In [1]:
```
!pip install category_encoders
import category_encoders as ce
import pandas as pd
import matplotlib.pyplot as plt
from sklearn import datasets
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn import tree
from sklearn.metrics import classification_report
from sklearn import preprocessing
```

```
Collecting category_encoders
  Downloading category_encoders-2.2.2-py2.py3-none-any.whl (80 kB)
Requirement already satisfied: patsy>=0.5.1 in c:\users\acer\anaconda3\lib\site-packages (from category_encoders) (0.5.1)
Requirement already satisfied: pandas>=0.21.1 in c:\users\acer\anaconda3\lib\site-packages (from category_encoders) (1.1.3)
Requirement already satisfied: scikit-learn>=0.20.0 in c:\users\acer\anaconda3\lib\site-packages (from category_encoders) (0.23.2)
Requirement already satisfied: scipy>=1.0.0 in c:\users\acer\anaconda3\lib\site-packages (from category_encoders) (1.5.2)
Requirement already satisfied: numpy>=1.14.0 in c:\users\acer\anaconda3\lib\site-packages (from category_encoders) (1.19.2)
Requirement already satisfied: statsmodels>=0.9.0 in c:\users\acer\anaconda3\lib\site-packages (from category_encoders) (0.12.0)
Requirement already satisfied: six in c:\users\acer\anaconda3\lib\site-packages (from patsy>=0.5.1->category_encoders) (1.15.0)
Requirement already satisfied: python-dateutil>=2.7.3 in c:\users\acer\anaconda3\lib\site-packages (from pandas>=0.21.1->category_encoders) (2.8.1)
Requirement already satisfied: pytz>=2017.2 in c:\users\acer\anaconda3\lib\site-packages (from pandas>=0.21.1->category_encoders) (2020.1)
Requirement already satisfied: threadpoolctl>=2.0.0 in c:\users\acer\anaconda3\lib\site-packages (from scikit-learn>=0.20.0->category_encoders) (2.1.0)
Requirement already satisfied: joblib>=0.11 in c:\users\acer\anaconda3\lib\site-packages (from scikit-learn>=0.20.0->category_encoders) (0.17.0)
Installing collected packages: category-encoders
Successfully installed category-encoders-2.2.2
```

In [2]:
```
# import fraud check set
fc = pd.read_csv('C:/Users/acer/Sandesh Pal/Data Science Assgn/DEsicion TRee.csv')
```

```
---------------------------------------------------------------------------
FileNotFoundError                         Traceback (most recent call last)
<ipython-input-2-1d6119d218a0> in <module>
      1 # import fraud check set
----> 2 fc = pd.read_csv('C:/Users/acer/Sandesh Pal/Data Science Assgn/DEsicion TRee.csv')

~\anaconda3\lib\site-packages\pandas\io\parsers.py in read_csv(filepath_or_buffer, sep, delimiter, header, names, index_col, usecols, squeeze, prefix, mangle_dupe_cols, dtype, engine, converters, true_values, false_values, skipinitialspace, skiprows, skipfooter, nrows, na_values, keep_default_na, na_filter, verbose, skip_blank_lines, parse_dates, infer_datetime_format, keep_date_col, date_parser, dayfirst, cache_dates, iterator, chunksize, compression, thousands, decimal, lineterminator, quotechar, quoting, doublequote, escapechar, comment, encoding, dialect, error_bad_lines, warn_bad_lines, delim_whitespace, low_memory, memory_map, float_precision)
    684     )
    685 
--> 686     return _read(filepath_or_buffer, kwds)
    687 
    688 

~\anaconda3\lib\site-packages\pandas\io\parsers.py in _read(filepath_or_buffer, kwds)
    450 
    451     # Create the parser.
--> 452     parser = TextFileReader(fp_or_buf, **kwds)
    453 
    454     if chunksize or iterator:

~\anaconda3\lib\site-packages\pandas\io\parsers.py in __init__(self, f, engine, **kwds)
    944             self.options["has_index_names"] = kwds["has_index_names"]
    945 
--> 946         self._make_engine(self.engine)
    947 
    948     def close(self):

~\anaconda3\lib\site-packages\pandas\io\parsers.py in _make_engine(self, engine)
   1176     def _make_engine(self, engine="c"):
   1177         if engine == "c":
-> 1178             self._engine = CParserWrapper(self.f, **self.options)
   1179         else:
   1180             if engine == "python":

~\anaconda3\lib\site-packages\pandas\io\parsers.py in __init__(self, src, **kwds)
   2006         kwds["usecols"] = self.usecols
   2007 
-> 2008         self._reader = parsers.TextReader(src, **kwds)
   2009         self.unnamed_cols = self._reader.unnamed_cols
   2010 

pandas\_libs\parsers.pyx in pandas._libs.parsers.TextReader.__cinit__()

pandas\_libs\parsers.pyx in pandas._libs.parsers.TextReader._setup_parser_source()

FileNotFoundError: [Errno 2] No such file or directory: 'C:/Users/acer/Sandesh Pal/Data Science Assgn/DEsicion TRee.csv'
```

In [4]:
```
# import fraud check set
fc = pd.read_csv('C:/Users/acer/Sandesh Pal/Data Science Assgn/DEsicion TRee/Fraud_check.csv')
```

In [5]:
```
# import fraud check set
fc = pd.read_csv('C:/Users/acer/Sandesh Pal/Data Science Assgn/DEsicion TRee/Fraud_check .xlsx')
```

```
---------------------------------------------------------------------------
FileNotFoundError                         Traceback (most recent call last)
<ipython-input-5-dfffa0083d23> in <module>
      1 # import fraud check set
----> 2 fc = pd.read_csv('C:/Users/acer/Sandesh Pal/Data Science Assgn/DEsicion TRee/Fraud_check .xlsx')

~\anaconda3\lib\site-packages\pandas\io\parsers.py in read_csv(filepath_or_buffer, sep, delimiter, header, names, index_col, usecols, squeeze, prefix, mangle_dupe_cols, dtype, engine, converters, true_values, false_values, skipinitialspace, skiprows, skipfooter, nrows, na_values, keep_default_na, na_filter, verbose, skip_blank_lines, parse_dates, infer_datetime_format, keep_date_col, date_parser, dayfirst, cache_dates, iterator, chunksize, compression, thousands, decimal, lineterminator, quotechar, quoting, doublequote, escapechar, comment, encoding, dialect, error_bad_lines, warn_bad_lines, delim_whitespace, low_memory, memory_map, float_precision)
    684     )
    685 
--> 686     return _read(filepath_or_buffer, kwds)
    687 
    688 

~\anaconda3\lib\site-packages\pandas\io\parsers.py in _read(filepath_or_buffer, kwds)
    450 
    451     # Create the parser.
--> 452     parser = TextFileReader(fp_or_buf, **kwds)
    453 
    454     if chunksize or iterator:

~\anaconda3\lib\site-packages\pandas\io\parsers.py in __init__(self, f, engine, **kwds)
    944             self.options["has_index_names"] = kwds["has_index_names"]
    945 
--> 946         self._make_engine(self.engine)
    947 
    948     def close(self):

~\anaconda3\lib\site-packages\pandas\io\parsers.py in _make_engine(self, engine)
   1176     def _make_engine(self, engine="c"):
   1177         if engine == "c":
-> 1178             self._engine = CParserWrapper(self.f, **self.options)
   1179         else:
   1180             if engine == "python":

~\anaconda3\lib\site-packages\pandas\io\parsers.py in __init__(self, src, **kwds)
   2006         kwds["usecols"] = self.usecols
   2007 
-> 2008         self._reader = parsers.TextReader(src, **kwds)
   2009         self.unnamed_cols = self._reader.unnamed_cols
   2010 

pandas\_libs\parsers.pyx in pandas._libs.parsers.TextReader.__cinit__()

pandas\_libs\parsers.pyx in pandas._libs.parsers.TextReader._setup_parser_source()

FileNotFoundError: [Errno 2] No such file or directory: 'C:/Users/acer/Sandesh Pal/Data Science Assgn/DEsicion TRee/Fraud_check .xlsx'
```

In [6]:
```
# import fraud check set
fc = pd.read_excel('C:/Users/acer/Sandesh Pal/Data Science Assgn/DEsicion TRee/Fraud_check .xlsx')
```

```
---------------------------------------------------------------------------
FileNotFoundError                         Traceback (most recent call last)
<ipython-input-6-5be54cbf894e> in <module>
      1 # import fraud check set
----> 2 fc = pd.read_excel('C:/Users/acer/Sandesh Pal/Data Science Assgn/DEsicion TRee/Fraud_check .xlsx')

~\anaconda3\lib\site-packages\pandas\util\_decorators.py in wrapper(*args, **kwargs)
    294                 )
    295                 warnings.warn(msg, FutureWarning, stacklevel=stacklevel)
--> 296             return func(*args, **kwargs)
    297 
    298         return wrapper

~\anaconda3\lib\site-packages\pandas\io\excel\_base.py in read_excel(io, sheet_name, header, names, index_col, usecols, squeeze, dtype, engine, converters, true_values, false_values, skiprows, nrows, na_values, keep_default_na, na_filter, verbose, parse_dates, date_parser, thousands, comment, skipfooter, convert_float, mangle_dupe_cols)
    302 
    303     if not isinstance(io, ExcelFile):
--> 304         io = ExcelFile(io, engine=engine)
    305     elif engine and engine != io.engine:
    306         raise ValueError(

~\anaconda3\lib\site-packages\pandas\io\excel\_base.py in __init__(self, path_or_buffer, engine)
    865         self._io = stringify_path(path_or_buffer)
    866 
--> 867         self._reader = self._engines[engine](self._io)
    868 
    869     def __fspath__(self):

~\anaconda3\lib\site-packages\pandas\io\excel\_xlrd.py in __init__(self, filepath_or_buffer)
     20         err_msg = "Install xlrd >= 1.0.0 for Excel support"
     21         import_optional_dependency("xlrd", extra=err_msg)
---> 22         super().__init__(filepath_or_buffer)
     23 
     24     @property

~\anaconda3\lib\site-packages\pandas\io\excel\_base.py in __init__(self, filepath_or_buffer)
    351             self.book = self.load_workbook(filepath_or_buffer)
    352         elif isinstance(filepath_or_buffer, str):
--> 353             self.book = self.load_workbook(filepath_or_buffer)
    354         elif isinstance(filepath_or_buffer, bytes):
    355             self.book = self.load_workbook(BytesIO(filepath_or_buffer))

~\anaconda3\lib\site-packages\pandas\io\excel\_xlrd.py in load_workbook(self, filepath_or_buffer)
     35             return open_workbook(file_contents=data)
     36         else:
---> 37             return open_workbook(filepath_or_buffer)
     38 
     39     @property

~\anaconda3\lib\site-packages\xlrd\__init__.py in open_workbook(filename, logfile, verbosity, use_mmap, file_contents, encoding_override, formatting_info, on_demand, ragged_rows)
    109     else:
    110         filename = os.path.expanduser(filename)
--> 111         with open(filename, "rb") as f:
    112             peek = f.read(peeksz)
    113     if peek == b"PK\x03\x04": # a ZIP file

FileNotFoundError: [Errno 2] No such file or directory: 'C:/Users/acer/Sandesh Pal/Data Science Assgn/DEsicion TRee/Fraud_check .xlsx'
```

In [7]:
```
# import fraud check set
fc = pd.read_csv('/Users/acer/Sandesh Pal/Data Science Assgn/DEsicion TRee/Fraud_check .xlsx')
```

```
---------------------------------------------------------------------------
FileNotFoundError                         Traceback (most recent call last)
<ipython-input-7-1c38d10fb806> in <module>
      1 # import fraud check set
----> 2 fc = pd.read_csv('/Users/acer/Sandesh Pal/Data Science Assgn/DEsicion TRee/Fraud_check .xlsx')

~\anaconda3\lib\site-packages\pandas\io\parsers.py in read_csv(filepath_or_buffer, sep, delimiter, header, names, index_col, usecols, squeeze, prefix, mangle_dupe_cols, dtype, engine, converters, true_values, false_values, skipinitialspace, skiprows, skipfooter, nrows, na_values, keep_default_na, na_filter, verbose, skip_blank_lines, parse_dates, infer_datetime_format, keep_date_col, date_parser, dayfirst, cache_dates, iterator, chunksize, compression, thousands, decimal, lineterminator, quotechar, quoting, doublequote, escapechar, comment, encoding, dialect, error_bad_lines, warn_bad_lines, delim_whitespace, low_memory, memory_map, float_precision)
    684     )
    685 
--> 686     return _read(filepath_or_buffer, kwds)
    687 
    688 

~\anaconda3\lib\site-packages\pandas\io\parsers.py in _read(filepath_or_buffer, kwds)
    450 
    451     # Create the parser.
--> 452     parser = TextFileReader(fp_or_buf, **kwds)
    453 
    454     if chunksize or iterator:

~\anaconda3\lib\site-packages\pandas\io\parsers.py in __init__(self, f, engine, **kwds)
    944             self.options["has_index_names"] = kwds["has_index_names"]
    945 
--> 946         self._make_engine(self.engine)
    947 
    948     def close(self):

~\anaconda3\lib\site-packages\pandas\io\parsers.py in _make_engine(self, engine)
   1176     def _make_engine(self, engine="c"):
   1177         if engine == "c":
-> 1178             self._engine = CParserWrapper(self.f, **self.options)
   1179         else:
   1180             if engine == "python":

~\anaconda3\lib\site-packages\pandas\io\parsers.py in __init__(self, src, **kwds)
   2006         kwds["usecols"] = self.usecols
   2007 
-> 2008         self._reader = parsers.TextReader(src, **kwds)
   2009         self.unnamed_cols = self._reader.unnamed_cols
   2010 

pandas\_libs\parsers.pyx in pandas._libs.parsers.TextReader.__cinit__()

pandas\_libs\parsers.pyx in pandas._libs.parsers.TextReader._setup_parser_source()

FileNotFoundError: [Errno 2] No such file or directory: '/Users/acer/Sandesh Pal/Data Science Assgn/DEsicion TRee/Fraud_check .xlsx'
```

