# Predicting Health Insurance Charges: A Data Science Approach

## Introduction

### Introduce the problem

Estimating premiums that are fair and accurate in the health insurance industry is an uncontestable difficulty. Insurers must assess what the healthcare cost will be before the insured gets any services, and typically, limited information is available for them to make that assessment. This creates premiums that do not accurately reflect an individual's true risk profile, resulting in lost profits for insurers and unjust pricing for customers. The uncertainty of being able to know what the charges will be can negatively impact profit margins, competitiveness in the market, and satisfaction levels from customers. This project aims to rectify that issue by implementing machine learning methods to model and predict the costs of health insurance charges based on generally relevant variables such as age, BMI, and smoking status.

### Justify Why It Is Important/Useful to Solve This Problem

Addressing this issue has important financial and operational consequences for health insurers. Well-formed charge predictions help craft fair premiums, limit adverse selection, and foster customer retention. With well-formed models, underwriting can improve the identification of high-risk applicants early on and engage in deliberate healthcare intervention. Approximately 20% of health insurance claims account for 80% of costs; the rise in overall healthcare costs, coupled with increased customer demand for insight and transparency, may create additional competitive advantages. Accordingly, data-driven pricing models enable insurers to address pricing within the context of the regulatory framework, which typically requires some evidence to be defensible (e.g., fair and evidence-based). Furthermore, by having predictive models help with estimating a data-based price, an insurer is improving the quality of that data-driven price, the marginal benefit to its customers, and its bottom line. Successfully embedding predictive models for estimating prices into

its pricing functions can improve customer trust and support the insurer's risk management and pursuit of sustainable growth.

**How would you pitch this problem to a group of stakeholders to gain buy-in to proceed?**

Think about being able to price every policyholder's insurance premium, importing only a few data points with a high level of confidence. That is this analysis's value proposition. Novel predictive modeling techniques can materially remove the uncertainty in our pricing decisions. This means risk mitigation, more competitive offerings, and ultimately, satisfied customers. This is an opportunity for stakeholders to align their business strategy with innovation. Our models do not subtract from actuarial models; they add to them utilizing data science. By investing in this now, we will be ready for the day when pricing precision and transparency are not optional.

**Explain where you obtained your data**

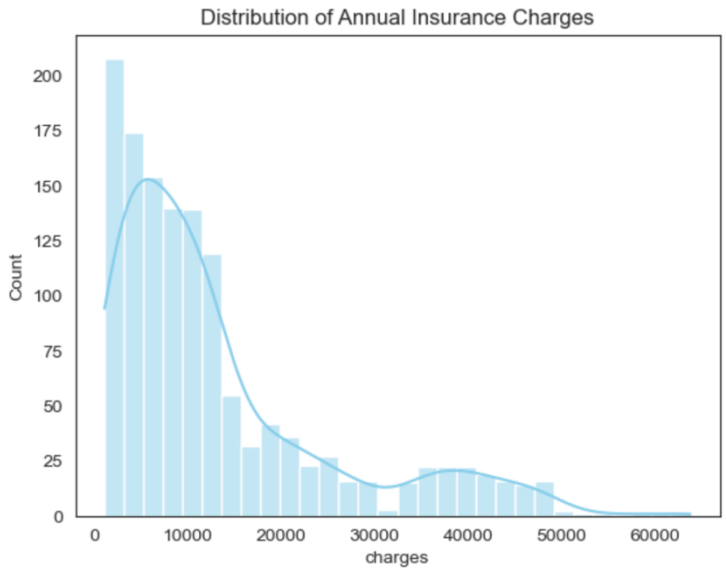
This analysis uses a publicly available Kaggle dataset that is commonly used in academic and industry benchmarking. The dataset contains anonymized health and demographic data for over 1,300 subjects with features such as age, BMI, smoking status, and total medical costs. Although this dataset is synthetic, it reflects insurance coverage from typical health scenarios, producing insurance situations, making this dataset appropriate to drive out prototyping predictive models. The structure and quality of the dataset suggest we can explore relationships between variables and understand perceived worth in terms of insurance cost. This dataset is clean and trustworthy in that we can model off it relatively safely and then deal with precision on the pay-for dataset.

**Exploratory Data Analysis (EDA)**

The exploratory data analysis revealed strong relationships between insurance charges and factors like smoking, age, and BMI. The below visualizations confirmed data quality and highlighted key trends for predictive modeling.

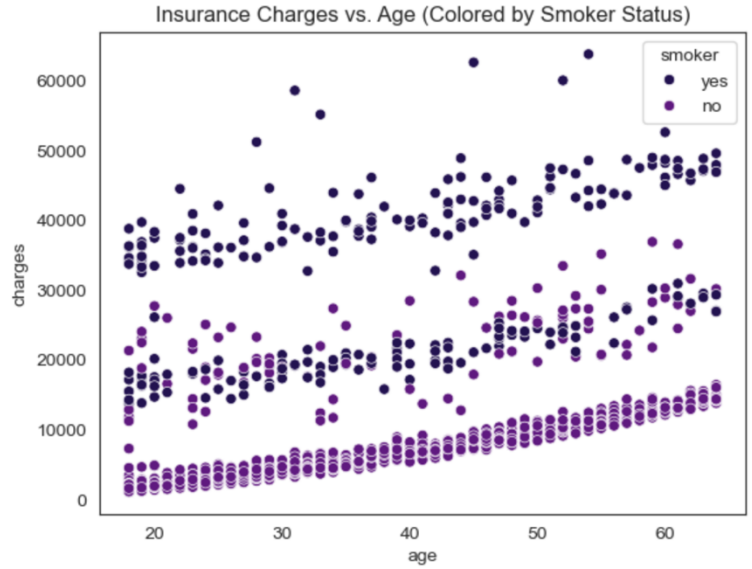
1. Distribution of Insurance Charges

```
sns.histplot(data=healthcare_insurance_df, x='charges', kde=True, color='skyblue')
plt.title('Distribution of Annual Insurance Charges')
plt.show()
```




2. Charges vs. Age by Smoker Status

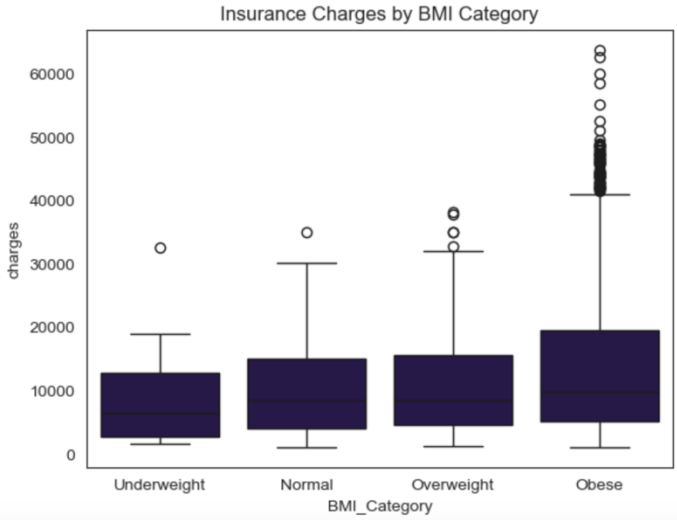
```
sns.scatterplot(data=healthcare_insurance_df, x='age', y='charges', hue='smoker')
plt.title('Insurance Charges vs. Age (Colored by Smoker Status)')
plt.show()
```



3. Charges by BMI Category

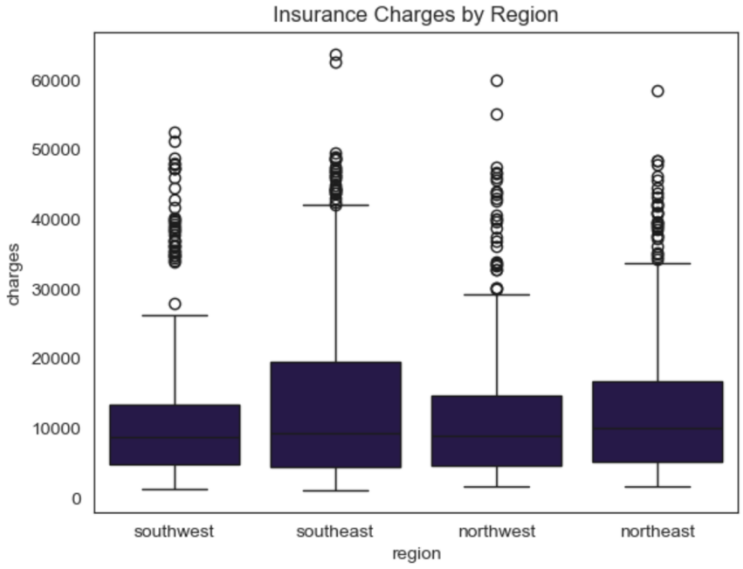
```
healthcare_insurance_df['BMI_Category'] = pd.cut(healthcare_insurance_df['bmi'], bins=[0, 18.5, 25, 30, 100],
                            labels=['Underweight', 'Normal', 'Overweight', 'Obese'])

sns.boxplot(data=healthcare_insurance_df, x='BMI_Category', y='charges')
plt.title('Insurance Charges by BMI Category')
plt.show()
```




4. Charges by Region

```
sns.boxplot(data=healthcare_insurance_df, x='region', y='charges')
plt.title('Insurance Charges by Region')
plt.show()
```

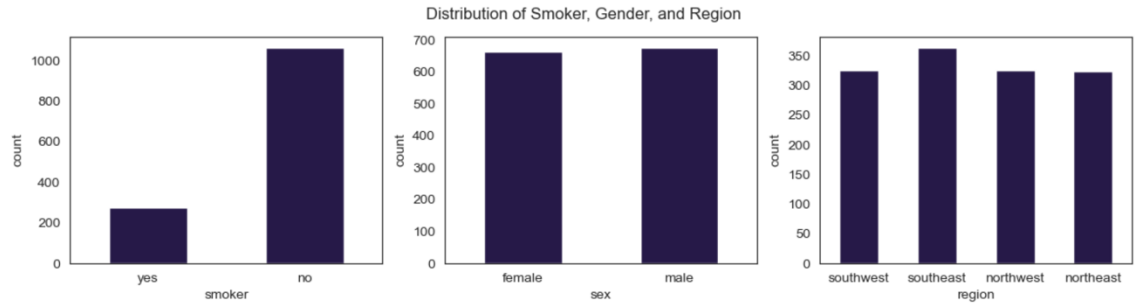


5. Distribution of Smoker, Gender, and Region

```
f, (ax_1, ax_2, ax_3) = plt.subplots(1, 3, figsize=(14,3))

sns.countplot(healthcare_insurance_df,x='smoker', ax=ax_1, width=0.5)
sns.countplot(healthcare_insurance_df,x='sex', ax=ax_2, width=0.5)
sns.countplot(healthcare_insurance_df,x='region', ax=ax_3, width=0.5)

plt.suptitle('Distribution of Smoker, Gender, and Region', fontsize = 12)
plt.show()
```




**Data preparation**

1. After inspecting the dataset, features—age, sex, bmi, children, smoker, and region — each have a straightforward and plausible connection to changes in the target variable, charges.

   For example,

   - Age and BMI are continuous measures of health risk, while the smoker status is a direct driver of health costs.
   - The region may even capture pricing differentials because of geography.
   - In fact, the number of children may influence the insurance coverage required and the costs associated with it.

   As a result, no features will be eliminated at this time, because all the variables have the potential to be predictive and relevant for building models of insurance charges.

2. At this point, we have not done any row-level filtering since each observation corresponds to an insured record and the records represent the entirety of valuable detail that can be used to forecast insurance charges; by keeping all of the records, we have all of the variation across all the demographic and health descriptors, which is important for building a solid and generalizable model.

3. Transforming features for model building

```
# Convert categorical features to numeric using one-hot encoding.
categorical_columns = ['sex', 'smoker', 'region', 'BMI_Category']
healthcare_insurance_df_encoded = pd.get_dummies(healthcare_insurance_df, columns=categorical_columns, drop_first=True)
```

```
#print the first 5 rows on the encoded dataset
healthcare_insurance_df_encoded.head()
```

| | age | bmi | children | charges | sex_male | smoker_yes | region_northwest | region_southeast | region_southwest | BMI_Category_Normal |
|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 19 | 27.900 | 0 | 16884.92400 | False | True | False | False | True | False |
| **1** | 18 | 33.770 | 1 | 1725.55230 | True | False | False | True | False | False |
| **2** | 28 | 33.000 | 3 | 4449.46200 | True | False | False | True | False | False |
| **3** | 33 | 22.705 | 0 | 21984.47061 | True | False | True | False | False | True |
| **4** | 32 | 28.880 | 0 | 3866.85520 | True | False | True | False | False | False |

For the analysis to take full advantage of machine learning algorithms, any categorical variable (like sex, smoker, region) had to be turned into dummy (one-hot encoded) variables so that these categorical features could be easily interpreted by models that use numbers to represent variables. When using dummy variables, the information contained in category memberships remains intact without introducing arbitrary ordinal relationships between the categorical values.

4. Throughout our feature transformation, we previously one-hot encoded categorical variables in the form of dummy variables (i.e., sex, smoker, region, age group, BMI_Category), to keep the data in a format that machine learning can interpret.

   Thus, there is no need to create more dummy variables now, and the data is ready to be used in model building.

```
# Convert the True/False columns to 1/0
healthcare_insurance_df_encoded = healthcare_insurance_df_encoded.astype(int)

# Print the first 5 rows of the cleaned dataset
healthcare_insurance_df_encoded.head()
```

| | age | bmi | children | charges | sex_male | smoker_yes | region_northwest | region_southeast | region_southwest | BMI_Category_Normal | E |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 19 | 27 | 0 | 16884 | 0 | 1 | 0 | 0 | 1 | 0 | |
| **1** | 18 | 33 | 1 | 1725 | 1 | 0 | 0 | 1 | 0 | 0 | |
| **2** | 28 | 33 | 3 | 4449 | 1 | 0 | 0 | 1 | 0 | 0 | |
| **3** | 33 | 22 | 0 | 21984 | 1 | 0 | 1 | 0 | 0 | 1 | |
| **4** | 32 | 28 | 0 | 3866 | 1 | 0 | 1 | 0 | 0 | 0 | |

**Model building and evaluation**

The objective of this project is to predict healthcare insurance charges, a continuous outcome. Therefore, regression models are most suitable

**Feature Selection**

The three features used — smoker_yes, age, and bmi — were selected based on domain knowledge and correlation with the target variable (charges):

- smoker_yes: Strongest predictor; smokers have significantly higher charges.

- age: Health risks and medical costs tend to increase with age.

- bmi: Higher BMI is linked to obesity-related health issues, which increase insurance costs.

- **Linear Regression Model:**

```
# Define independent and dependent variables
x = healthcare_insurance_df_encoded[['smoker_yes', 'age', 'bmi']].values
y = np.array(healthcare_insurance_df_encoded['charges'])

# Create training and test sets
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.1, random_state=0)
```

```
# Train a linear regression model
from sklearn.linear_model import LinearRegression
model_MLR = LinearRegression().fit(x_train, y_train)

# Predict on training and test sets
y_train_pred = model_MLR.predict(x_train)
y_test_pred = model_MLR.predict(x_test)

# Evaluate using R²
from sklearn.metrics import r2_score, mean_squared_error, mean_absolute_error

r2_train = r2_score(y_train, y_train_pred)
r2_test = r2_score(y_test, y_test_pred)
rmse_test = mean_squared_error(y_test, y_test_pred, squared=False)
mae_test = mean_absolute_error(y_test, y_test_pred)

print(f"R² (Train): {r2_train:.4f}")
print(f"R² (Test): {r2_test:.4f}")
print(f"RMSE (Test): {rmse_test:.2f}")
print(f"MAE (Test): {mae_test:.2f}")
print(f"Intercept: {model_MLR.intercept_}")
print(f"Coefficients: {model_MLR.coef_}")
```

```
R² (Train): 0.7350
R² (Test): 0.8188
RMSE (Test): 5927.89
MAE (Test): 4263.21
Intercept: -10921.57014439852
Coefficients: [23505.86253837   256.66657492   306.85949071]
```

- **Quadratic Polynomial Regression**

```python
# Quadratic Polynomial Regression
from sklearn.preprocessing import PolynomialFeatures
from sklearn.linear_model import LinearRegression
from sklearn.metrics import r2_score, mean_squared_error, mean_absolute_error

# Generate polynomial features (degree 2)
poly = PolynomialFeatures(degree=2, include_bias=False)
x_train_poly = poly.fit_transform(x_train)
x_test_poly = poly.transform(x_test)

# Fit the model
model_PR = LinearRegression().fit(x_train_poly, y_train)

# Predictions
y_train_pred_poly = model_PR.predict(x_train_poly)
y_test_pred_poly = model_PR.predict(x_test_poly)

# Evaluation
r2_train_poly = r2_score(y_train, y_train_pred_poly)
r2_test_poly = r2_score(y_test, y_test_pred_poly)
rmse_test_poly = mean_squared_error(y_test, y_test_pred_poly, squared=False)
mae_test_poly = mean_absolute_error(y_test, y_test_pred_poly)

# Output
print(f"R² (Train): {r2_train_poly:.4f}")
print(f"R² (Test): {r2_test_poly:.4f}")
print(f"RMSE (Test): {rmse_test_poly:.2f}")
print(f"MAE (Test): {mae_test_poly:.2f}")
print(f"Intercept: {model_PR.intercept_}")
print(f"Coefficients: {model_PR.coef_}")
```

```
R² (Train): 0.8292
R² (Test): 0.8970
RMSE (Test): 4468.04
MAE (Test): 2927.23
Intercept: -4047.9709688815565
Coefficients: [-9.79594794e+03  4.18637251e+01  4.04860376e+02 -9.79594794e+03
  2.97522468e+00  1.43470002e+03  1.89608200e+00  2.48047605e+00
 -8.07747969e+00]
```

- **Cubic Polynomial Regression**

```python
# Cubic Polynomial Regression
from sklearn.preprocessing import PolynomialFeatures
from sklearn.linear_model import LinearRegression
from sklearn.metrics import r2_score, mean_squared_error, mean_absolute_error

# Generate polynomial features (degree 3)
poly3 = PolynomialFeatures(degree=3, include_bias=False)
x_train_poly3 = poly3.fit_transform(x_train)
x_test_poly3 = poly3.transform(x_test)

# Fit the model
model_PR3 = LinearRegression().fit(x_train_poly3, y_train)

# Predictions
y_train_pred_poly3 = model_PR3.predict(x_train_poly3)
y_test_pred_poly3 = model_PR3.predict(x_test_poly3)

# Evaluation
r2_train_poly3 = r2_score(y_train, y_train_pred_poly3)
r2_test_poly3 = r2_score(y_test, y_test_pred_poly3)
rmse_test_poly3 = mean_squared_error(y_test, y_test_pred_poly3, squared=False)
mae_test_poly3 = mean_absolute_error(y_test, y_test_pred_poly3)

# Output
print(f"R² (Train): {r2_train_poly3:.4f}")
print(f"R² (Test): {r2_test_poly3:.4f}")
print(f"RMSE (Test): {rmse_test_poly3:.2f}")
print(f"MAE (Test): {mae_test_poly3:.2f}")
print(f"Intercept: {model_PR3.intercept_}")
print(f"Coefficients: {model_PR3.coef_}")
```

```
R² (Train): 0.8319
R² (Test): 0.8965
RMSE (Test): 4480.78
MAE (Test): 2955.88
Intercept: 22057.12732227883
Coefficients: [-6.62007236e+03  2.39758815e+02 -2.49151640e+03 -6.62007236e+03
 -2.96830514e+01  7.70321964e+02 -7.45537666e-01 -2.02374890e+00
  8.78001624e+01 -6.62007236e+03 -2.96830513e+01  7.70321964e+02
 -2.66220071e+00  8.95223974e+00 -6.85915828e+00  8.06967957e-03
  7.22461233e-02 -6.90969680e-02 -9.47140610e-01]
```

## Model Overview and Insights

To predict healthcare insurance costs, we created and assessed three regression models in this project using a subset of important characteristics, including age, BMI, and smoker_yes. Investigating both linear and nonlinear relationships between these predictors and insurance costs, evaluating model performance, and extracting useful business insights were the objectives.

**1. Linear Regression makes for a strong and interpretable baseline.**

- smoker_yes is overwhelmingly the biggest contributor, increasing predicted charges by about $23,506.
- age and bmi also have moderate, positive effects.

Further, the test $R^2$ of about 0.82 indicates that this simple model represents most of the signal.

**2. Polynomial models fit additional nonlinear trends.**

- The quadratic model's diagnostics show it significantly improves training $R^2$, has a lower test error (RMSE and MAE), and captures some curvature and interactions that the linear regression misses.
- The cubic model, however, achieves even more training accuracy but performs worse than the quadratic model on the test set, indicating overfitting.

**3. The relationship between model complexity and generalization must be established.**

- Adding complexity beyond a quadratic model did not improve accuracy in a meaningful way and hurt generalization.
- The quadratic model provided the best balance between accuracy and stability.

## Conclusion

The analysis examined all three linear, quadratic, and cubic regression models for predicting healthcare insurance charges using the critical predictors smoking status, age, and BMI. While all models performed reasonably well, the quadratic model achieved the best compromise between

predictive accuracy and generalization ability. The non-linear relationships that remained important for predictive accuracy were contained within the quadratic model, as the cubic model was overfitting. As a whole, the results support the importance of smoking as the most impactful predictive factor found, and support the use of a quadratic model as precise, reliable, and interpretable to inform cost prediction in insurance decision-making.