Predicting Unanticipated Overtime using Machine Learning

Time to clock out. Construction workers on scaffolding (© Bits and Splits/Shutterstock)

Jay Kalantar

2023-09-12

Preventing Unanticipated Overtime – Summary

1 Problem Statement

Unanticipated overtime (OT) costs are a common problem for many service sector businesses, as they can erode their profit margins and affect their employees' well-being. To help them address this problem, I applied various machine learning methods to determine the main factors that can forecast the risk of overtime occurrence. This study can offer our clients valuable insights and suggestions on how to improve their workforce management and reduce overtime costs.

2 Outcomes

The occurrence of overtime can be predicted accurately by using Decision Tree or Logistic Regression models. These models use the following features to determine the likelihood of overtime:

- Positive Correlations:
  - High percentage of incomplete or open punches
  - High percentage of time changes
  - High correction lags in fixing punch exceptions
  - More supervisors per employee
  - Increased daily punch count per employee
- Negative Correlations:
  - Higher number of employees doing the same task
  - Number of sites visited by an employee daily

3 Data Sources

Transactional data from one of our larger clients from January 1, 2017 to January 31, 2017. I then tested the model against their 2018 data. The data includes the following:

- **Timecards**: This section shows the employee's time in and time out for each day, along with other relevant fields such as employee number, location, position, etc.
- **Timesheets**: This section shows the employee's calculated pay for each pay period, including regular pay, overtime pay, expenses, etc.
- **Correction Lags**: This section shows the Timecard correction lag times in hours for any modified Timecard. This helps to identify any adjustments
- **Daily Budget:** This section is the expected daily budget for each task at a worksite.

4 Data Preparation

First, data cleanups were performed to remove irrelevant or inaccurate data, such as data added after the pay period, PTO data, and rows without time entries. Next, feature engineering is done to create new variables and metrics from the existing data, such as the number of punches per day, total REG and OT hours per day, the budget by day-of-week, the correction counts and lag hours, and the combination of punches and correction lags. Finally, all the data sources are consolidated into a single dataframe for further analysis and modeling.

5 Models

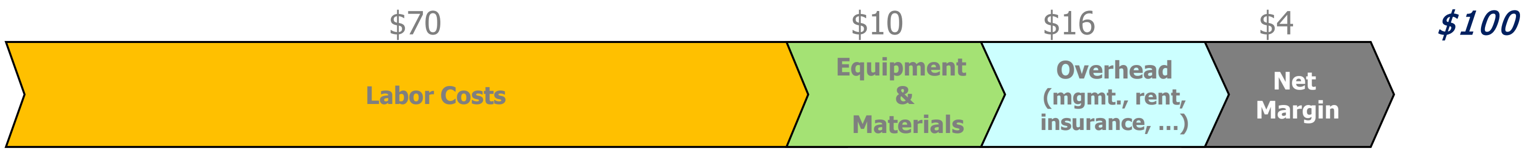
A comparative analysis of different machine learning models for predicting overtime occurrence was performed. The dataset was highly unbalanced, with only 3% of the cases having overtime occurrence. Therefore, high Precision and Recall scores were needed to accurately identify overtime cases and reduce false positives and negatives. After tuning, Decision Tree and Logistic Regression models showed good performance on this task. The most important features for predicting overtime occurrence were also identified based on the feature importance scores of the models.

6 Model Evaluation

Both our Decision Tree and Logistic Regression models achieved excellent results, with Recall and Precision scores above 90% for each of them. I tested the robustness of our models by applying the one I trained on 2017 data to the 2018 data, and I obtained similar results. I also used the models to make predictions for a different client over a longer period in 2019, and I obtained similar results. These results demonstrate that our models were effective and captured factors that are consistent across multiple clients.

Business Background

Our clients consist of service contractor operations that have high labor contents and small profit margins. According to the Construction Financial Management Association, average profit margins for general contractors is at 2.2-3.5%. This is what a typical value chain for a service contractor looks like:



Labor costs also influence the overhead costs due to taxes, worker's comp, and other payroll-related expenses. So, ***reducing labor costs can boost the profit margin significantly***.

