In [1]:
```
import pandas as pd
from sklearn.tree import DecisionTreeClassifier # Import Decision Tree Classifier
from sklearn.model_selection import train_test_split # Import train_test_split function
from sklearn import metrics
from sklearn.preprocessing import LabelEncoder
from sklearn.preprocessing import StandardScaler
import joblib
from flask import Flask, request, jsonify, render_template
import pickle
```

In [2]:
```
#Load the csv file
data = pd.read_csv("/Users/srilathasirigala/Documents/Intern/Kerala_Loksabha_1962_2019(1).csv")
#data = pd.get_dummies(data, columns=["Ambalapuzha"])
#Alternatively, you can use scikit-learn's LabelEncoder to encode categorical variables as integer values. For
data.info()
data.head()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 296 entries, 0 to 295
Data columns (total 12 columns):
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   PC_Name               296 non-null    object
 1   No                    296 non-null    int64
 2   Type                  296 non-null    object
 3   State                 296 non-null    object
 4   Winning_candidate     296 non-null    object
 5   Party                 296 non-null    object
 6   Electors              296 non-null    int64
 7   Vote                  296 non-null    int64
 8   Turnout               296 non-null    float64
 9   Margin                296 non-null    int64
 10  Margin_in_percentage  296 non-null    float64
 11  year                  296 non-null    int64
dtypes: float64(2), int64(5), object(5)
memory usage: 27.9+ KB
```

Out[2]:

	PC_Name	No	Type	State	Winning_candidate	Party	Electors	Vote	Turnout	Margin	Margin_in_percentage	year
0	Ambalapuzha	143	GEN	Kerala	P. K. Vasudevan Nair	Communist Party Of India	445802	334846	75.1	11233	3.4	1962
1	Badagara	133	GEN	Kerala	A. V. Raghavan	Independent	463498	343312	74.1	72907	21.2	1962
2	Chirayinkil	147	GEN	Kerala	M. K. Kumaran	Communist Party Of India	437189	311762	71.3	33219	10.7	1962
3	Ernakulam	140	GEN	Kerala	A. M. Thomas	Indian National Congress	455280	363493	79.8	23399	6.4	1962
4	Kasergod	131	GEN	Kerala	A. K. Gopalan	Communist Party Of India	460358	308449	67.0	83363	27.0	1962

In [3]:
```
data.isnull().sum()
```

Out[3]:
```
PC_Name                 0
No                      0
Type                    0
State                   0
Winning_candidate       0
Party                   0
Electors                0
Vote                    0
Turnout                 0
Margin                  0
Margin_in_percentage    0
year                    0
dtype: int64
```

In [4]:
```
#Select the independent and dependent variables
X=data[['PC_Name','No','Type','State','Winning_candidate','Electors','Vote','Turnout','Margin','Margin_in_perce
y=data['Party']
```

In [5]:
```
# perform one-hot encoding on the categorical features
X = pd.get_dummies(X)
```

In [6]:
```
#split the data into train and test
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)
```

In [7]:
```
#feature Scaling
Sc=StandardScaler()
X_train=Sc.fit_transform(X_train)
X_test=Sc.transform(X_test)
```

In [8]:
```
from sklearn.ensemble import RandomForestClassifier

classifier=RandomForestClassifier()
```

In [9]:
```
classifier.fit(X_train, y_train)
```

Out[9]:
```
RandomForestClassifier()
```

In [10]:
```
pickle.dump(classifier,open("model2.pkl",'wb'))
```

In [11]:
```
#Save the trained model to a file

# Define the Flask app
app = Flask(__name__)
modele=pickle.load(open("model2.pkl",'rb'))



# Define the API endpoint for making predictions
@app.route("/")
def Home():
    return render_template("index1.html")

@app.route("/predict", methods=["POST"])
def predict():
    # Get the input features from the request
    data = request.get_json()
    features = [data["feature1"], data["feature2"], data["feature3"], data["feature4"],data["feature5"], data["

    # Make a prediction with the model
    prediction = modele.predict([features])[0]

    # Return the prediction as a JSON object
    response = {"prediction": prediction}
    return render_template('index1.html',prediction_text_="Kerala_Loksabha_1962_2019".format(prediction))

# Start the app
if __name__ == "__main__":
    app.run(debug=True)
```

```
 * Serving Flask app "__main__" (lazy loading)
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployment.
   Use a production WSGI server instead.
 * Debug mode: on
 * Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
 * Restarting with watchdog (fsevents)
Traceback (most recent call last):
  File "/Users/srilathasirigala/opt/anaconda3/lib/python3.9/site-packages/ipykernel_launcher.py", line 15, in <
module>
    from ipykernel import kernelapp as app
  File "/Users/srilathasirigala/opt/anaconda3/lib/python3.9/site-packages/ipykernel/kernelapp.py", line 18, in
<module>
    from IPython.core.application import (
  File "/Users/srilathasirigala/opt/anaconda3/lib/python3.9/site-packages/IPython/__init__.py", line 56, in <mo
dule>
    from .terminal.embed import embed
  File "/Users/srilathasirigala/opt/anaconda3/lib/python3.9/site-packages/IPython/terminal/embed.py", line 16,
in <module>
    from IPython.terminal.interactiveshell import TerminalInteractiveShell
  File "/Users/srilathasirigala/opt/anaconda3/lib/python3.9/site-packages/IPython/terminal/interactiveshell.py"
, line 35, in <module>
    from .debugger import TerminalPdb, Pdb
  File "/Users/srilathasirigala/opt/anaconda3/lib/python3.9/site-packages/IPython/terminal/debugger.py", line 6
, in <module>
    from IPython.core.completer import IPCompleter
  File "/Users/srilathasirigala/opt/anaconda3/lib/python3.9/site-packages/IPython/core/completer.py", line 146,
in <module>
    import jedi
  File "/Users/srilathasirigala/opt/anaconda3/lib/python3.9/site-packages/jedi/__init__.py", line 32, in <modul
e>
    from jedi.api import Script, Interpreter, set_debug_function, preload_module
  File "/Users/srilathasirigala/opt/anaconda3/lib/python3.9/site-packages/jedi/api/__init__.py", line 13, in <m
odule>
    import parso
  File "/Users/srilathasirigala/opt/anaconda3/lib/python3.9/site-packages/parso/__init__.py", line 42, in <modu
le>
    from parso.grammar import Grammar, load_grammar
  File "/Users/srilathasirigala/opt/anaconda3/lib/python3.9/site-packages/parso/grammar.py", line 13, in <modul
e>
    from parso.cache import parser_cache, load_module, try_to_save_module
  File "/Users/srilathasirigala/opt/anaconda3/lib/python3.9/site-packages/parso/cache.py", line 76, in <module>
    _default_cache_path = _get_default_cache_path()
  File "/Users/srilathasirigala/opt/anaconda3/lib/python3.9/site-packages/parso/cache.py", line 73, in _get_def
ault_cache_path
    return dir_.expanduser()
AttributeError: 'PosixPath' object has no attribute 'expanduser'
An exception has occurred, use %tb to see the full traceback.

SystemExit: 1
```

```
/Users/srilathasirigala/opt/anaconda3/lib/python3.9/site-packages/IPython/core/interactiveshell.py:3465: UserWa
rning: To exit: use 'exit', 'quit', or Ctrl-D.
  warn("To exit: use 'exit', 'quit', or Ctrl-D.", stacklevel=1)
```

In [ ]:

In [ ]: