**Problem statement**

Profitability for oil & gas companies are largely driven by commodity prices that are set by the market. These can depend on various factors such as supply and demand (both local and international), weather conditions, seasonal demand, political unrest in the Middle East etc. Weather is actually a big factor for natural gas demand because daily weather has a significant impact on power plants for electricity generation. Over the last few years, natural gas has replaced coal to become the leading source for power generation in the country. It would be very beneficial to natural gas producers if they could predict spot prices based on weather patterns and accordingly scale production back or up to not oversupply the market. The goal of this project is to build a model that would predict the effect of temperature data on natural gas spot prices at the Henry Hub natural gas distribution hub in Erath, Louisiana. Henry Hub is the leading distribution hub for natural gas in the US.

**Methodology**

**Data Wrangling:**

The dependent variable is natural gas spot prices and the independent variables are daily average temperature (in Celsius), daily average pressure (in inches of mercury) and daily average relative humidity (percentage). The two datasets used for this project are public and accessible to the general population using Web APIs. The EIA government website provides historical data for daily natural gas spot prices. Weather Underground provides daily weather data for Erath, LA. Let us look at each dataset in more detail.

**EIA:** Using an EIA provided API key, this data was loaded into a .csv file for analysis. It is a fairly simple dataset with just 2 fields, the date in yyymmdd format and the spot price for that day. There was data all the way from 1997 through current date. This turned out to be very clean and nicely formatted and did not need any data wrangling except for filtering the data down to our dates of interest

**Weather Underground:** A similar API call was obtained from Weather Underground to extract weather data. The dataset had several weather related variables and the fields of interest were narrowed down to temperature, humidity and pressure. There was a fair amount of data wrangling done to get this dataset into the correct format. Some of the readings showed a value of -9999 which one assumes was just a placeholder for a reading not recorded or an incorrect reading. These rows were dropped from the dataset. The date field was of the text datatype and it had to be converted to an integer in the yyyymmdd format to match the EIA data. Additionally, the dataset had readings for every 20 minutes so the data had to be averaged to report at the day level. Each API call would extract data for one day and it was written to an individual csv file. A for loop in R was used to increment the API syntax to pass the following day’s date for the subsequent call. All the files were combined into one dataset prior to the data wrangling exercise.

Additionally, Weather Underground has restrictions on how much data one could extract within the scope of a free account. Only 500 days of history could be extracted each day, so multiple runs over different date ranges had to be made to fill out the training and test datasets. Another restriction was on the number of calls per minute so a pause had to be built in. To generate the training set, it was decided to use the dataset for the most recent two full years and that was 2015 and 2016. After the necessary data wrangling was done on the weather dataset, the two datasets were combined keyed on date and the resulting dataset was written out to a csv file to generate the training set data.

**Machine Learning:**

Since we are looking to predict spot prices over a period of time, our plan of action was to build different linear regression models to settle on the best combination of independent variables. The best model turned out to be the one where all three independent variables were used. A test dataset for 2017 was generated to test the model and predictions for spot prices were run over this.

Different techniques were tried to improve the accuracy of our model. We looked at the impact of adding new variables which would be a combination of two or more of the existing variables based on their correlation. We looked at feature transformation using centering and scaling techniques to normalize the data. Additionally, histograms and boxplots were created to look at data distribution and locate outliers to exclude from the model.

As a last step, in addition to linear regression, a regression tree model was created to compare and contrast between different model types. Tree pruning techniques were used to improve the predictive abilities of the model.

**Results**

The standard regression measures of R-squared, adjusted R-squared, RMSE, SSE, F-statistic and p-value were used to measure the strength of the models.

The best linear regression model only had an adjusted R-squared value of 10% and a SSE of 100. A likely explanation for this is that we are missing additional key data points. Though weather is indeed a big factor that goes into the determination of commodity prices, there is another equally important factor that plays a role. Storage levels of natural gas drives supply and demand and we’re not taking that into account for this exercise. Also, we are only looking at weather readings for the Henry Hub location, it would make the model more robust if we picked weather readings across the country and averaged them out.

The regression tree model actually had a lower SSE of 75 and is a worthy candidate to use in conjunction with the linear model.

**Recommendations**

Based on our findings, here are a few recommendations for a natural gas producer.

1. Use the linear regression and regression tree models together when running predictions.
2. Improve the quality of your data by excluding outliers and filling in missing values before running predictions.
3. Due to changing weather patterns, keep improving the model by incorporating more recent weather data into the training set.