Tobacco_Mortality_Project

May 12, 2025

[1]:
```
import pandas as pd

# Load all datasets
df_admissions = pd.read_csv("admissions.csv")
df_fatalities = pd.read_csv("fatalities.csv")
df_metrics = pd.read_csv("metrics.csv")
df_prescriptions = pd.read_csv("prescriptions.csv")
df_smokers = pd.read_csv("smokers.csv")

# Preview one of them
df_admissions.head()
```

[1]:

	Year	ICD10 Code	ICD10 Diagnosis	Diagnosis Type	Metric	Sex	Value
0	2014/15	All codes	All admissions	All admissions	Number of admissions	NaN	11011882
1	2014/15	C33-C34 & C00-C14 & C15 & C32 & C53 & C67 & C6…	All diseases which can be caused by smoking	All diseases which can be caused by smoking	Number of admissions	NaN	1713330
2	2014/15	C00-D48	All cancers	All cancers	Number of admissions	NaN	1691035
3	2014/15	J00-J99	All respiratory diseases	All respiratory diseases	Number of admissions	NaN	611002
4	2014/15	I00-I99	All circulatory diseases	All circulatory diseases	Number of admissions	NaN	

```
4    907157
```

[3]:
```
df_fatalities.head()  # Preview fatalities
```

[3]:
```
   Year                                          ICD10 Code  \
0  2014                                           All codes
1  2014  C33-C34 & C00-C14 & C15 & C32 & C53 & C67 & C6…
2  2014                                             C00-D48
3  2014                                             J00-J99
4  2014                                             I00-I99

                            ICD10 Diagnosis  \
0                                All deaths
1  All deaths which can be caused by smoking
2                               All cancers
3                 All respiratory diseases
4                 All circulatory diseases

                            Diagnosis Type                      Metric  Sex  \
0                                All deaths  Number of observed deaths  NaN
1  All deaths which can be caused by smoking  Number of observed deaths  NaN
2                               All cancers  Number of observed deaths  NaN
3                 All respiratory diseases  Number of observed deaths  NaN
4                 All circulatory diseases  Number of observed deaths  NaN

    Value
0  459087
1  235820
2  136312
3   61744
4  126101
```

[5]:
```
df_smokers.head()  # Preview smokers
```

[5]:

	Year	Method	Sex	16 and Over	16-24	25-34	35-49	50-59	60 and Over
0	1974	Unweighted	NaN	46	44	51	52	50	33
1	1976	Unweighted	NaN	42	42	45	48	48	30
2	1978	Unweighted	NaN	40	39	45	45	45	30
3	1980	Unweighted	NaN	39	37	46	44	45	29
4	1982	Unweighted	NaN	35	35	38	39	41	27

[7]:
```
df_deaths = df_fatalities[df_fatalities['ICD10 Diagnosis'].str.contains('caused␣
 ↪by smoking', case=False, na=False)]  # Only keep rows for smoking-related␣
 ↪deaths

df_deaths = df_deaths[['Year', 'Value']].rename(columns={'Value': 'Deaths'})  #␣
 ↪Drop unnecessary columns
```

```
df_deaths.head()
```

[7]:
```
      Year  Deaths
1     2014  235820
54    2014  123135
107   2014  112685
160   2013  241683
213   2013  124504
```

[9]:
```
df_smoking = df_smokers.drop(columns=['Method', 'Sex'])  # Drop unnecessary columns

df_smoking['SmokingRate'] = df_smoking.iloc[:, 1:].mean(axis=1)  # Group by year and calculate average smoking rate

df_smoking = df_smoking[['Year', 'SmokingRate']]  # Keep only Year and SmokingRate

df_smoking.head()
```

[9]:
```
   Year  SmokingRate
0  1974    46.000000
1  1976    42.500000
2  1978    40.666667
3  1980    40.000000
4  1982    35.833333
```

[11]:
```
df_deaths['Year'] = df_deaths['Year'].astype(int)  # Convert 'Year' to int for both before merging
df_smoking['Year'] = df_smoking['Year'].astype(int)

df_model = pd.merge(df_deaths, df_smoking, on='Year')  # Merge on Year

df_model.head()
```

[11]:
```
   Year  Deaths  SmokingRate
0  2014  235820    19.500000
1  2014  235820    20.833333
2  2014  235820    18.166667
3  2014  123135    19.500000
4  2014  123135    20.833333
```

[13]:
```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score
```

```
X = df_model[['SmokingRate']]  # Split features and target
y = df_model['Deaths']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,␣
↪random_state=42)  # Train-test split

model = LinearRegression()  # Model
model.fit(X_train, y_train)

y_pred = model.predict(X_test)  # Predict

mse = mean_squared_error(y_test, y_pred)  # Evaluation
r2 = r2_score(y_test, y_pred)

print("Mean Squared Error:", mse)
print("R² Score:", r2)
```

```
Mean Squared Error: 3469303566.261036
R² Score: -0.08055108669904687
```

[ ]:
```
##  Final Thoughts

- The goal was to predict mortality caused by smoking using smoking rate as the␣
↪main feature.
- I cleaned and merged historical data from two sources: observed deaths and␣
↪smoking trends.
- A Linear Regression model was trained to explore the relationship.

###  Key Metrics:
- **Mean Squared Error:** 3.47 billion
- **R² Score:** -0.08

###  Conclusion:
The model did not perform well – likely due to:
- Small dataset (few years)
- Using only one simple feature (SmokingRate)

Still, this was a valuable experience in merging real datasets, preparing␣
↪features, and evaluating models. More features and more data could␣
↪significantly improve the result.
```

[15]:
```
df_metrics.head()
df_prescriptions.head()
df_admissions.head()
```

[15]:
```
      Year                                          ICD10 Code  \
0  2014/15                                           All codes
```

```
1  2014/15  C33-C34 & C00-C14 & C15 & C32 & C53 & C67 & C6…
2  2014/15                                              C00-D48
3  2014/15                                              J00-J99
4  2014/15                                              I00-I99

                              ICD10 Diagnosis  \
0                              All admissions
1  All diseases which can be caused by smoking
2                                  All cancers
3                    All respiratory diseases
4                    All circulatory diseases

                              Diagnosis Type                Metric  Sex  \
0                              All admissions  Number of admissions  NaN
1  All diseases which can be caused by smoking  Number of admissions  NaN
2                                  All cancers  Number of admissions  NaN
3                    All respiratory diseases  Number of admissions  NaN
4                    All circulatory diseases  Number of admissions  NaN

      Value
0  11011882
1   1713330
2   1691035
3    611002
4    907157
```

[17]:
```
df_admit = df_admissions[df_admissions['ICD10 Diagnosis'].str.contains('caused by smoking', case=False, na=False)]  # Filter only smoking-related admissions

df_admit = df_admit[['Year', 'Value']].rename(columns={'Value': 'Admissions'}) # Keep Year and Value, rename for clarity

df_admit['Year'] = df_admit['Year'].str[:4].astype(int)  # Convert year format from '2014/15' to 2014

df_model = pd.merge(df_model, df_admit, on='Year')  # Merge into your main df_model

df_model.head()  # Check new structure
```

[17]:
```
   Year  Deaths  SmokingRate Admissions
0  2014  235820    19.500000    1713330
1  2014  235820    19.500000     931001
2  2014  235820    19.500000     782329
3  2014  235820    20.833333    1713330
4  2014  235820    20.833333     931001
```

[19]:
```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score

X = df_model[['SmokingRate', 'Admissions']]  # Use both features now
y = df_model['Deaths']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,␣
↪random_state=42)  # Split

model = LinearRegression()  # Train
model.fit(X_train, y_train)

y_pred = model.predict(X_test)  # Predict

mse = mean_squared_error(y_test, y_pred)  # Evaluate
r2 = r2_score(y_test, y_pred)

print("Mean Squared Error:", mse)
print("R² Score:", r2)
```

```
Mean Squared Error: 3898577318.8656363
R² Score: 0.010465077041139859
```

[21]:
```
df_prescriptions.head()
```

[21]:

	Year	All Pharmacotherapy Prescriptions	Nicotine Replacement Therapy (NRT) Prescriptions	Bupropion (Zyban) Prescriptions	Varenicline (Champix) Prescriptions	Net Ingredient Cost of All Pharmacotherapies
0	2014/15	1348	766	21	561.0	
1	2013/14	1778	1059	22	697.0	
2	2012/13	2203	1318	26	859.0	
3	2011/12	2532	1545	30	957.0	
4	2010/11	2564	1541	36	987.0	

```
0                                          38145
1                                          48767
2                                          58121
3                                          64552
4                                          65883

   Net Ingredient Cost of Nicotine Replacement Therapies (NRT)  \
0                                                      18208
1                                                      24257
2                                                      28069
3                                                      30951
4                                                      30808

   Net Ingredient Cost of Bupropion (Zyban)  \
0                                       807
1                                       865
2                                       994
3                                      1216
4                                      1581

   Net Ingredient Cost of Varenicline (Champix)
0                                       19129.0
1                                       23646.0
2                                       29058.0
3                                       32385.0
4                                       33494.0
```

[23]:
```
df_prescriptions.columns
```

[23]:
```
Index(['Year', 'All Pharmacotherapy Prescriptions',
       'Nicotine Replacement Therapy (NRT) Prescriptions',
       'Bupropion (Zyban) Prescriptions',
       'Varenicline (Champix) Prescriptions',
       'Net Ingredient Cost of All Pharmacotherapies',
       'Net Ingredient Cost of Nicotine Replacement Therapies (NRT)',
       'Net Ingredient Cost of Bupropion (Zyban)',
       'Net Ingredient Cost of Varenicline (Champix)'],
      dtype='object')
```

[25]:
```
df_rx = df_prescriptions[['Year', 'All Pharmacotherapy Prescriptions']].copy() ␣
 ↪# Extract relevant columns
df_rx = df_rx.rename(columns={'All Pharmacotherapy Prescriptions':␣
 ↪'Prescriptions'})

df_rx['Year'] = df_rx['Year'].str[:4].astype(int)  # Convert Year from '2014/
 ↪15' to 2014
```

```
df_model = pd.merge(df_model, df_rx, on='Year')  # Merge into main model␣
 ↪dataframe

df_model.head()  # Preview updated df_model
```

[25]:

	Year	Deaths	SmokingRate	Admissions	Prescriptions
0	2014	235820	19.500000	1713330	1348
1	2014	235820	19.500000	931001	1348
2	2014	235820	19.500000	782329	1348
3	2014	235820	20.833333	1713330	1348
4	2014	235820	20.833333	931001	1348

[27]:
```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score

X = df_model[['SmokingRate', 'Admissions', 'Prescriptions']]  # Now use all␣
 ↪three features
y = df_model['Deaths']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,␣
 ↪random_state=42)  # Train-test split

model = LinearRegression()  # Model training
model.fit(X_train, y_train)

y_pred = model.predict(X_test)  # Prediction

mse = mean_squared_error(y_test, y_pred)  # Evaluation
r2 = r2_score(y_test, y_pred)

print("Mean Squared Error:", mse)
print("R² Score:", r2)
```

```
Mean Squared Error: 3901880937.2253637
R² Score: 0.009626554300226298
```

[ ]:

[ ]: