# The Problem

* Thousands of potential customers visit our website every day for a free horoscope report, some of which actually result in a conversion.
* Due to limited human resources, we are unable to reach out to each one of those thousands of potential customers each day. In addition to being infeasible for us, it is probably not necessary either.
* To help filter the long list into something manageable by the sales team, we came up with a baseline model that prioritizes the customers we reach out to each day. The baseline model was built in a hurry without any serious data analysis and it is just a static formula taking as input certain values generated from browsing sessions.
* Over the years, we have found that a majority of the potential customers we reach out to do not result in an immediate conversion.
* We want to use data and technology to maximize conversions from our contacts each day.

# The Expected Solution

* We want to build a Machine Learning system which prioritizes the potential list of customers based on the patterns found in historical data. We already have a baseline model in place.
* The solution should consist of two parts, or reports to be more specific: -
  + For the sales team: A report containing the predictions for the day in increasing order of their conversion probabilities. The number of potential customers should be restricted to the top 250.
  + For the business managers: A reports containing the Actual conversions vs the Predicted conversion along with some KPI’s like Recall score, conversion ratio etc.

# The Data

* **Sessions table**: - This dataset had data points which are collected when a customer visits our website. Here, we have data points like where are the clicks a customer made, what kind of products they viewed, did they attempt making a purchase, the log in time etc.
* **Beacons table**: - It is linked to the Sessions table and is a like a heartbeat for the sessions table.
* **Customers table: -** This dataset contains the personal information of all the customers that have visited our website in a time frame of 3 months.
* **Transactions table: -** The transaction table contains the details of the transaction made by any customer. This is the table which contains our target variable and tells whether a customer got converted or not.
* **Products table: -** This table is again linked to the transactions table and contains a list of all types of products and reports that a customer has purchased again in a time frame of 3 months.

# Development Process and Model Evaluation

* The first step as is with any ML project was to analyze the data, have a clear picture of it and find connections between different tables to merge them into a single dataset.
* After the merging was done, we explored the data further to remove any potential outliers that could disrupt the pattern recognition. The final dataset that we obtained after pre-processing steps was highly imbalanced, meaning we had more non-conversions than we had conversions so to adjust for that, we needed some kind of resampling. We went with an under-sampling technique called Tomek Link which made the dataset fairy balanced.
* Our dataset was prepared and we moved to the next stage of experimenting with various machine learning algorithms. Since, we wanted probabilities in our end result, we chose algorithms such as Logistic Regression and Stochastic Gradient Descent classifier. For making more solid comparisons, we also experimented a little with Gradient Boosting algorithm.
* Before we started fitting algorithms, out of the 16 features that were available in our final merged consolidated dataset, we created 5 subsets of features and these subsets were chosen based on Correlation Analysis, feature importance given by the algorithms.
* To add another layer of experimentation, we also scaled every feature set using two different techniques: Standard Scaling and Max Absolute Scaling.
* Our primary metric chosen for evaluation was Balanced Accuracy and secondary metric is Recall.
* After experimenting with many combinations of feature sets and taking into account the balanced accuracy and recall scores threshold given by the domain expert, we found that Stochastic Gradient Descent was the best performing model among all.
* Next, we tuned the models that we obtained using Grid Search and cross validated them again. The following slide shows the top 5 models that we obtained after rigorous experimentation.

# High Level Architecture

* The current slide shows the High-Level Architecture of out solution design. So, the solution is a bundled combination of 4 specific modules.
* **Data Extraction and preparation module**: - The purpose of this module is to extract and provide prepared data to other modules. It accepts the consolidated merged dataset prepares it for the requested date and passes the output to the other three modules.
* **Incremental Training module**: - So, we will generate new data everyday after the sales team makes calls and tries to convert potential customers. This module is prepared with the objective of feeding our model data from 3 days ago to help it learn any new patterns it can find. It accepts the selected model and data from three days ago, does a partial fit and returns the updated model.
* **Prediction module**: - This module is used for making predictions on today’s data after the model has been updated by the Incremental training module. It accepts the updated model and prepared data as parameters and generates the top 250 prediction report that can be used by the sales team to make calls.
* **PvA module**: - For a given day, this module generates a report comparing the actual conversions and predicted conversions that can be viewed by business managers.

After we created the modules and made sure that our solution was working, we used the Flask API to create a web application and hosted it as a web service using Heroku. The interface looks something like this…

# Challenges

* The first major challenge for me was the complexity of the project. I was not able to open and get the dataset working on my local machine.
* As they say in the industry, 70% of the time in a Data Science is taken to just exploring and prepare the data.
* Exploring more Machine Learning models which we could not do.