (0 = benign, 1 = malign).

**Caracteristici fizice:** Include parametri cum ar fi:

- **Raza medie:** Diametrul mediu al celulelor tumorale.
- **Textura medie:** Distributia intensitatii culorii in imagine.
- **Perimetrul mediu:** Perimetrul mediu al celulelor tumorale.
- **Aria medie:** Suprafata medie a celulelor tumorale.
- **Netezimea medie:** Masura netezimii marginilor celulelor.

Informatiile cuprinse in acest fisier au fost adunate de la 570 de pacienti. Setul de date contine 6 atribute ( coloane ) din care 5 sunt caracteristici folosite pentru a prezice al 6-lea eveniment si anume daca tumoarea este maligna sau benigna.

Pentru a incepe analiza datelor am incarcat csv-ul in fisierul python ce il vom folosi pentru efectuarea tuturor proceselor de dezvoltare a modelului de predictie. Instantele si atributele au fost citite cu functia:

```
data = pd.read_csv('Breast_cancer_data.csv').
```

Obtinerea unei priviri de ansamblu asupra datelor o obtinem folosind

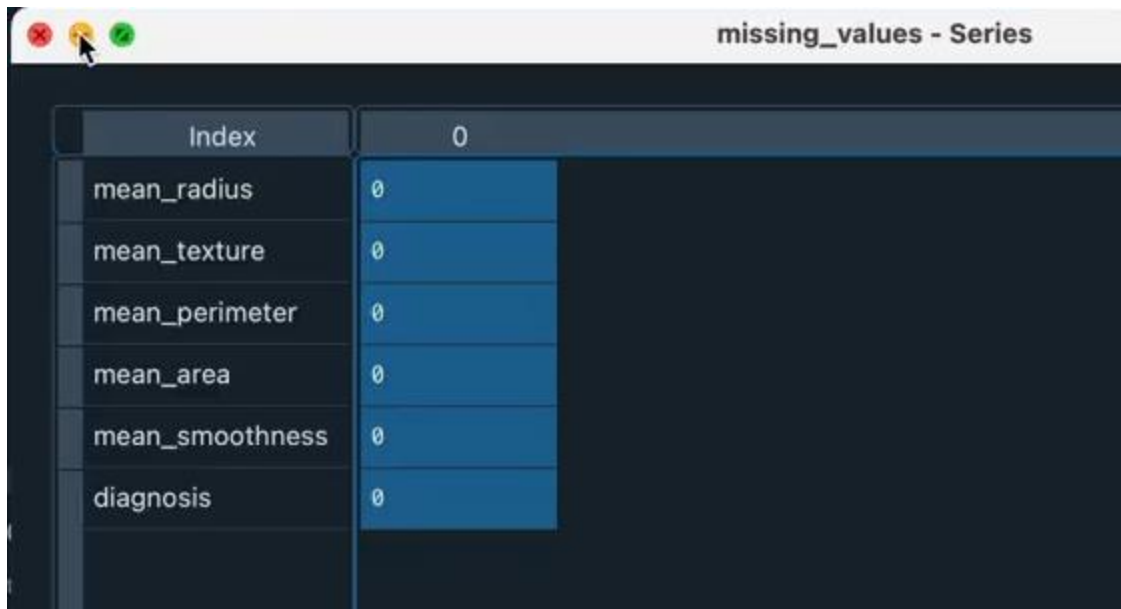
```
descriptive_stats = data.describe()
```

	Index	mean_radius	mean_texture	mean_perimete	mean_area	mean_smoothne	diagnosis
	count	569	569	569	569	569	569
	mean	14.1273	19.2896	91.969	654.889	0.0963603	0.627417
	std	3.52405	4.30104	24.299	351.914	0.0140641	0.483918
	min	6.981	9.71	43.79	143.5	0.05263	0
	25%	11.7	16.17	75.17	420.3	0.08637	0
	50%	13.37	18.84	86.24	551.1	0.09587	1
	75%	15.78	21.8	104.1	782.7	0.1053	1
	max	28.11	39.28	188.5	2501	0.1634	1

Figura 1. Rezultat in urma rularii data.describe()

Pentru a ne asigura ca fisierul utilizat nu contine date goale, efectuam o verificare folosind

```
missing_values = isnull().sum().
```



Index	0
mean_radius	0
mean_texture	0
mean_perimeter	0
mean_area	0
mean_smoothness	0
diagnosis	0

Figura 2. Rezultat in urma cautarii de date goale (null)

In continuare putem observa ca varianta independenta Diagnosis se incadreaza in tipul de date boolean/binar (0 - benigna, 1 - maligna).

```
data.hist(bins=15, figsize=(15, 10), layout=(3, 3))
```

*Mentiune – am ales layout 3-3 din cauza ca aveam erori de afisare cu 2-3.*

Aceasta observatie este sustinuta de urmatoarea figura:

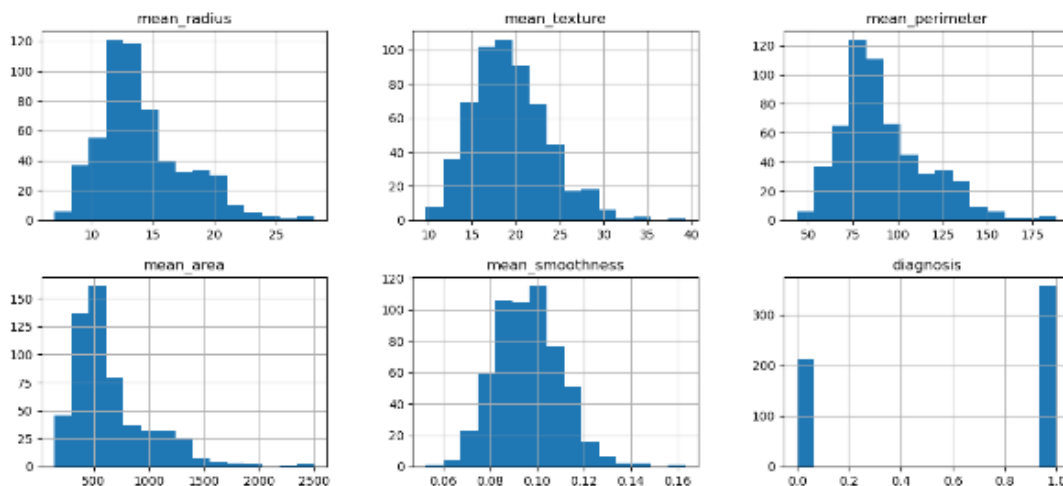


Figura 3. Distributia atributelor numerice

Pentru a putea merge mai departe cu analiza am decis ca atributul Diagnosis sa fie luat ca atribut tinta pentru cercetarea noastra. Urmatoarea linie de cod va produce ca output o distributie a valorilor stocate in atributul Diagnosis.

```
target_distribution = data['diagnosis'].value_counts(normalize=True)
```



diagnosis	proportion
1	0.627417
0	0.372583

Figura 4. Distributia valorilor din atributul tinta Diagnosis

Modelul de predictie va primi datele impartite. Noi am ales le impartim intr-o proportie de 80-20. 80% din datele noastre vor fi puse in setul de antrenament iar restul de 20% ca set de test. Initial, fold-ul nostru va fi constituit de primele 80% - antrenament, ultimele 20% - test. In urma parcurgerii mai multor modele de predictie, vom alege modelul pe care il consideram ca este mai favorabil in cazul nostru, si vom rula apoi modelul in 5 fold-uri - detalierea procesului va fi realizata cand ajungem la acel pas.

## 3.Rezultate si Discutii

### 3.1 Rularea a mai multor modele predictie

Pentru o mai mare precizie in predictia diagnosticului, am ales sa incercam 4 modele de predictie.

#### 3.1.1 Regresia Logistica

Incepem procesul de incercare a modelelor de predictie cu modelul de Regresie Logistica. Instantam obiectul.

LR\_model = 500 reprezinta numarul maxim de iteratii pe care algoritmul sa le execute.

```
lr_model = LogisticRegression(max_iter=500)
```

Antrenam modelul folosind coloanele cu caracteristici – folosite in procesul de diagnosticare – din setul salvat in X\_train, si coloana cu diagnosticul binar reprezentativ malign/benign salvata

in y\_train – aceste coloane constituind primele 80% dintre datele noastre disponibile din fisierul csv.

```
lr_model.fit(X_train, y_train)
```

Modelul antrenat este apoi folosit pentru a prezice diagnosticul pentru restul de 20% date, pastrate ca date de test in variabila X\_test.

```
y_pred_lr = lr_model.predict(X_test)
```

Rezultatele incarcate in y\_pred, sunt comparate cu valorile adevarate de diagnostic. Afisam apoi acurateta modelului.

```
lr_accuracy = accuracy_score(y_test, y_pred_lr)
print(f"Logistic Regression Accuracy: {lr_accuracy:.4f}")
```

Rezultatul afisarii este: **Logistic Regression Accuracy: 0.9298**

### 3.1.2 Support Vector Classifier

Asemenea modelului anterior , incepem Support Vector Classifier cu instantarea obiectului.

```
svc_model = SVC()
```

Antrenam modelul folosind coloanele cu caracteristici – folosite in procesul de diagnosticare – din setul salvat in X\_train, si coloana cu diagnosticul binar reprezentativ malign/benign salvata in y\_train – aceste coloane constituind primele 80% dintre datele noastre disponibile din fisierul csv.

```
svc_model.fit(X_train, y_train)
```

Modelul antrenat este apoi folosit pentru a prezice diagnosticul pentru restul de 20% date , pastrate ca date de test in variabila x\_test

```
y_pred_svc = svc_model.predict(X_test)
```

Rezultatele incarcate in y\_pred, sunt comparate cu valorile adevarate de diagnostic. Afisam apoi acurateta modelului.

```
svc_accuracy = accuracy_score(y_test, y_pred_svc)
print(f"Support Vector Classifier: {svc_accuracy:.4f}")
```

Rezultatul afisarii este: `Support Vector Classifier: 0.9211`

### 3.1.3 Random Forest

Asemenea modelelor anterioare , incepem Random Forest cu instantarea obiectului.

N\_estimators = 500 reprezinta numarul total de arbori decizionali .Fiecare arbore este antrenat pe un subset aleator de date cu un numar random de coloane din setul de date. Predictia finala este obtinuta din calculul mediei predictiilor

```
rf_model = RandomForestClassifier(n_estimators=500)
```

Antrenam modelul folosind coloanele cu caracteristici – folosite in procesul de diagnosticare – din setul salvat in X\_train, si coloana cu diagnosticul binar reprezentativ malign/benign salvata in y\_train – aceste coloane constituind primele 80% dintre datele noastre disponibile din fisierul csv.

```
rf_model.fit(X_train, y_train)
```

Modelul antrenat este apoi folosit pentru a prezice diagnosticul pentru restul de 20% date , pastrate ca date de test in variabila x\_test

```
y_pred_rf = rf_model.predict(X_test)
```

Rezultatele incarcate in y\_pred, sunt comparate cu valorile adevarate de diagnostic. Afisam apoi acurateta modelului.

```
rf_accuracy = accuracy_score(y_test, y_pred_rf)
```



```
print(f"Random Forest Accuracy: {rf_accuracy:.4f}")
```

Rezultatul afisarii este: **Random Forest Accuracy: 0.9474**

### 3.1.4 XGBoost

Asemenea modelelor anterioare , incepem Random Forest cu instantarea obiectului.

Parametrul `use_label_encoder = False` indica programului sa nu foloseasca encoderul intern pe care il pune la dispozitie XGBosst deoarece datele noastre sunt preprocesate.

`Eval_metric = 'logloss'` seteaza ca metrica de evaluare folosita sa fie logarithmic loss. Aceasta masoara acuratetea variabilelor prezise cu valori mai mici indicand o performanta mai buna.

`N_estimators = 500` specifica modelului sa construiasca 500 de arbori secventiali , unde fiecare arbore incearca sa corecteze eroarea arborilor precedenti.

```
xgb_model = XGBClassifier (use_label_encoder=False,  
                           eval_metric='logloss',  
                           n_estimators=500)
```

Antrenam modelul folosind coloanele cu caracteristici – folosite in procesul de diagnosticare – din setul salvat in `X_train`, si coloana cu diagnosticul binar reprezentativ malign/benign salvata in `y_train` – aceste coloane constituind primele 80% dintre datele noastre disponibile din fisierul csv.

```
xgb_model.fit(X_train, y_train)
```

Modelul antrenat este apoi folosit pentru a prezice diagnosticul pentru restul de 20% date , pastrate ca date de test in variabila `x_test`

```
y_pred_xgb = xgb_model.predict(X_test)
```

Rezultatele incarcate in `y_pred`, sunt comparate cu valorile adevarate de diagnostic. Afisam apoi acurateta modelului.

```
xgboost_accuracy = accuracy_score(y_test, y_pred_xgb)
print(f"XGboost Accuracy: {xgboost_accuracy:.4f}")
```

Rezultatul afisarii este: **XGboost Accuracy: 0.9298**

*Random Forest se dovedeste a fi cel mai precis model in cazul nostru. Cu o precizie de 94.74%.*

## 3.2 Cautarea de hyper parametrui pentru Random Forest si antrenarea modelului pe 5 impartiri de date (5-fold)

### 3.2.1 Cautarea de hyper parametrui pentru Random Forest

Pentru a eficientiza procesul de predictie, vom incerca mai multe combinatii de hyper parametrui, hyper parametrui ne ofera posibilitatea personalizarii functionarii modelului Random Forest.

Initializam din nou modelul si alegem valori pentru parametrui.

```
rf_model = RandomForestClassifier()
```

```
param_grid = {
```

```
# numar de copaci/number of trees
```

```
    'n_estimators': [500],
```

```
# adancimea maxima a copacilor, none poate aduce o precizie mai mare dar care sa fie relevanta
DOAR la setul de date analizat in cadrul proiectului (overfitting), in timp ce 5 si 10 sunt valori ce
previn overfitting, ducand la un model mai precis in afara datelor din proiect.
```

```
    'max_depth': [None, 5, 10],
```

```
# impartirea mostrelor, 2 – care este valoarea default - duce la un model mai complex, in
timp ce 5 ne va produce un model mai simplu deoarece imparte mostrele de date in mai putine
seturi
```

```
    'min_samples_split': [2, 5],
```

# la fiecare nod parcurs, setam un numar minim de mostre, 1 – care e valoarea default – ne ajuta sa constituim un model mai detaliat dar cu risc de overfitting. 2 si 4 pot preveni acest caz.

```
'min_samples_leaf': [1, 2, 4],
```

# true este valoarea default, reduce variatia si previne overfittingul, false face ca fiecare copac din model sa fie antrenat pe fiecare set de date, precis dar poate duce la overfitting.

```
'bootstrap': [True, False]
```

```
}
```

Aceste posibilitati de configurare, ne ofera 36 de combinatii – numiti candidati (candidates).

### 3.2.2 Configurarea obiectului de tip grid search, pentru antrenamentul pe date impartite in 5 seturi

Cele 36 de posibilitati de combinatii, urmeaza sa fie antrenate in mai multe impartiri de date. Daca initial modelele noastre erau antrenate pe primele 80% de date din setul nostru, lasand restul de 20% ca set de antrenament, acum vom avea mai multe combinatii separate pe baza de folds. Un fold reprezinta o sectiune a setului nostru de date.

	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5
▶	Test	Train	Train	Train	Train
▶	Train	Test	Train	Train	Train
▶	Train	Train	Test	Train	Train
▶	Train	Train	Train	Test	Train
▶	Train	Train	Train	Train	Test

Figura 5. Vizualizarea fold-urilor

Daca antrenam modelul pe diferite combinatii - un exemplu este primele 40% si ultimele 40% ca set de antrenament si 20% de la mijlocul setului de date ca set de test – obtinem un total de date mai mare, si putem obtine rezultate de precizie a modelului mai bune.

```
grid_search = GridSearchCV(  
  
# Selectam modelul pe care dorim sa il optimizam , in cazul nostru rf_model care este un  
RandomForestClassifier  
  
    estimator=rf_model,  
  
# Indicam obiectului grid_search sa foloseasca parametrii din dictionarul param_grid.  
  
    param_grid=param_grid,  
  
  
    # Metoda de evaluare a performantei modelelor pe care o folosim este acuratetea.  
  
    scoring='accuracy',  
  
    # Aici se defineste cate nuclee ale procesorului se vor folosi. In cazul nostru sunt toate  
nucleele procesorului, pentru a accelera cautarea.  
  
  
    n_jobs=-1,  
  
# Imparte setul de date in 5 folduri pentru validare incrucisata, asigurandu-ne ca fiecare parte a  
setului de date este utilizata pentru ambele procese, de testare si antrenare.  
  
    cv=5,  
  
# Controleaza nivelul de iesire de detalii in timpul rularii, unde 1 indica programului sa  
furnizeze ca output doar solutia optima  
  
    verbose=1  
  
)
```

Avand 5 fold-uri si 36 de combinari de hiper parametrii (36 candidates), obtinem un total de 180 rulari pentru determinarea celor mai bune setari pentru model.

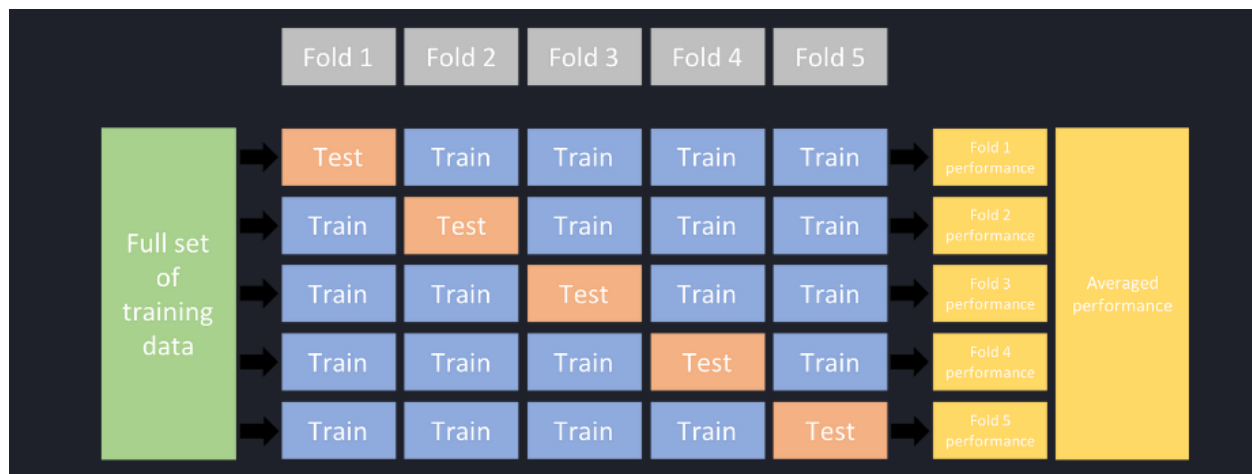


Figura 6. Vizualizarea foldurilor in ansamblu cu restul elementelor

In urma cautarii:

```
grid_search.fit(X_train, y_train)
```

Obinem urmatoarea combinatie de parametrii:

```
Fitting 5 folds for each of 36 candidates, totalling 180 fits
{'bootstrap': True, 'max_depth': None, 'min_samples_leaf': 2, 'min_samples_split': 2, 'n_estimators': 500}
```

### 3.3 Rularea modelului cu cei mai buni parametrii gasiti

Folosind parametrii gasiti in pasul anterior, folosim acum modelul Random Forest pentru a realiza un model de predictie mai precis.

```
rf_model = RandomForestClassifier(bootstrap=True, max_depth=None,
                                  min_samples_leaf=1, min_samples_split=2, n_estimators=500)
```

In urma ultimei rulari pe testul de antrenament obtinem:

```
Accuracy: 0.9561
```

### 3.4 Determinarea celui mai important parametru in procesul de diagnosticare

In procesul de diagnosticare avem disponibili 5 caracteristici insa fiecare are o relevanta diferita, dupa cum putem observa in imaginea de mai jos, mean\_perimeter este factorul care influenteaza cel mai mult procesul.

```
feature_importances = rf_model.feature_importances_  
plt.barh(X.columns, feature_importances)  
plt.xlabel("Feature Importance")
```

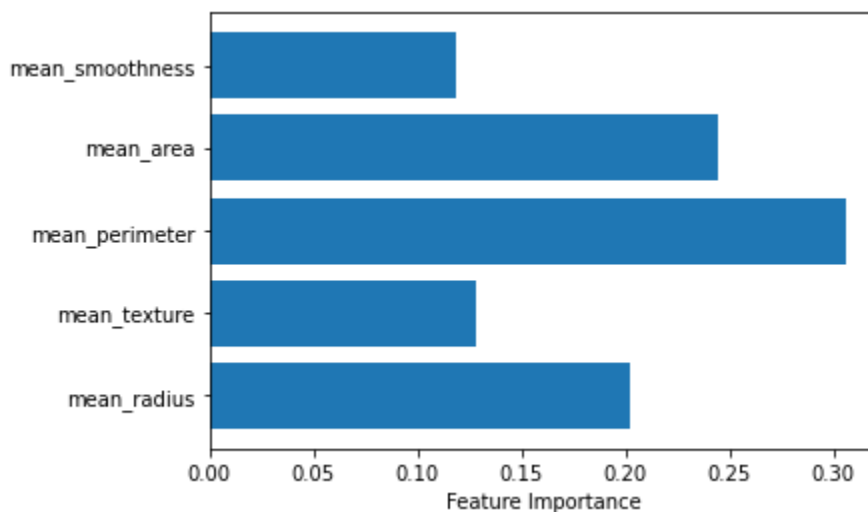


Figura 7. Importanta caracteristicilor tumorei

### 3.5 Discutie asupra intrebarilor de la inceputul proiectului

Intrebarile de la inceputul proiectului aveau in principal ca scop, determinarea utilitatii unui model de predictie in ajutorul diagnosticarii de cancer mamar, dar si aflarea limitarilor setului de date.

Cei mai importanti factori s-au dovedit a fi mean\_perimeter, avand un coeficient de importanta un pic peste 0.30. Caracteristica de mean\_area este a doua cea mai importanta cu un coeficient aproape de 0.25, iar mean\_radius cu 0.20.

Procentajul de precizie necesar pentru ca modelul sa fie utilizat este un lucru subiectiv, corpurile guvernamentale vor fi intotdeauna in favoarea reglementarii admisiei de modele cu o rata de precizie 98-100%. Insa noi consideram ca o rata de precizie de cel putin 95% ar trebui sa fie admisibila, un procent care este realizabil fara a incuraja manipularea datelor de precizie in scopul ca modelul sa fie unul admis pentru utilizare.

Cand vine vorba de utilitatea modelului de predictie, in prezent un doctor va avea intotdeauna ultimul cuvant. Daca un model de predictie are o rata de succes apropiata de 100%, acesta poate fi un adjuvant de valoare ce poate corecta o greseala umana facuta la prima analiza a tumorii. Desigur, simptomatologia unei tumori nu se limiteaza doar la forma acesteia. Ea poate avea diverse alte simptome precum durere persistenta, eruptii cutanate, inrosirea pielii in zonele adiacente si multe altele. Aceste variabile nu sunt luate in calcul de modelul de predictie, din cauza lipsei datelor. De aceea modelul ar trebuii utilizat impreuna cu expertiza unui specialist pentru a maximiza rata de diagnostic corect.

## 4.Concluzia

Documentul prezinta un studiu asupra utilizarii modelelor de predictie in diagnosticarea cancerului mamar malign. Concluzia principala este ca modelul Random Forest s-a dovedit a fi cel mai eficient, obtinand o precizie de 95,61%. Desi modelele de predictie pot imbunatati procesul de diagnosticare, este esential ca acestea sa fie utilizate in completarea expertizei medicilor pentru a asigura cele mai precise si complete diagnostice. Totodata, viitoarele studii si alcatuiri de seturi de date mai complete pot duce la o crestere a preciziei modelelor.

## 5.Bibliografie

<https://archive.ics.uci.edu/dataset/17/breast+cancer+wisconsin+diagnostic>

<https://www.kaggle.com/datasets/merishnasuwal/breast-cancer-prediction-dataset>

[https://docs.deepchecks.com/stable/api/generated/deepchecks.tabular.datasets.classification.breast\\_cancer.html](https://docs.deepchecks.com/stable/api/generated/deepchecks.tabular.datasets.classification.breast_cancer.html)

<https://scikit-learn.org/stable/modules/ensemble.html#forest>

<https://www.kaggle.com/code/alexisbcook/xgboost>

<https://docs.ultralytics.com/guides/kfold-cross-validation/>

<https://www.reginamaria.ro/articole-medicale/investigatiile-recomandate-pentru-depistarea-cancerului-mamar-dr-isabela-botea>

<https://www.medlife.ro/glosar-medical/afectiuni-medicale/cancer-mamar-cauze-simptome-tratament>