iStroke

BRAIN STROKE PREDICTOR

Problem

Stroke is a silent killer

You may be at a risk and not know it.

- Upto 80% of the strokes are preventable.
- Stroke is second leading cause of death worldwide.
- Sixteen million people a year experience a stroke.
- One in four people experience stroke in their lifetime.
- Stroke can occur at any age and at any time and may of those occur to people with low to medium risk of having stroke.

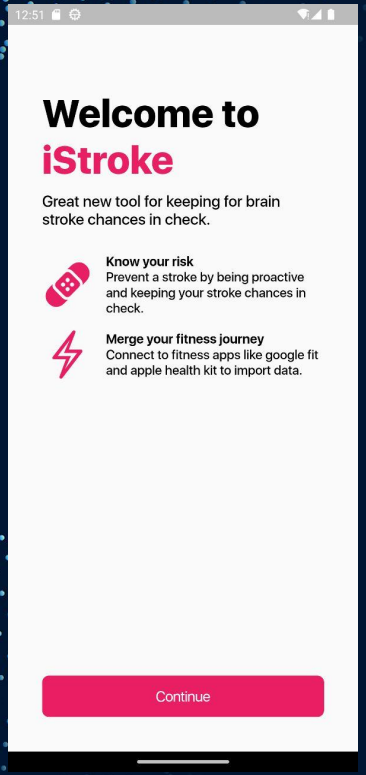
Solution - iStroke

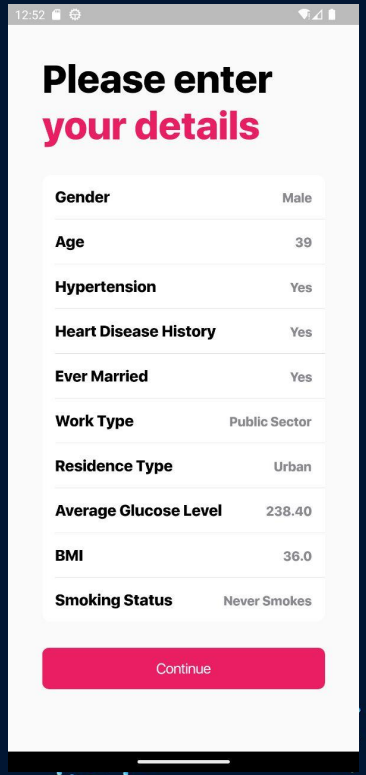
The app helps in early prediction of Strokes.

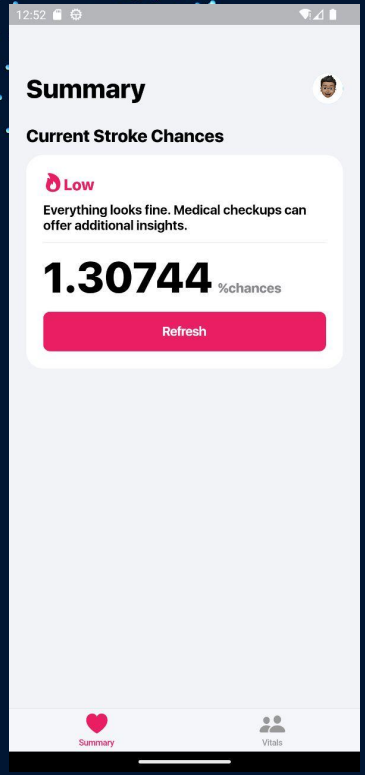
The app takes in data, like age, gender, hypertension, work type, residence type, etc. , from the user, passes that data to the ML model, the model processes the information and gives a prediction of risk of stroke.

The app also suggests whether or not the user requires to see a doctor.

UI


12:51
Welcome to
iStroke
Great new tool for keeping for brain stroke chances in check.
Know your risk
Prevent a stroke by being proactive and keeping your stroke chances in check.
Merge your fitness journey
Connect to fitness apps like google fit and apple health kit to import data.
Continue


12:52
Please enter
your details
Gender Male
Age 39
Hypertension Yes
Heart Disease History Yes
Ever Married Yes
Work Type Public Sector
Residence Type Urban
Average Glucose Level 238.40
BMI 36.0
Smoking Status Never Smokes
Continue


12:52
Summary
Current Stroke Chances
Low
Everything looks fine. Medical checkups can offer additional insights.
1.30744 %chances
Refresh
Summary
Vitals

Code Snippets

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.preprocessing import LabelEncoder
from sklearn.feature_selection import SelectKBest, f_classif
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, f1_score,classification_report,precision_score,recall_score
from imblearn.over_sampling import SMOTE
from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import RandomForestClassifier
from sklearn.svm import SVC
from xgboost import XGBClassifier
```

In [19]:
```
classifier = SelectKBest(score_func=f_classif,k=5)
fits = classifier.fit(data.drop('stroke',axis=1),data['stroke'])
x=pd.DataFrame(fits.scores_)
columns = pd.DataFrame(data.drop('stroke',axis=1).columns)
fscores = pd.concat([columns,x],axis=1)
fscores.columns = ['Attribute','Score']
fscores.sort_values(by='Score',ascending=False)
```

Out[19]:

	Attribute	Score
1	age	322.045639
3	heart_disease	91.884086
7	avg_glucose_level	89.971820
2	hypertension	88.245189
4	ever_married	59.199605
8	bmi	16.187059
5	work_type	8.729184
9	smoking_status	3.758131
6	Residence_type	1.354940
0	gender	0.391752

In [20]:
```
cols=fscores[fscores['Score']>50]['Attribute']
print(cols)
```

```
1                 age
2        hypertension
3       heart_disease
4        ever_married
7    avg_glucose_level
Name: Attribute, dtype: object
```

In [21]:
```
train_x,test_x,train_y,test_y=train_test_split(data[cols],data['stroke'],random_state=1255,test_size=0.25)

train_x.shape,test_x.shape,train_y.shape,test_y.shape
```

Out[21]: ((3735, 5), (1246, 5), (3735,), (1246,))

In [22]:
```
smote=SMOTE()
train_x,train_y=smote.fit_resample(train_x,train_y)
test_x,test_y=smote.fit_resample(test_x,test_y)
```

In [23]:
```
print(train_x.shape,train_y.shape,test_x.shape,test_y.shape)
```

```
(7090, 5) (7090,) (2376, 5) (2376,)
```

· Used SelectKBest algorithm to select the top K features from dataset that had the highest statistical significance for brain stroke prediction. Used SMOTE to balance the class distribution by generating synthetic data points for the minority class to help ML model better learn relevant patterns.

Code Snippets

```
xgc=XGBClassifier(objective='binary:logistic',n_estimators=100000,max_depth=5,learning_rate=0.001,n_jobs=-1)
xgc.fit(train_x,train_y)
predict=xgc.predict(test_x)
print('Accuracy --> ',accuracy_score(predict,test_y))
print('F1 Score --> ',f1_score(predict,test_y))
print('Classification Report  --> \n',classification_report(predict,test_y))
```

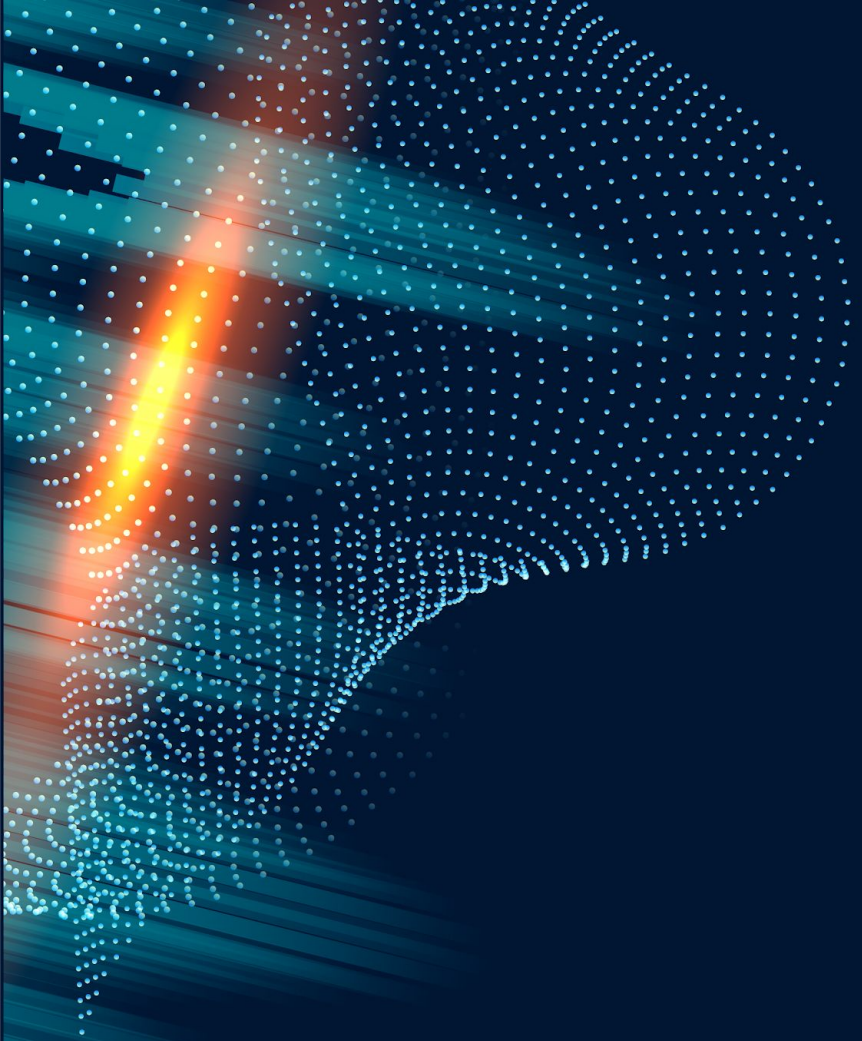
```
Accuracy -->  0.9183501683501684
F1 Score -->  0.9141592920353983
Classification Report  -->
              precision    recall  f1-score   support

           0       0.97      0.88      0.92      1304
           1       0.87      0.96      0.91      1072

    accuracy                           0.92      2376
   macro avg       0.92      0.92      0.92      2376
weighted avg       0.92      0.92      0.92      2376
```

```python
from flask import Flask, jsonify

app = Flask(__name__)

@app.route('/getprediction', methods=['GET'])
def get_prediction():
    testjson = {'prediction': 0.5}
    return jsonify(testjson)

if __name__ == '__main__':
    app.run(host='localhost', port=5000)
```

- Used XGBoost and SVC classifiers on selected features to make final prediction on likelihood of a stroke.
- Built the iStroke App on Flutter, used Flask for the backend and deployed it using Gunicorn and Heroku.

Future Prospects

Getting the paid API from google fit and connecting it to the app to fetch real time data to accurately predict stroke.



Thank You!