 **4222-SURYA GROUP OF INSTITUTIONS**  

**VIKARAVANDI -605 652**

**PROJECT NAME:**

**EARTHQUAKE-PREDICTION-USING-PYTHON**

**PHASE 2: INNOVATION**

**PREPARED BY:**

**M.GAYATHRI**

**REG NO:422221106008**

**ECE DEPARTMENT**

**INNOVATION:**

In this phase, we can explore innovative advanced techniques such as hyperparameter tuning and feature engineering to improve the prediction model's performance and also used ensemble methods and deep learning architectures to improve the prediction system's accuracy and robustness.

### **HYPERPARAMETER TUNING**

from keras.wrappers.scikit\_learn import KerasClassifier  
  
model = KerasClassifier(build\_fn=create\_model, verbose=0)  
  
# param\_grid = {  
# "neurons": [16, 64],   
# "batch\_size": [10, 20],   
# "epochs": [10],  
# "activation": ['sigmoid', 'relu'],  
# "optimizer": ['SGD', 'Adadelta'],  
# "loss": ['squared\_hinge']  
# }  
param\_grid = {  
 "neurons": [16],   
 "batch\_size": [10, 20],   
 "epochs": [10],  
 "activation": ['sigmoid', 'relu'],  
 "optimizer": ['SGD', 'Adadelta'],  
 "loss": ['squared\_hinge']  
}

X\_train = np.asarray(X\_train).astype(np.float32)  
y\_train = np.asarray(y\_train).astype(np.float32)  
X\_test = np.asarray(X\_test).astype(np.float32)  
y\_test = np.asarray(y\_test).astype(np.float32)

grid = GridSearchCV(estimator=model, param\_grid=param\_grid, n\_jobs=-1)  
grid\_result = grid.fit(X\_train, y\_train)  
  
best\_params = grid\_result.best\_params\_  
best\_params

2023-02-12 14:30:16.688729: I tensorflow/core/common\_runtime/process\_util.cc:146] Creating new thread pool with default inter op setting: 4. Tune using inter\_op\_parallelism\_threads for best performance.  
2023-02-12 14:30:16.721324: I tensorflow/core/common\_runtime/process\_util.cc:146] Creating new thread pool with default inter op setting: 4. Tune using inter\_op\_parallelism\_threads for best performance.  
2023-02-12 14:30:16.733601: I tensorflow/core/common\_runtime/process\_util.cc:146] Creating new thread pool with default inter op setting: 4. Tune using inter\_op\_parallelism\_threads for best performance.  
2023-02-12 14:30:16.761165: I tensorflow/core/common\_runtime/process\_util.cc:146] Creating new thread pool with default inter op setting: 4. Tune using inter\_op\_parallelism\_threads for best performance.  
2023-02-12 14:30:17.151828: I tensorflow/compiler/mlir/mlir\_graph\_optimization\_pass.cc:185] None of the MLIR Optimization Passes are enabled (registered 2)  
2023-02-12 14:30:17.151828: I tensorflow/compiler/mlir/mlir\_graph\_optimization\_pass.cc:185] None of the MLIR Optimization Passes are enabled (registered 2)  
2023-02-12 14:30:17.151827: I tensorflow/compiler/mlir/mlir\_graph\_optimization\_pass.cc:185] None of the MLIR Optimization Passes are enabled (registered 2)  
2023-02-12 14:30:17.164576: I tensorflow/compiler/mlir/mlir\_graph\_optimization\_pass.cc:185] None of the MLIR Optimization Passes are enabled (registered 2)  
2023-02-12 14:34:25.381389: I tensorflow/core/common\_runtime/process\_util.cc:146] Creating new thread pool with default inter op setting: 2. Tune using inter\_op\_parallelism\_threads for best performance.  
2023-02-12 14:34:25.461923: I tensorflow/compiler/mlir/mlir\_graph\_optimization\_pass.cc:185] None of the MLIR Optimization Passes are enabled (registered 2)

{'activation': 'sigmoid',  
 'batch\_size': 20,  
 'epochs': 10,  
 'loss': 'squared\_hinge',  
 'neurons': 16,  
 'optimizer': 'SGD'}

# Feature Engineering

data

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Date | Time | Latitude | Longitude | Type | Depth | Magnitude | Magnitude Type | Root Mean Square | Source | Location Source | Magnitude Source | Status |
| 0 | 01/02/1965 | 13:44:18 | 19.2460 | 145.6160 | Earthquake | 131.60 | 6.0 | MW | 1.022784 | ISCGEM | ISCGEM | ISCGEM | Automatic |
| 1 | 01/04/1965 | 11:29:49 | 1.8630 | 127.3520 | Earthquake | 80.00 | 5.8 | MW | 1.022784 | ISCGEM | ISCGEM | ISCGEM | Automatic |
| 2 | 01/05/1965 | 18:05:58 | -20.5790 | -173.9720 | Earthquake | 20.00 | 6.2 | MW | 1.022784 | ISCGEM | ISCGEM | ISCGEM | Automatic |
| 3 | 01/08/1965 | 18:49:43 | -59.0760 | -23.5570 | Earthquake | 15.00 | 5.8 | MW | 1.022784 | ISCGEM | ISCGEM | ISCGEM | Automatic |
| 4 | 01/09/1965 | 13:32:50 | 11.9380 | 126.4270 | Earthquake | 15.00 | 5.8 | MW | 1.022784 | ISCGEM | ISCGEM | ISCGEM | Automatic |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| 23404 | 12/28/2016 | 08:22:12 | 38.3917 | -118.8941 | Earthquake | 12.30 | 5.6 | ML | 0.189800 | NN | NN | NN | Reviewed |
| 23405 | 12/28/2016 | 09:13:47 | 38.3777 | -118.8957 | Earthquake | 8.80 | 5.5 | ML | 0.218700 | NN | NN | NN | Reviewed |
| 23406 | 12/28/2016 | 12:38:51 | 36.9179 | 140.4262 | Earthquake | 10.00 | 5.9 | MWW | 1.520000 | US | US | US | Reviewed |
| 23407 | 12/29/2016 | 22:30:19 | -9.0283 | 118.6639 | Earthquake | 79.00 | 6.3 | MWW | 1.430000 | US | US | US | Reviewed |
| 23408 | 12/30/2016 | 20:08:28 | 37.3973 | 141.4103 | Earthquake | 11.94 | 5.5 | MB | 0.910000 | US | US | US | Reviewed |

23409 rows × 13 columns

data['Month'] = data['Date'].apply(lambda x: x[0:2])  
data['Year'] = data['Date'].apply(lambda x: x[-4:])  
  
data = data.drop('Date', axis=1)

data['Month'] = data['Month'].astype(np.int)

data[data['Year'].str.contains('Z')]

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Time | Latitude | Longitude | Type | Depth | Magnitude | Magnitude Type | Root Mean Square | Source | Location Source | Magnitude Source | Status | Month | Year |
| 3378 | 1975-02-23T02:58:41.000Z | 8.017 | 124.075 | Earthquake | 623.0 | 5.6 | MB | 1.022784 | US | US | US | Reviewed | 19 | 000Z |
| 7510 | 1985-04-28T02:53:41.530Z | -32.998 | -71.766 | Earthquake | 33.0 | 5.6 | MW | 1.300000 | US | US | HRV | Reviewed | 19 | 530Z |
| 20647 | 2011-03-13T02:23:34.520Z | 36.344 | 142.344 | Earthquake | 10.1 | 5.8 | MWC | 1.060000 | US | US | GCMT | Reviewed | 20 | 520Z |

invalid\_year\_indices = data[data['Year'].str.contains('Z')].index  
  
data = data.drop(invalid\_year\_indices, axis=0).reset\_index(drop=True)

data['Year'] = data['Year'].astype(np.int)

data['Hour'] = data['Time'].apply(lambda x: np.int(x[0:2]))  
  
data = data.drop('Time', axis=1)

data

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Latitude | Longitude | Type | Depth | Magnitude | Magnitude Type | Root Mean Square | Source | Location Source | Magnitude Source | Status | Month | Year | Hour |
| 0 | 19.2460 | 145.6160 | Earthquake | 131.60 | 6.0 | MW | 1.022784 | ISCGEM | ISCGEM | ISCGEM | Automatic | 1 | 1965 | 13 |
| 1 | 1.8630 | 127.3520 | Earthquake | 80.00 | 5.8 | MW | 1.022784 | ISCGEM | ISCGEM | ISCGEM | Automatic | 1 | 1965 | 11 |
| 2 | -20.5790 | -173.9720 | Earthquake | 20.00 | 6.2 | MW | 1.022784 | ISCGEM | ISCGEM | ISCGEM | Automatic | 1 | 1965 | 18 |
| 3 | -59.0760 | -23.5570 | Earthquake | 15.00 | 5.8 | MW | 1.022784 | ISCGEM | ISCGEM | ISCGEM | Automatic | 1 | 1965 | 18 |
| 4 | 11.9380 | 126.4270 | Earthquake | 15.00 | 5.8 | MW | 1.022784 | ISCGEM | ISCGEM | ISCGEM | Automatic | 1 | 1965 | 13 |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| 23401 | 38.3917 | -118.8941 | Earthquake | 12.30 | 5.6 | ML | 0.189800 | NN | NN | NN | Reviewed | 12 | 2016 | 8 |
| 23402 | 38.3777 | -118.8957 | Earthquake | 8.80 | 5.5 | ML | 0.218700 | NN | NN | NN | Reviewed | 12 | 2016 | 9 |
| 23403 | 36.9179 | 140.4262 | Earthquake | 10.00 | 5.9 | MWW | 1.520000 | US | US | US | Reviewed | 12 | 2016 | 12 |
| 23404 | -9.0283 | 118.6639 | Earthquake | 79.00 | 6.3 | MWW | 1.430000 | US | US | US | Reviewed | 12 | 2016 | 22 |
| 23405 | 37.3973 | 141.4103 | Earthquake | 11.94 | 5.5 | MB | 0.910000 | US | US | US | Reviewed | 12 | 2016 | 20 |

23406 rows × 14 columns

data['Status'].unique()

array(['Automatic', 'Reviewed'], dtype=object)

data['Status'] = data['Status'].apply(lambda x: 1 if x == 'Reviewed' else 0)