In [8]:
```
# import fraud check set
fc = pd.read_excel('/Users/acer/Sandesh Pal/Data Science Assgn/DEsicion TRee/Fraud_check .xlsx')
```

```
---------------------------------------------------------------------------
FileNotFoundError                         Traceback (most recent call last)
<ipython-input-8-c731c84b18f9> in <module>
      1 # import fraud check set
----> 2 fc = pd.read_excel('/Users/acer/Sandesh Pal/Data Science Assgn/DEsicion TRee/Fraud_check .xlsx')

~\anaconda3\lib\site-packages\pandas\util\_decorators.py in wrapper(*args, **kwargs)
    294                 )
    295                 warnings.warn(msg, FutureWarning, stacklevel=stacklevel)
--> 296             return func(*args, **kwargs)
    297 
    298         return wrapper

~\anaconda3\lib\site-packages\pandas\io\excel\_base.py in read_excel(io, sheet_name, header, names, index_col, usecols, squeeze, dtype, engine, converters, true_values, false_values, skiprows, nrows, na_values, keep_default_na, na_filter, verbose, parse_dates, date_parser, thousands, comment, skipfooter, convert_float, mangle_dupe_cols)
    302 
    303     if not isinstance(io, ExcelFile):
--> 304         io = ExcelFile(io, engine=engine)
    305     elif engine and engine != io.engine:
    306         raise ValueError(

~\anaconda3\lib\site-packages\pandas\io\excel\_base.py in __init__(self, path_or_buffer, engine)
    865         self._io = stringify_path(path_or_buffer)
    866 
--> 867         self._reader = self._engines[engine](self._io)
    868 
    869     def __fspath__(self):

~\anaconda3\lib\site-packages\pandas\io\excel\_xlrd.py in __init__(self, filepath_or_buffer)
     20         err_msg = "Install xlrd >= 1.0.0 for Excel support"
     21         import_optional_dependency("xlrd", extra=err_msg)
---> 22         super().__init__(filepath_or_buffer)
     23 
     24     @property

~\anaconda3\lib\site-packages\pandas\io\excel\_base.py in __init__(self, filepath_or_buffer)
    351             self.book = self.load_workbook(filepath_or_buffer)
    352         elif isinstance(filepath_or_buffer, str):
--> 353             self.book = self.load_workbook(filepath_or_buffer)
    354         elif isinstance(filepath_or_buffer, bytes):
    355             self.book = self.load_workbook(BytesIO(filepath_or_buffer))

~\anaconda3\lib\site-packages\pandas\io\excel\_xlrd.py in load_workbook(self, filepath_or_buffer)
     35             return open_workbook(file_contents=data)
     36         else:
---> 37             return open_workbook(filepath_or_buffer)
     38 
     39     @property

~\anaconda3\lib\site-packages\xlrd\__init__.py in open_workbook(filename, logfile, verbosity, use_mmap, file_contents, encoding_override, formatting_info, on_demand, ragged_rows)
    109     else:
    110         filename = os.path.expanduser(filename)
--> 111         with open(filename, "rb") as f:
    112             peek = f.read(peeksz)
    113     if peek == b"PK\x03\x04": # a ZIP file

FileNotFoundError: [Errno 2] No such file or directory: '/Users/acer/Sandesh Pal/Data Science Assgn/DEsicion TRee/Fraud_check .xlsx'
```

In [9]:
```
# import fraud check set
fc = pd.read_excel('C:/Users/acer/Sandesh Pal/Data Science Assgn/DEsicion TRee/Fraud_check.xlsx')
```

```
---------------------------------------------------------------------------
FileNotFoundError                         Traceback (most recent call last)
<ipython-input-9-de0c62fab913> in <module>
      1 # import fraud check set
----> 2 fc = pd.read_excel('C:/Users/acer/Sandesh Pal/Data Science Assgn/DEsicion TRee/Fraud_check.xlsx')

~\anaconda3\lib\site-packages\pandas\util\_decorators.py in wrapper(*args, **kwargs)
    294                 )
    295                 warnings.warn(msg, FutureWarning, stacklevel=stacklevel)
--> 296             return func(*args, **kwargs)
    297 
    298         return wrapper

~\anaconda3\lib\site-packages\pandas\io\excel\_base.py in read_excel(io, sheet_name, header, names, index_col, usecols, squeeze, dtype, engine, converters, true_values, false_values, skiprows, nrows, na_values, keep_default_na, na_filter, verbose, parse_dates, date_parser, thousands, comment, skipfooter, convert_float, mangle_dupe_cols)
    302 
    303     if not isinstance(io, ExcelFile):
--> 304         io = ExcelFile(io, engine=engine)
    305     elif engine and engine != io.engine:
    306         raise ValueError(

~\anaconda3\lib\site-packages\pandas\io\excel\_base.py in __init__(self, path_or_buffer, engine)
    865         self._io = stringify_path(path_or_buffer)
    866 
--> 867         self._reader = self._engines[engine](self._io)
    868 
    869     def __fspath__(self):

~\anaconda3\lib\site-packages\pandas\io\excel\_xlrd.py in __init__(self, filepath_or_buffer)
     20         err_msg = "Install xlrd >= 1.0.0 for Excel support"
     21         import_optional_dependency("xlrd", extra=err_msg)
---> 22         super().__init__(filepath_or_buffer)
     23 
     24     @property

~\anaconda3\lib\site-packages\pandas\io\excel\_base.py in __init__(self, filepath_or_buffer)
    351             self.book = self.load_workbook(filepath_or_buffer)
    352         elif isinstance(filepath_or_buffer, str):
--> 353             self.book = self.load_workbook(filepath_or_buffer)
    354         elif isinstance(filepath_or_buffer, bytes):
    355             self.book = self.load_workbook(BytesIO(filepath_or_buffer))

~\anaconda3\lib\site-packages\pandas\io\excel\_xlrd.py in load_workbook(self, filepath_or_buffer)
     35             return open_workbook(file_contents=data)
     36         else:
---> 37             return open_workbook(filepath_or_buffer)
     38 
     39     @property

~\anaconda3\lib\site-packages\xlrd\__init__.py in open_workbook(filename, logfile, verbosity, use_mmap, file_contents, encoding_override, formatting_info, on_demand, ragged_rows)
    109     else:
    110         filename = os.path.expanduser(filename)
--> 111         with open(filename, "rb") as f:
    112             peek = f.read(peeksz)
    113     if peek == b"PK\x03\x04": # a ZIP file

FileNotFoundError: [Errno 2] No such file or directory: 'C:/Users/acer/Sandesh Pal/Data Science Assgn/DEsicion TRee/Fraud_check.xlsx'
```

In [10]:
```
series = pd.read_excel("/Users/acer/Sandesh Pal/Data Science Assgn/DEsicion TRee/Fraud_check.xlsx", header=0, index_col=0, parse_dates=True)
```

```
---------------------------------------------------------------------------
FileNotFoundError                         Traceback (most recent call last)
<ipython-input-10-09c9b3f636b7> in <module>
----> 1 series = pd.read_excel("/Users/acer/Sandesh Pal/Data Science Assgn/DEsicion TRee/Fraud_check.xlsx", header=0, index_col=0, parse_dates=True)

~\anaconda3\lib\site-packages\pandas\util\_decorators.py in wrapper(*args, **kwargs)
    294                 )
    295                 warnings.warn(msg, FutureWarning, stacklevel=stacklevel)
--> 296             return func(*args, **kwargs)
    297 
    298         return wrapper

~\anaconda3\lib\site-packages\pandas\io\excel\_base.py in read_excel(io, sheet_name, header, names, index_col, usecols, squeeze, dtype, engine, converters, true_values, false_values, skiprows, nrows, na_values, keep_default_na, na_filter, verbose, parse_dates, date_parser, thousands, comment, skipfooter, convert_float, mangle_dupe_cols)
    302 
    303     if not isinstance(io, ExcelFile):
--> 304         io = ExcelFile(io, engine=engine)
    305     elif engine and engine != io.engine:
    306         raise ValueError(

~\anaconda3\lib\site-packages\pandas\io\excel\_base.py in __init__(self, path_or_buffer, engine)
    865         self._io = stringify_path(path_or_buffer)
    866 
--> 867         self._reader = self._engines[engine](self._io)
    868 
    869     def __fspath__(self):

~\anaconda3\lib\site-packages\pandas\io\excel\_xlrd.py in __init__(self, filepath_or_buffer)
     20         err_msg = "Install xlrd >= 1.0.0 for Excel support"
     21         import_optional_dependency("xlrd", extra=err_msg)
---> 22         super().__init__(filepath_or_buffer)
     23 
     24     @property

~\anaconda3\lib\site-packages\pandas\io\excel\_base.py in __init__(self, filepath_or_buffer)
    351             self.book = self.load_workbook(filepath_or_buffer)
    352         elif isinstance(filepath_or_buffer, str):
--> 353             self.book = self.load_workbook(filepath_or_buffer)
    354         elif isinstance(filepath_or_buffer, bytes):
    355             self.book = self.load_workbook(BytesIO(filepath_or_buffer))

~\anaconda3\lib\site-packages\pandas\io\excel\_xlrd.py in load_workbook(self, filepath_or_buffer)
     35             return open_workbook(file_contents=data)
     36         else:
---> 37             return open_workbook(filepath_or_buffer)
     38 
     39     @property

~\anaconda3\lib\site-packages\xlrd\__init__.py in open_workbook(filename, logfile, verbosity, use_mmap, file_contents, encoding_override, formatting_info, on_demand, ragged_rows)
    109     else:
    110         filename = os.path.expanduser(filename)
--> 111         with open(filename, "rb") as f:
    112             peek = f.read(peeksz)
    113     if peek == b"PK\x03\x04": # a ZIP file

FileNotFoundError: [Errno 2] No such file or directory: '/Users/acer/Sandesh Pal/Data Science Assgn/DEsicion TRee/Fraud_check.xlsx'
```

In [11]:
```
# import fraud check set
fc = pd.read_csv('/Users/acer/Sandesh Pal/Data Science Assgn/DEsicion TRee/Fraud_check.csv')
```

In [12]:
```
fc
```

Out[12]:

	Undergrad	Marital.Status	Taxable.Income	City.Population	Work.Experience	Urban
0	NO	Single	68833	50047	10	YES
1	YES	Divorced	33700	134075	18	YES
2	NO	Married	36925	160205	30	YES
3	YES	Single	50190	193264	15	YES
4	NO	Married	81002	27533	28	NO
...	...	...	...	...	...	...
595	YES	Divorced	76340	39492	7	YES
596	YES	Divorced	69967	55369	2	YES
597	NO	Divorced	47334	154058	0	YES
598	YES	Married	98592	180083	17	NO
599	NO	Divorced	96519	158137	16	NO

600 rows × 6 columns

In [13]:
```
# checking for null values
fc.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 600 entries, 0 to 599
Data columns (total 6 columns):
 #   Column           Non-Null Count  Dtype 
---  ------           --------------  ----- 
 0   Undergrad        600 non-null    object
 1   Marital.Status   600 non-null    object
 2   Taxable.Income   600 non-null    int64 
 3   City.Population  600 non-null    int64 
 4   Work.Experience  600 non-null    int64 
 5   Urban            600 non-null    object
dtypes: int64(3), object(3)
memory usage: 28.2+ KB
```

In [14]:
```
fc.describe()
```

Out[14]:

	Taxable.Income	City.Population	Work.Experience
count	600.000000	600.000000	600.000000
mean	55208.375000	108747.368333	15.558333
std	26204.827597	49850.075134	8.842147
min	10003.000000	25779.000000	0.000000
25%	32871.500000	66966.750000	8.000000
50%	55074.500000	106493.500000	15.000000
75%	78611.750000	150114.250000	24.000000
max	99619.000000	199778.000000	30.000000

In [15]:
```
import category_encoders as ce
# encode variables with ordinal encoding
encoder = ce.OrdinalEncoder(cols=['Undergrad', 'Marital.Status', 'Urban'])
fc1 = encoder.fit_transform(fc)
```

```
C:\Users\acer\anaconda3\lib\site-packages\category_encoders\utils.py:21: FutureWarning: is_categorical is deprecated and will be removed in a future version.  Use is_categorical_dtype instead
  elif pd.api.types.is_categorical(cols):
```

In [17]:
```
# Converting the Target column i.e. Taxable Income into Categorical value
tax_val = []
for value in fc["Taxable.Income"]:
    if value<=30000:
        tax_val.append("Risky")
    else:
        tax_val.append("Good")

fc1["tax_val"]= tax_val
```

In [18]:
```
fc1
```

Out[18]:

	Undergrad	Marital.Status	Taxable.Income	City.Population	Work.Experience	Urban	tax_val
0	1	1	68833	50047	10	1	Good
1	2	2	33700	134075	18	1	Good
2	1	3	36925	160205	30	1	Good
3	2	1	50190	193264	15	1	Good
4	1	3	81002	27533	28	2	Good
...	...	...	...	...	...	...	...
595	2	2	76340	39492	7	1	Good
596	2	2	69967	55369	2	1	Good
597	1	2	47334	154058	0	1	Good
598	2	3	98592	180083	17	2	Good
599	1	2	96519	158137	16	2	Good

600 rows × 7 columns

In [19]:
```
x = fc1.drop(['tax_val','Taxable.Income'], axis =1)
y = fc1['tax_val']
```

In [20]:
```
x.head()
```

Out[20]:

	Undergrad	Marital.Status	City.Population	Work.Experience	Urban
0	1	1	50047	10	1
1	2	2	134075	18	1
2	1	3	160205	30	1
3	2	1	193264	15	1
4	1	3	27533	28	2

In [21]:
```
y.value_counts()
```

Out[21]:
```
Good     476
Risky    124
Name: tax_val, dtype: int64
```

In [22]:
```
# Splitting data into training and testing data set
x_train, x_test,y_train,y_test = train_test_split(x,y, test_size=0.3,random_state=40)
```

Building Decision Tree Classifier using Entropy Criteria Iteration-1: Max Depth = 2

In [23]:
```
model1 = DecisionTreeClassifier(criterion = 'entropy',max_depth=2)
model1.fit(x_train,y_train)
```

Out[23]: `DecisionTreeClassifier(criterion='entropy', max_depth=2)`

In [24]:
```
#Predicting on test data
pred_test1 = model1.predict(x_test)
#Accuracy on test data
print('Test data Accuracy is:',np.mean(pred_test1==y_test))
#Predicting on train data
pred_train1 = model1.predict(x_train)
#Accuracy on train data
print('Train data Accuracy is:',np.mean(pred_train1==y_train))
```

```
Test data Accuracy is: 0.7777777777777778
Train data Accuracy is: 0.8
```

Iteration-2: Max Depth = 3

In [25]:
```
model2 = DecisionTreeClassifier(criterion = 'entropy',max_depth=3)
model2.fit(x_train,y_train)
```

Out[25]: `DecisionTreeClassifier(criterion='entropy', max_depth=3)`

In [26]:
```
#Predicting on test data
pred_test2 = model2.predict(x_test)
#Accuracy on test data
print('Test data Accuracy is:',np.mean(pred_test2==y_test))
#Predicting on train data
pred_train2 = model2.predict(x_train)
#Accuracy on train data
print('Train data Accuracy is:',np.mean(pred_train2==y_train))
```

```
Test data Accuracy is: 0.7722222222222223
Train data Accuracy is: 0.8047619047619048
```

Iteration-3: Max Depth = 4

In [27]:
```
model3 = DecisionTreeClassifier(criterion = 'entropy',max_depth=4)
model3.fit(x_train,y_train)
```

Out[27]: `DecisionTreeClassifier(criterion='entropy', max_depth=4)`

In [28]:
```
#Predicting on test data
pred_test3 = model3.predict(x_test)
#Accuracy on test data
print('Test data Accuracy is:',np.mean(pred_test3==y_test))
#Predicting on train data
pred_train3 = model3.predict(x_train)
#Accuracy on train data
print('Train data Accuracy is:',np.mean(pred_train3==y_train))
```

```
Test data Accuracy is: 0.7611111111111111
Train data Accuracy is: 0.8119047619047619
```

Iteration-4: Max Depth = 5

In [29]:
```
model4 = DecisionTreeClassifier(criterion = 'entropy',max_depth=5)
model4.fit(x_train,y_train)
```

Out[29]: `DecisionTreeClassifier(criterion='entropy', max_depth=5)`

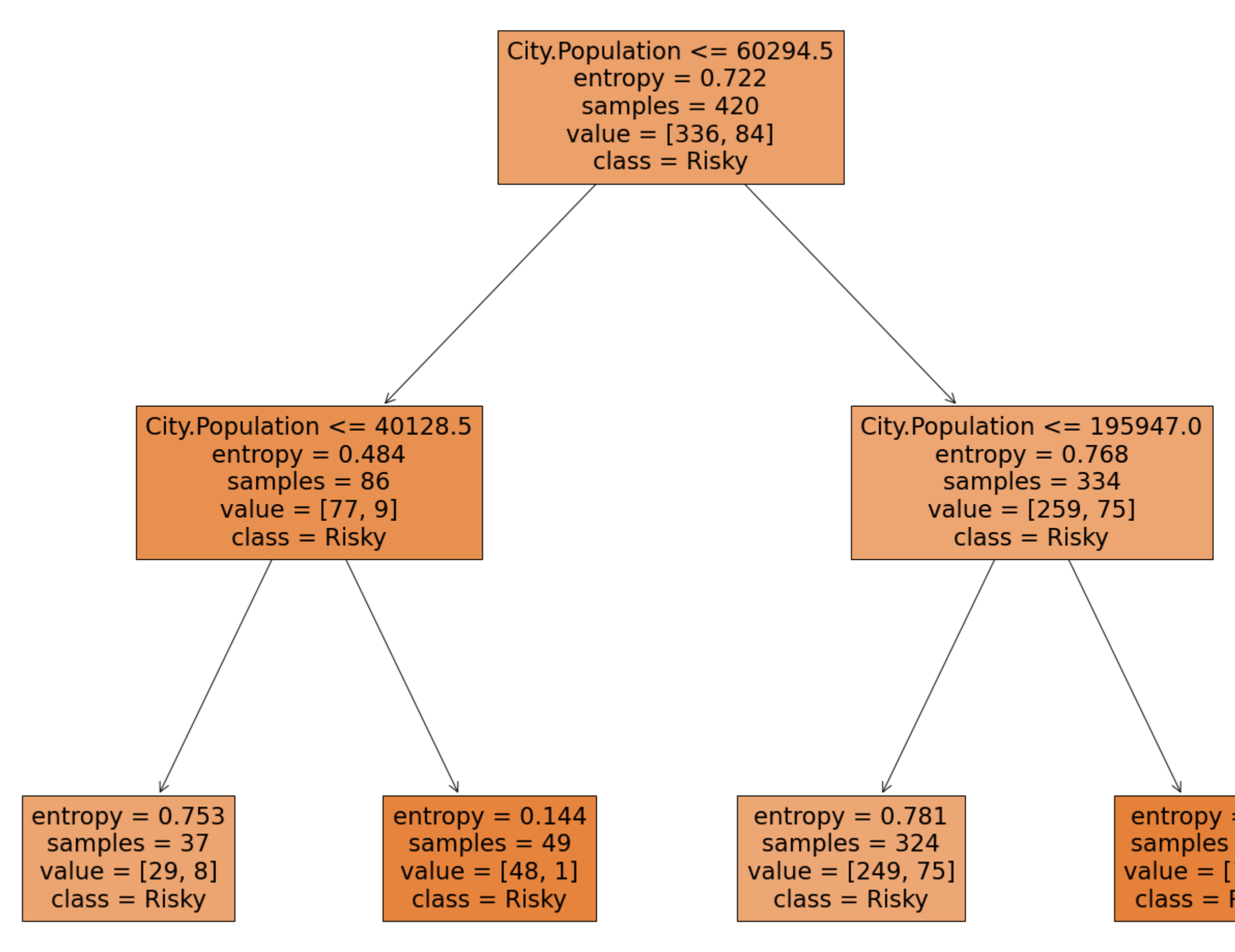
In [30]:
```
#Predicting on test data
pred_test4 = model4.predict(x_test)
#Accuracy on test data
print('Test data Accuracy is:',np.mean(pred_test4==y_test))
#Predicting on train data
pred_train4 = model4.predict(x_train)
#Accuracy on train data
print('Train data Accuracy is:',np.mean(pred_train4==y_train))
```

```
Test data Accuracy is: 0.7666666666666667
Train data Accuracy is: 0.8166666666666667
```

We get the best test results at Iteration-1 max depth = 2. so we will consider that final

In [31]:
```
# let's plot the decision tree
fig = plt.figure(figsize=(25,20))
fig = tree.plot_tree(model1,
 feature_names= ['Undergrad','Marital.Status','City.Population','Work.Experience'], class_names= ['Risky', 'Good'], filled=True)
plt.title('Decision tree using Entropy',fontsize=22)
```

Out[31]: `Text(0.5, 1.0, 'Decision tree using Entropy')`



In [ ]:
```
From the above decision tree the most important feature affecting the class is City Population
Building Decision Tree Classifier (CART) using Gini Criteria
Iteration-1: Max Depth = 2
```

In [32]:
```
from sklearn.tree import DecisionTreeClassifier
model_gini1 = DecisionTreeClassifier(criterion='gini', max_depth=2)
model_gini1.fit(x_train,y_train)
```

Out[32]: `DecisionTreeClassifier(max_depth=2)`

In [33]:
```
#Predicting on test data
pred_testg1 = model_gini1.predict(x_test)
#Accuracy on test data
print('Test data Accuracy is:',np.mean(pred_testg1==y_test))
#Predicting on train data
pred_traing1 = model_gini1.predict(x_train)
#Accuracy on train data
print('Train data Accuracy is:',np.mean(pred_traing1==y_train))
```

```
Test data Accuracy is: 0.7777777777777778
Train data Accuracy is: 0.8
```

Iteration-2: Max Depth = 3

In [34]:
```
model_gini2 = DecisionTreeClassifier(criterion='gini', max_depth=3)
model_gini2.fit(x_train,y_train)
```

Out[34]: `DecisionTreeClassifier(max_depth=3)`

In [35]:
```
#Predicting on test data
pred_testg2 = model_gini2.predict(x_test)
#Accuracy on test data
print('Test data Accuracy is:',np.mean(pred_testg2==y_test))
#Predicting on train data
pred_traing2 = model_gini2.predict(x_train)
#Accuracy on train data
print('Train data Accuracy is:',np.mean(pred_traing2==y_train))
```

```
Test data Accuracy is: 0.7777777777777778
Train data Accuracy is: 0.8
```

In [36]:
```
model_gini3 = DecisionTreeClassifier(criterion='gini', max_depth=4)
model_gini3.fit(x_train,y_train)
```

Out[36]: `DecisionTreeClassifier(max_depth=4)`

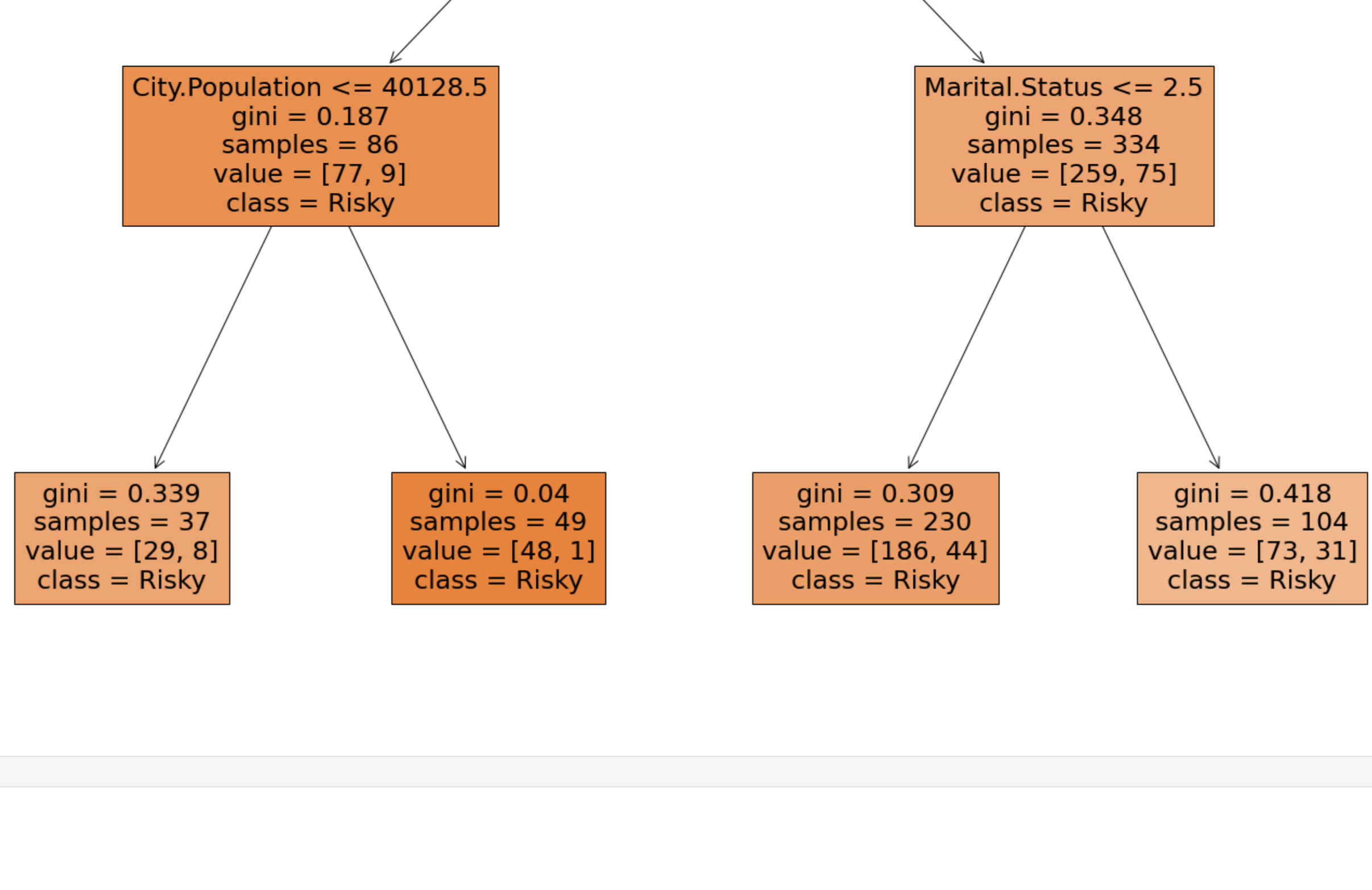
In [37]:
```
#Predicting on test data
pred_testg3 = model_gini3.predict(x_test)
#Accuracy on test data
print('Test data Accuracy is:',np.mean(pred_testg3==y_test))
#Predicting on train data
pred_traing3 = model_gini3.predict(x_train)
#Accuracy on train data
print('Train data Accuracy is:',np.mean(pred_traing3==y_train))
```

```
Test data Accuracy is: 0.7666666666666667
Train data Accuracy is: 0.8142857142857143
```

We get the best test results at Iteration-1 max depth = 2. so we will consider that final

In [38]:
```
# let's plot the decision tree
fig = plt.figure(figsize=(25,20))
fig = tree.plot_tree(model_gini1,
 feature_names= ['Undergrad','Marital.Status','City.Population','Work.Experience'], class_names= ['Risky', 'Good'], filled=True)
plt.title('Decision tree using Gini',fontsize=22)
```

Out[38]: `Text(0.5, 1.0, 'Decision tree using Gini')`



In [ ]: