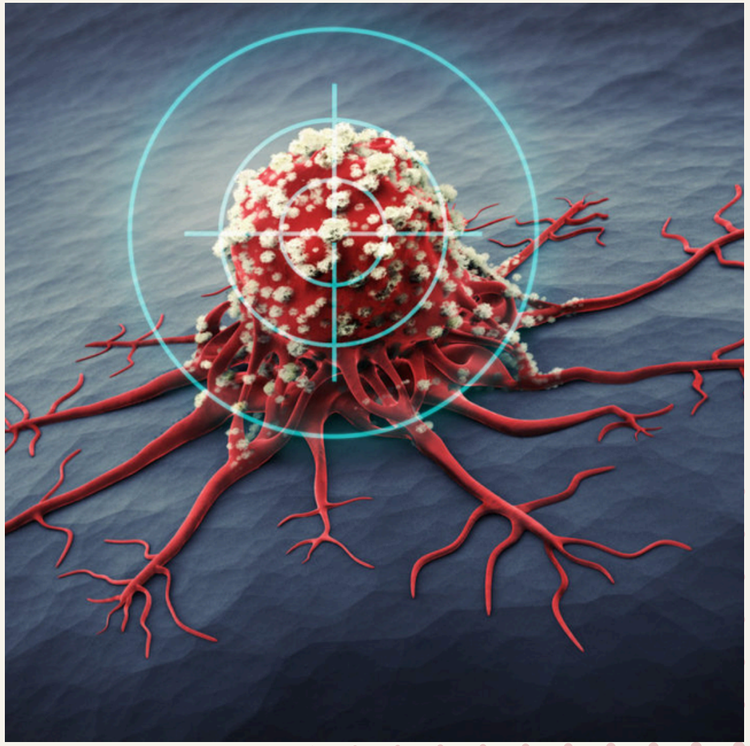
PREDICTING CANCER DIAGNOSIS

Presented By : Lauren Henry, Cathy Matthee, Zahra Razook, Yingyi (Aria) Li & Jiuyuan (Jess) Zhang

Data Analytics Bootcamp | 2024

INTRODUCTION

- Purpose:
  - To assist with breast cancer survial research through early intervention
- Aim:
  - To create an application that utilises machine learning for researchers to input visual charactertistics of a cancer and predict whether it is Benign or Malignant.
- Data:
  - Breast Cancer Wisconsin (Diagnostic) Data Set https://www.kaggle.com/datasets/erdemtaha/cancer-data/data



ETHICAL CONSIDERATIONS

Licensing

This Data has a CC BY-NC-SA 4.0 License which allows for the following:

- Share — copy and redistribute the material in any medium or format
- Adapt — remix, transform, and build upon the material

Ethics

Personally Identifiable Information (PII):

- This dataset includes an ID number that is anonymized to protect the privacy of individuals.

Misuse:

- A disclaimer has been included in the readme and when the diagnosis is presented on the app.
  - Aims to prevent misuse of app and remind users it does not replace a professional medical diagnosis.

DATA FETCHING & INTEGRATION

Database

Postgres pgAdmin 4

Flask

Jupyter NB

4 x machine learning models trained

{.json}

Trained ML models

Front End

Predictor app for cancer

scikit learn

jupyter

TensorFlow

DATA PREPROCESSING

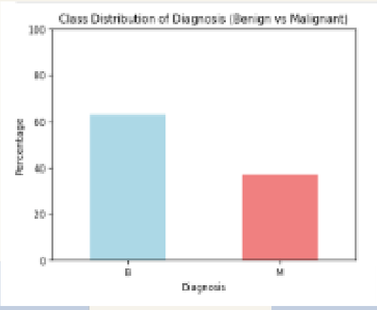
Breast cancer dataset

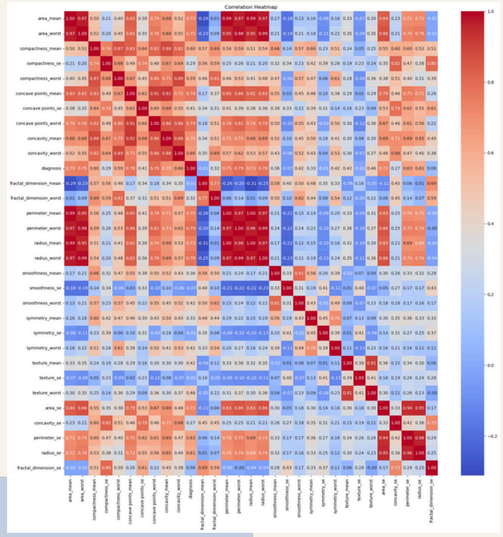
Exploratory Data Analysis

32 columns, 569 rows

target variable:
diagnosis
(malignant or benign)

Class imbalance





some features are highly correlated

Data Cleaning and Preprocessing

Drop 'Unnamed: 32' and 'id' columns
Label Encoding

Outliers capped at the 1st and 99th percentiles.
Treating skewness of features.

Split data
Feature selection
scale data

LOGISTIC REGRESSION

Preprocessed data

initialisation of Model
Logistic Regression

```
LogisticRegression
LogisticRegression(class_weight='balanced', max_iter=200, random_state=1)
```

Correlation-based feature selection

Recursive Feature Elimination (RFE)

Model Training

Model with reduced features:
Test Accuracy after RFE: 0.96

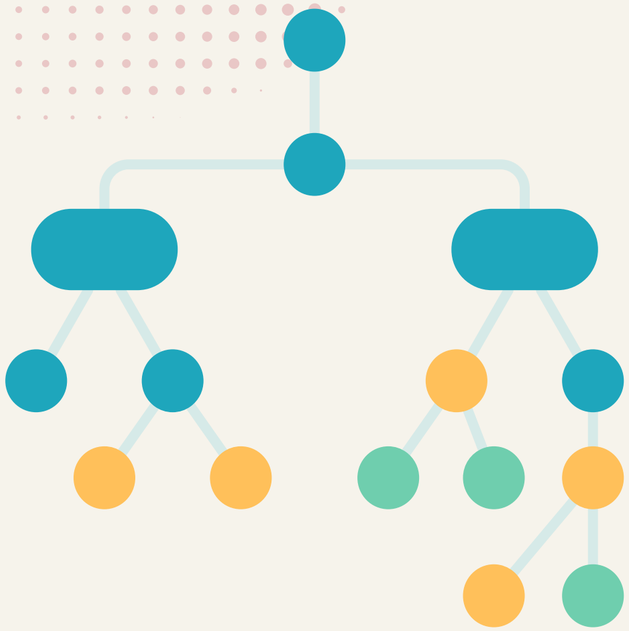
Hyperparameter
Tuning
GridSearchCV

Model with all features:
Test accuracy base model: 0.97
Test accuracy tunned model .98

Model Evaluation

classes	Test Accuracy	Precision	Recall	f1-score	Confusion Matrix	Confusion Matrix	ROC-AUC	Log Loss	Cross-Validation Accu
					FP	FN			
0	0.98	0.98	0.99	0.98	1	2	0.9964	0.0997	0.9788 ± 0.0188
1		0.98	0.96	0.97					

RANDOM FOREST



- Handling Complexity: well-suited for handling complex datasets with many features
- Feature Importance: provides insights into which features are most important for prediction

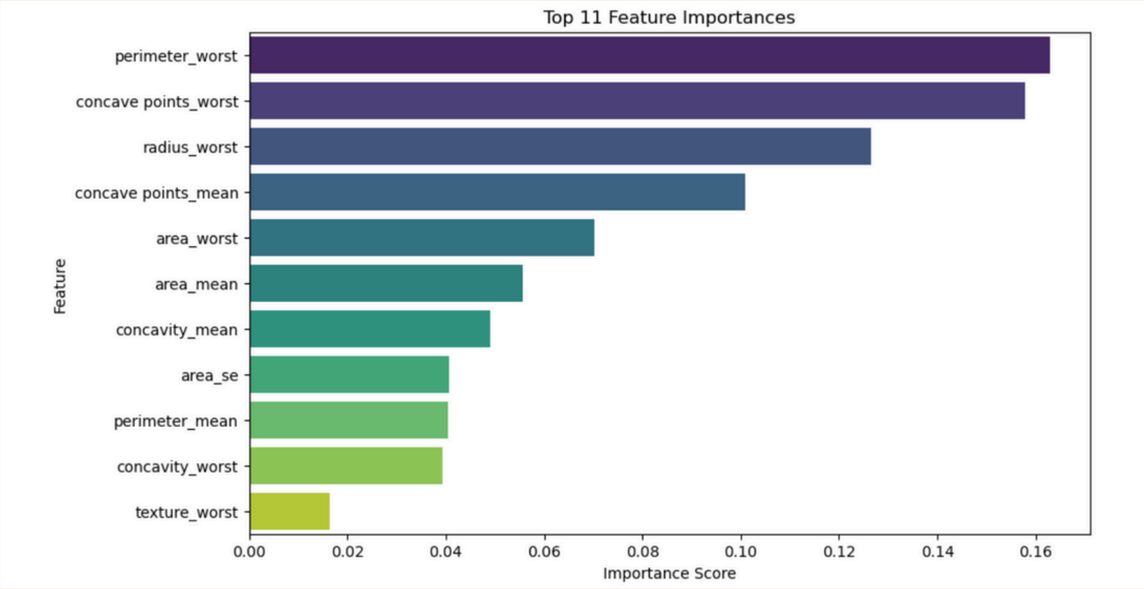
- Reduces the risk of overfitting
- Versatility and Accuracy: Random Forest can handle both continuous and categorical features, and its ensemble nature often leads to higher predictive accuracy compared to individual models like decision trees or logistic regression.
- Managing Imbalanced Data: This method is effective in handling class imbalance, which is common in medical datasets where malignant cases are typically fewer than benign ones.

RANDOM FOREST

Optimized Model with Rfe

Basic Random Forest Model

Random Forest Model with Feature Importance ranking

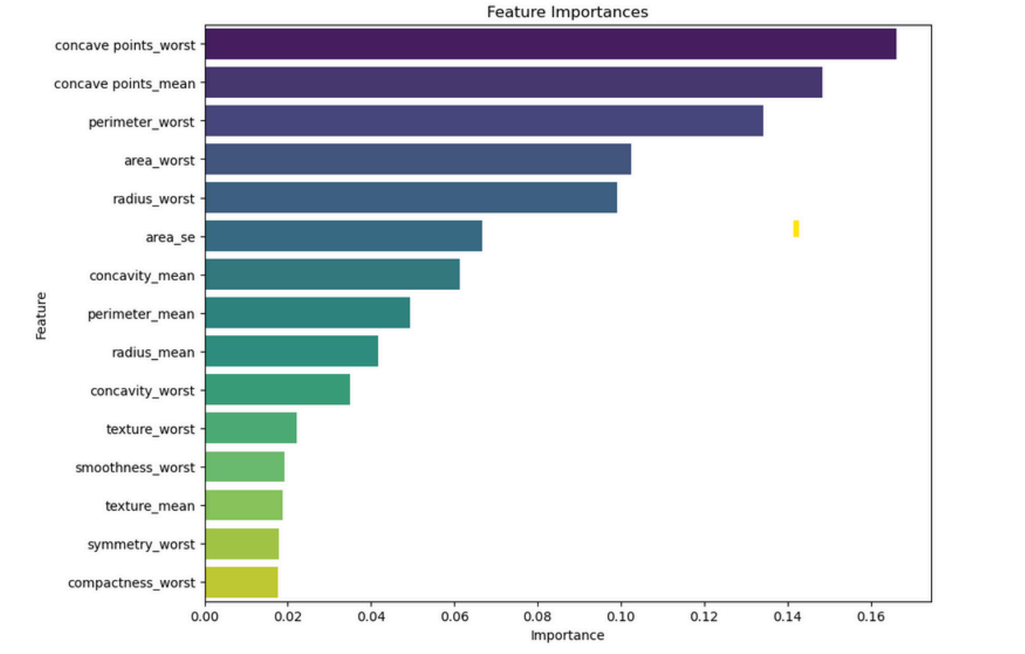


optimized the model using a hyperparameter tuning method-Gridsearch

-Perform RFE with Cross-Validation
-Select the feature count with the highest cross-validation score.

```
rf2_model = RandomForestClassifier(n_estimators=100, random_state=42)
scores = []
for i in range(5, X.shape[1] + 1, 5):
# Check in increments of 5 features
    rfe = RFE(estimator=rf2_model, n_features_to_select=i)
    score = cross_val_score(rfe, X, y, cv=5, scoring='accuracy').mean()
    scores.append((i, score))
# Find the best number of features
best_n_features = max(scores, key=lambda x: x[1])[0]
print(f"Best number of features: {best_n_features}")
```

```
Best number of features: 15
```

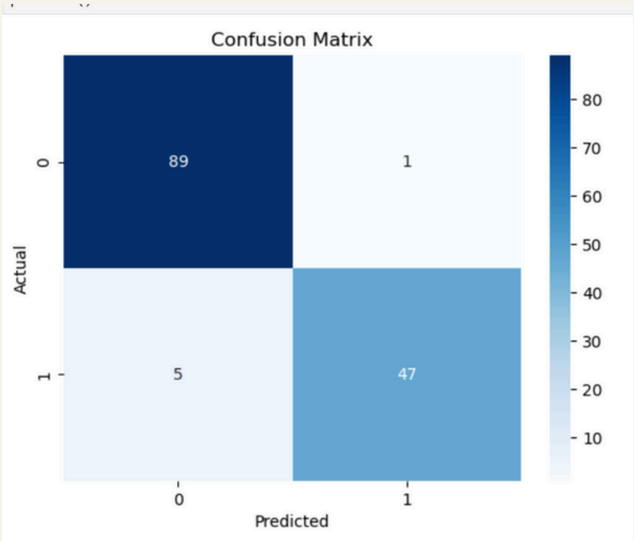


EVALUATION

Base

	precision	recall	f1-score	support
0	0.95	0.99	0.97	90
1	0.98	0.90	0.94	52
accuracy			0.96	142
macro avg	0.96	0.95	0.95	142
weighted avg	0.96	0.96	0.96	142

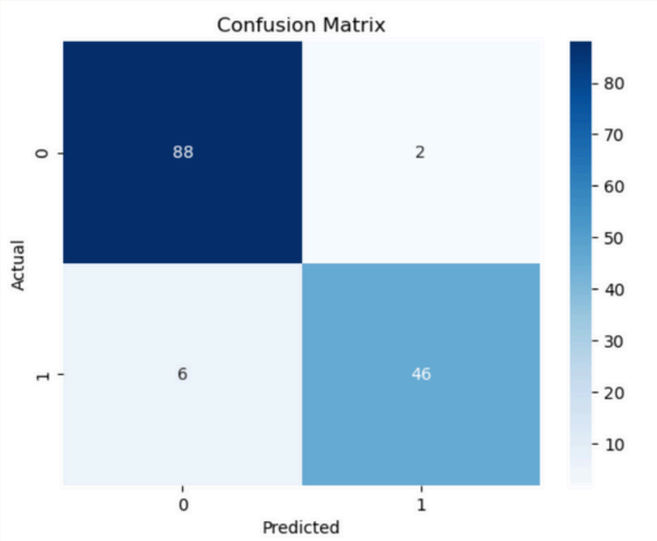
Accuracy:95.77%



Feature importance ranking

	precision	recall	f1-score	support
0	0.94	0.98	0.96	90
1	0.96	0.88	0.92	52
accuracy			0.94	142
macro avg	0.95	0.93	0.94	142
weighted avg	0.94	0.94	0.94	142

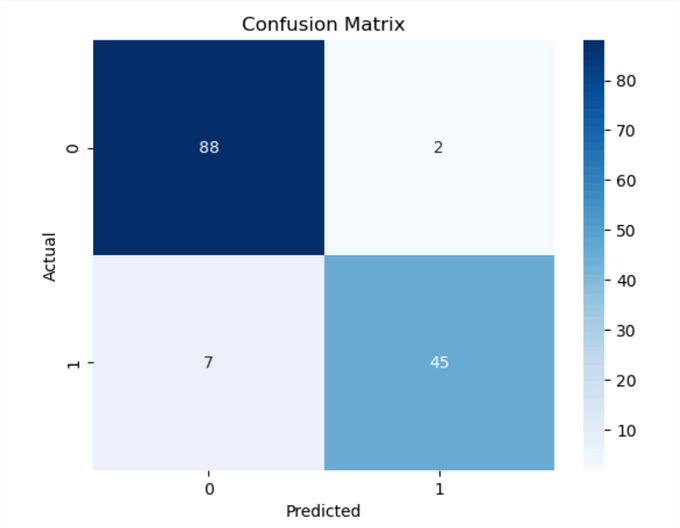
Accuracy:94.37%



Rfe optimized model

	precision	recall	f1-score	support
0	0.94	0.99	0.96	90
1	0.98	0.88	0.93	52
accuracy			0.95	142
macro avg	0.96	0.94	0.95	142
weighted avg	0.95	0.95	0.95	142

Accuracy:95.07%

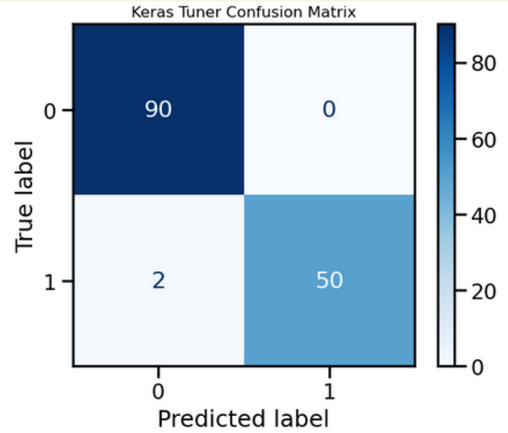


The increase in false negatives indicates that some of the removed features were essential for correctly identifying malignant cases. Their exclusion led to a decrease in the model's ability to detect these critical instances.

DEEP NEURAL NETWORK (KERAS TUNER)

Inital Model (Trial & Error)

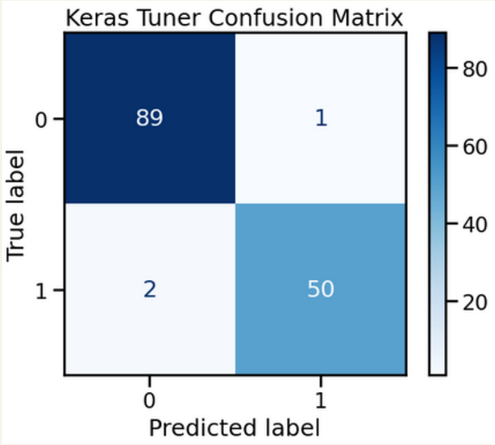
Accuracy = 0.9859



	precision	recall	f1-score	support
0	0.98	1.00	0.99	90
1	1.00	0.96	0.98	52
accuracy			0.99	142
macro avg	0.99	0.98	0.98	142
weighted avg	0.99	0.99	0.99	142

Final Model (Hyperparameter Search)

Accuracy = 0.97887



	precision	recall	f1-score	support
0	0.98	0.99	0.98	90
1	0.98	0.96	0.97	52
accuracy			0.98	142
macro avg	0.98	0.98	0.98	142
weighted avg	0.98	0.98	0.98	142

DEEP NEURAL NETWORK (KERAS TUNER)

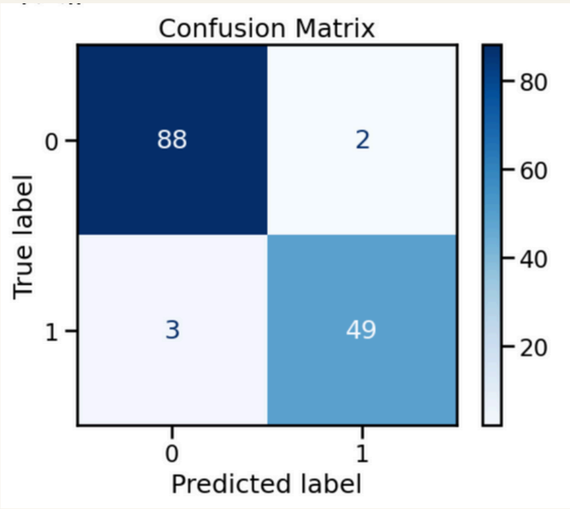
Observations

- Re-running the keras tuner resulted in accuracy fluctuations within 2% for both models
- The two models appeared to perform similarly as when re-running the models, the first model sometimes performed better than the second.
    - This is considered normal and is likely due to the random nature of neural network training.
        - e.g. Initial weights of the neural network are set randomly at the beginning of a training run.

SVM

```
SVC
SVC(C=1, class_weight='balanced', kernel='linear', random_state=42)
```

	precision	recall	f1-score	support
1	0.97	0.98	0.97	90
0	0.96	0.94	0.95	52
accuracy			0.96	142
macro avg	0.96	0.96	0.96	142
weighted avg	0.96	0.96	0.96	142

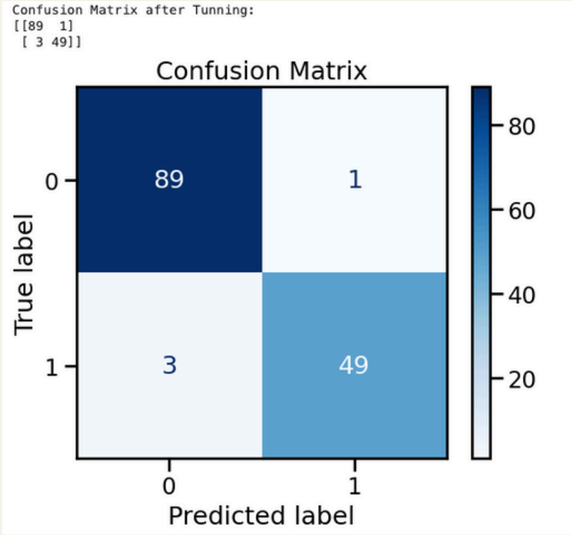


Model Accuracy

0.965

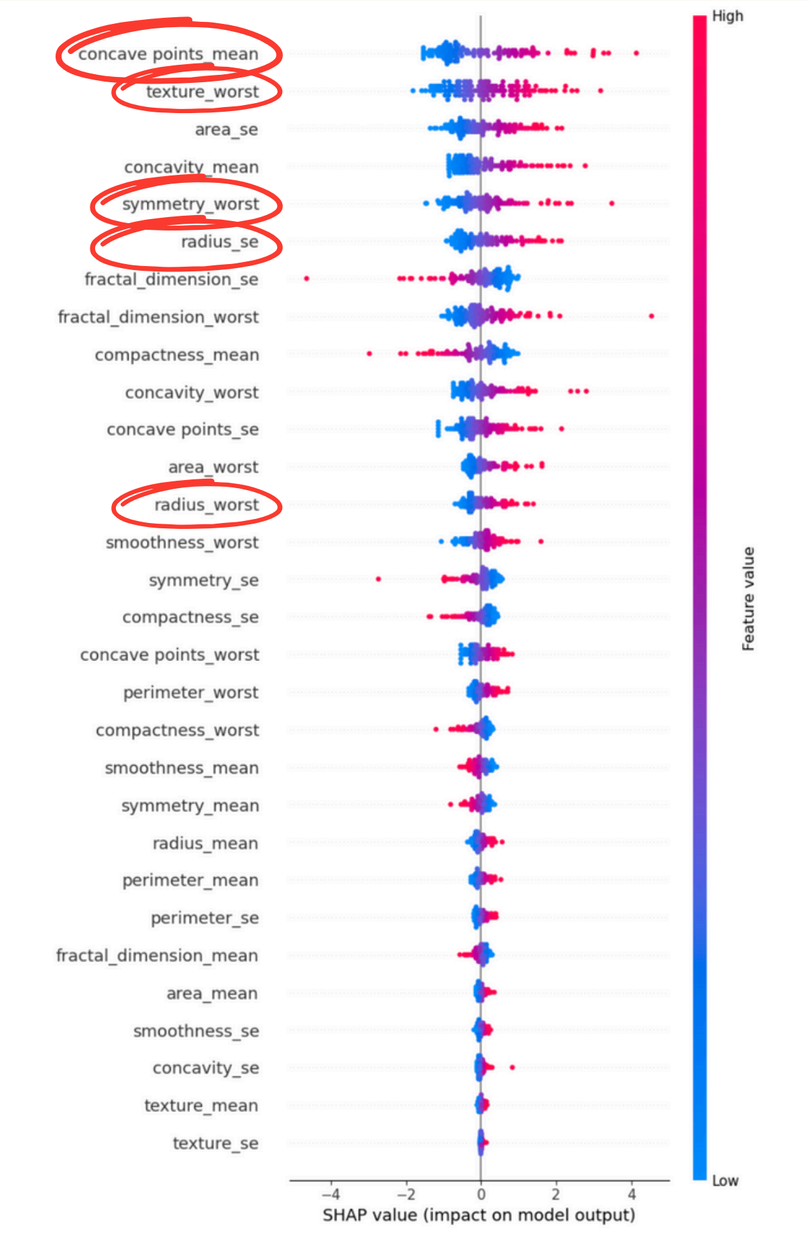
Perform grid search for tuning

	precision	recall	f1-score	support
0	0.97	0.99	0.98	90
1	0.98	0.94	0.96	52
accuracy			0.97	142
macro avg	0.97	0.97	0.97	142
weighted avg	0.97	0.97	0.97	142



Tuned Model Accuracy

0.972



THE RFE SELECTOR

'CONCAVE POINTS_MEAN'

'RADIUS_WORST'

'SYMMETRY_WORST'

'TEXTURE_WORST'

'RADIUS_SE'

	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean	com
567	20.60	29.33	140.1000	1265.000	0.11780	
568	7.76	24.54	53.8276	215.664	0.05263	

2 rows × 30 columns

```
X_unseen.shape
```

```
(2, 30)
```

```
X_unseen_scaled = scaler.transform(X_unseen)
X_unseen_selected = selector.transform(X_unseen_scaled)
```

```
pre = tunned_model.predict(X_unseen_selected)
print(pre)
```

```
[1 0]
```

```
y_unseen
```

```
567    1
568    0
Name: diagnosis, dtype: int64
```

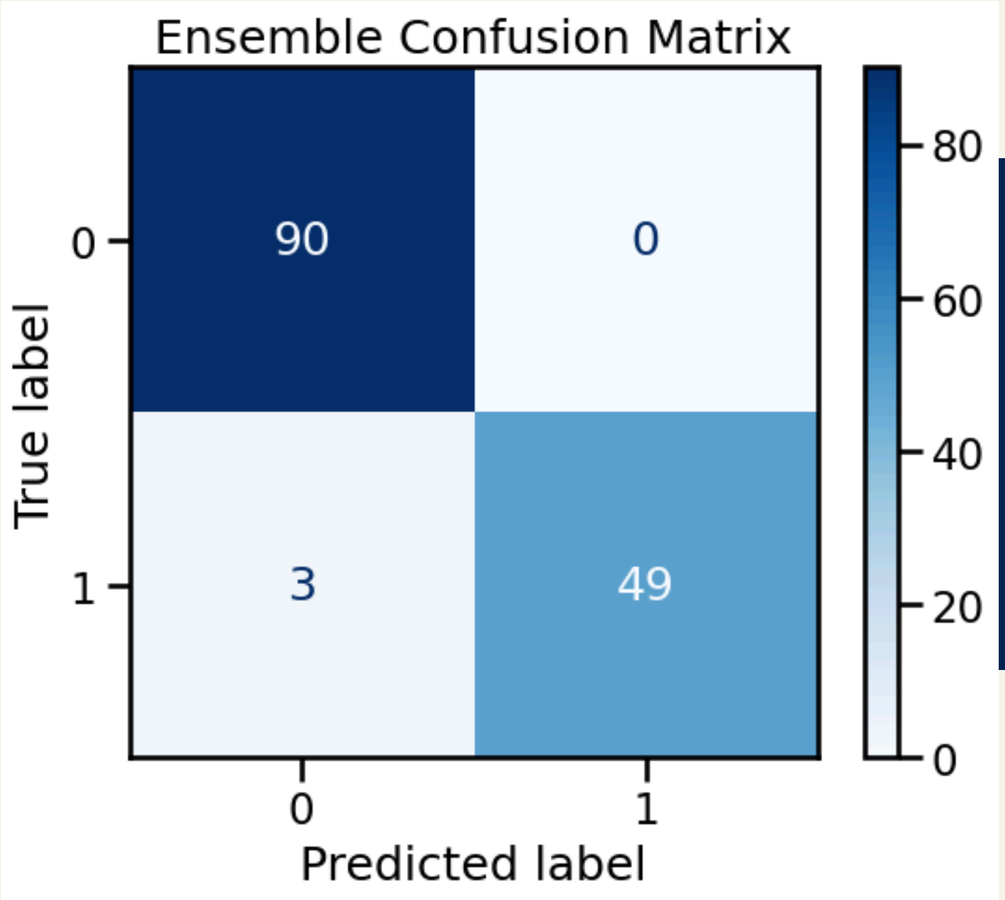
Yeah!! The prediction is same as the true label!!

```
Classification Report after select feature:
              precision    recall  f1-score   support

           0       0.97      0.99      0.98        90
           1       0.98      0.94      0.96        52

    accuracy                           0.97       142
   macro avg       0.97      0.97      0.97       142
weighted avg       0.97      0.97      0.97       142
```

ENSEMBLE APPROACH

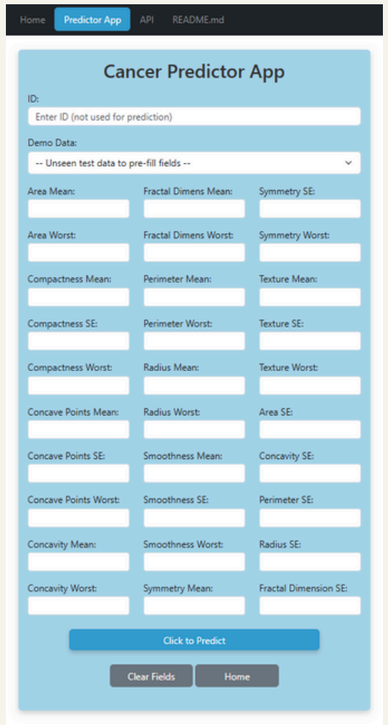
Max Count

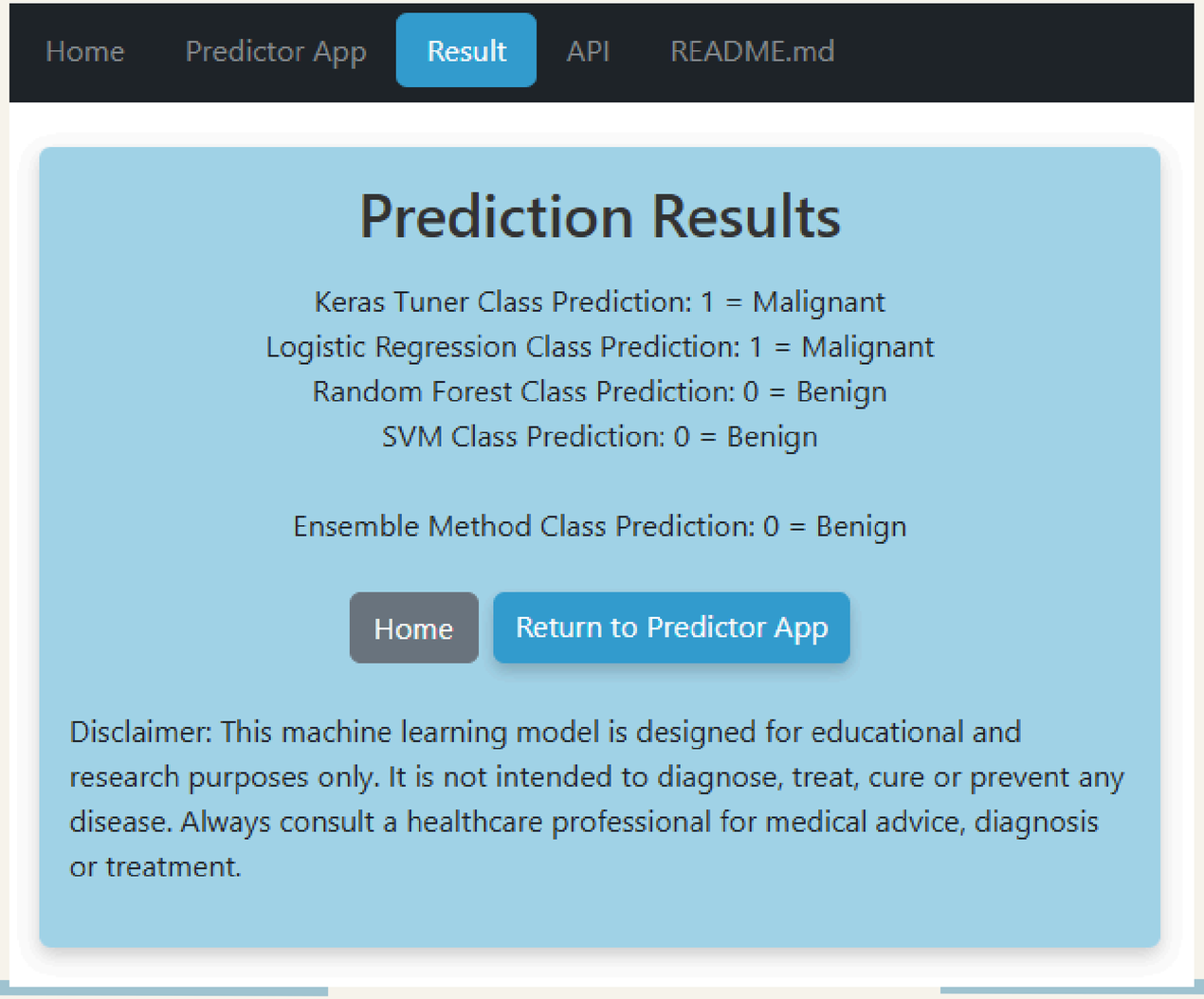
- Majority Rules
- SVM resolves the tie

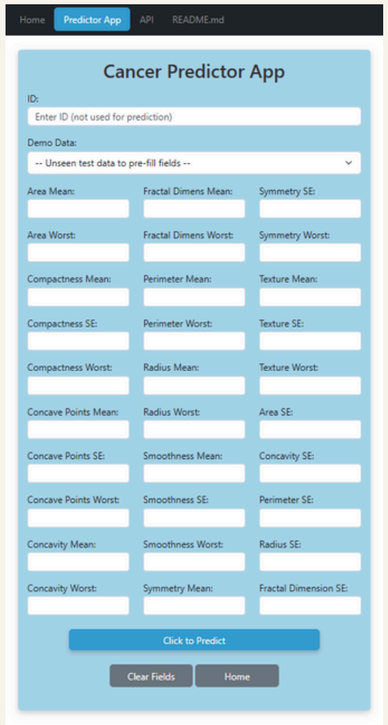


	precision	recall	f1-score	support
0	0.97	1.00	0.98	90
1	1.00	0.94	0.97	52
accuracy			0.98	142
macro avg	0.98	0.97	0.98	142
weighted avg	0.98	0.98	0.98	142

CONCLUSION







Disclaimer: This machine learning model is designed strictly for educational and research purposes. It is not intended to diagnose, treat, cure, or prevent any disease. Always consult a healthcare professional for medical advice, diagnosis, or treatment.

LIMITATIONS & FUTURE EXPLORATION

- Data Quantity
- Lack of Medical Knowledge
- Include different data types (e.g. imaging, clinical)
- Varied Data Sources (different countries/cities)

APP DEMONSTRATION

THANK YOU