One of the factors that affect labor costs is Overtime (OT). OT occurs when employees work more than their regular hours and get paid higher than the normal hourly rate (1.5X, 2X, …). Most service contractors avoid OT because it increases labor costs and creates stress for the employees and clients.

Business Goals

I aimed to build machine learning models that can do two things: first, to forecast when overtime happens, and second, to identify what factors can help managers prevent them.

I focused on the operational drivers that the client can influence and ignored factors specific to the activity, such as specific worksites, tasks/positions, employees, or regions. This would help us generalize our findings across all clients.

Our ultimate goal is to help our clients improve the efficiency and well-being of their employees, as well as reduce the costs and risks associated with overtime.

In Progress

Data Acquisition

Our system collects time entries by users and converts them to calculated pay data. The following datasets describe most of the transactional activities in the system:

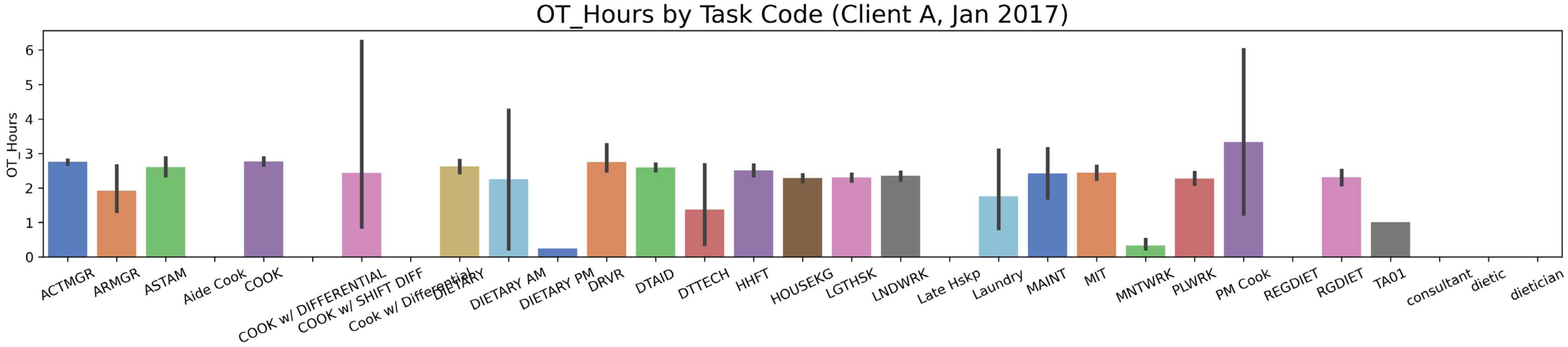
- **Timecards** – Employee's Time IN/OUT with all associated fields (e.g., Emp No, Location, Position, ….)
- **Audits** – Archive of timecard adjustments (e.g., Original Time IN, Current Time IN, …)
- **Timesheets** – Employee's calculated pay (e.g., REG Pay, Overtime Pay, Expenses, …)
- **Budgets / Schedules** – Planned Budget or attendance for the upcoming timecards.
- **Employees** – Employee demographics such as the hire dates, base salary, and positions
- **Managers** – Manager details such as title, number of employees, and locations they manage

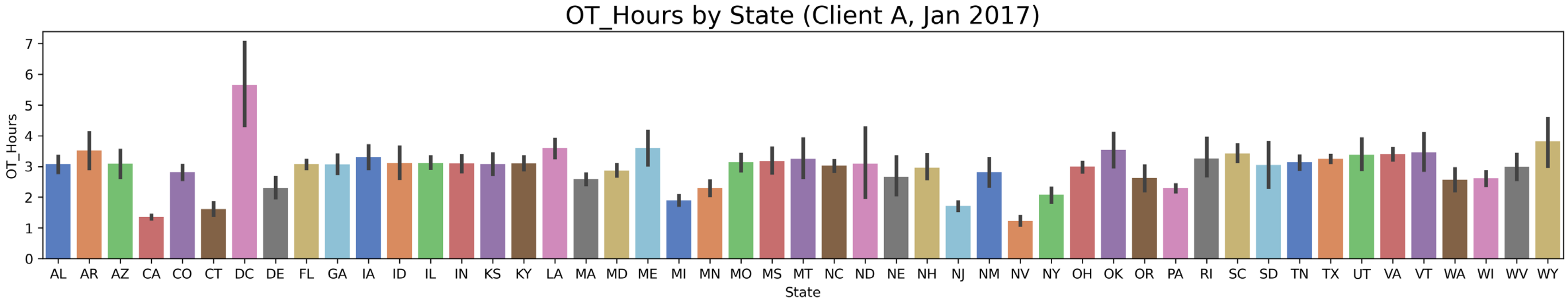
In addition, the system tracks all record manipulations and can provide the following adjustment categorizations:

- **OPEN punches** - List of punches with IN and/or OUT added by a manager
- **Time Changed punches** - List of punches with time change (in minutes)
- **Miss Punches** - List of punches with missed FP, IVR, Face or GPS, Schedule, …
- **Correction Lags** - list of punch correction lag times in hours for any modified punch (correction – exception time stamp)

Ignoring Client-Specific Features

Ignored Features: One of the challenges of developing a model for multiple clients is that each client may have unique features that can bias the model (e.g., emp no, tasks, states, address, ..). To avoid creating multiple versions of the same model, I ignore any client-specific features and focus on the core features that are common to all clients. This way, the model can be more maintainable, scalable, and reusable. Here are some examples of ignored client-specific features:





Selecting Features

Feature Engineering: To identify the key factors that affect the client's performance, I applied feature engineering techniques to the data. I selected the features under the client's control and can be modified to improve the outcomes. This way, we can ensure that our analysis is relevant and applicable to any client situation.

I used the following steps to clean and transform the data:

- Removed irrelevant or inaccurate data, such as data added after the pay period, PTO data, and rows missing time entries.
- Created new variables and metrics from the data:
  - **PunchCount**: Number of daily punches by the same employee on the same task.
  - **EmpCount**: Number of different employees performing the same task at a worksite daily.
  - **SiteCount**: Number of different sites an employee attends daily performing the same task.
  - **MgrCount**: Number of different supervisors adjusting an employee's timecards daily.
  - **Corr_Lag**: Time lags in hours for employee's timecard adjustment by type
  - **%Exceptions**: % of daily employee's timecard adjustment by type
  - **OT_Hours** is the dependent variable that measures an employee's resulting overtime hours in a given day.

The resulting data set was ready for further analysis and visualization.

Selected Features

```
Int64Index: 848077 entries, 0 to 848076
Data columns (total 11 columns):
 #   Column                    Non-Null Count   Dtype
---  ------                    --------------   -----
 0   PunchCount                848077 non-null  int64
 1   EmpCount                  848077 non-null  int64
 2   SiteCount                 848077 non-null  int64
 3   Corr_Lag_Hr_Miss_Punch    848077 non-null  float64
 4   Corr_Lag_Hr_Open_Punch    848077 non-null  float64
 5   Corr_Lag_Hr_Time_Changed  848077 non-null  float64
 6   MgrCount                  848077 non-null  float64
 7   %Miss_Punch               848077 non-null  float64
 8   %Open_Punch               848077 non-null  float64
 9   %Time_Changed             848077 non-null  float64
 10  OT_Hours                  23708 non-null   float64
dtypes: float64(8), int64(3)
memory usage: 77.6 MB
```

Modeling Techniques

Explored several machine learning methods that can predict overtime occurrence and determine the main factors. Based on our preliminary assessment, I chose the following methods due to their high performance and fast training times.

Decision Tree

Decision tree classification is a method of predicting the class or outcome of a data point based on a set of rules derived from a decision tree algorithm. A decision tree is a graphical representation of a decision process that resembles a tree structure. It has a root node that corresponds to the main question, and then it splits into branches that represent the possible choices or conditions. *Decision tree models can deal with nonlinear relationships and unbalanced data, which makes it particularly suitable for our OT use case.*

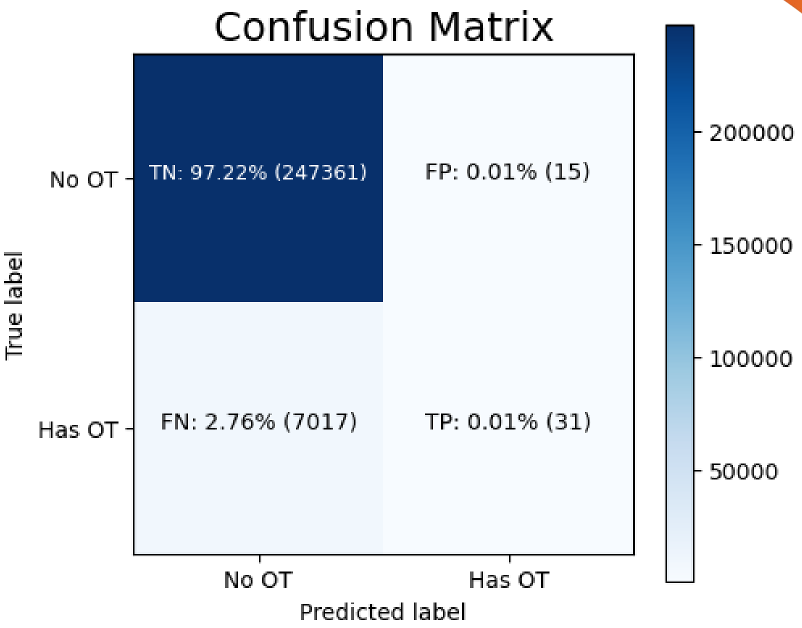
Logistic Regression

Logistic regression is a method of statistical analysis that can classify data into different groups based on how likely they are to have a certain outcome. It does this by using a mathematical equation called the logit function on a combination of the input variables. The equation can be fine-tuned to match the data better by changing the values of the coefficients or weights of the input variables. *Logistic regression quantifies the relationship between model features which can be useful for understanding how different features influence each other and the outcome variable.*

Model Evaluation

I evaluate the models by examining how many predictions are correct. A **Confusion Matrix** is a way to visualize the accuracy of a classification model.

One of the challenges of machine learning is dealing with **unbalanced** data. These datasets have a significant class imbalance (e.g., our data with ~97 %REG vs. 3 %OT). This can compromise the performance and accuracy of the machine learning models, as they might be skewed towards the most frequent class and neglect to learn the features of the less frequent class. ***Precision** and **Recall** are more informative than **Accuracy** when working with unbalanced datasets,*



Precision is the fraction of true positives out of all the predicted positives. It indicates how reliable the model is when it predicts a positive case.

Recall is the fraction of true positives out of all the actual positives. It indicates how effective the model is at capturing the positive cases.

Accuracy is the fraction of correct predictions out of all the predictions. It indicates how well the model can predict both positive and negative cases.

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

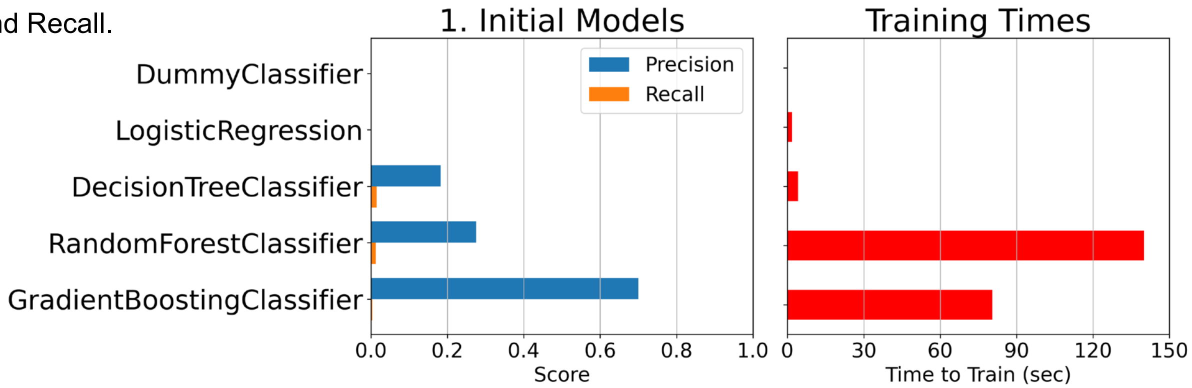
$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

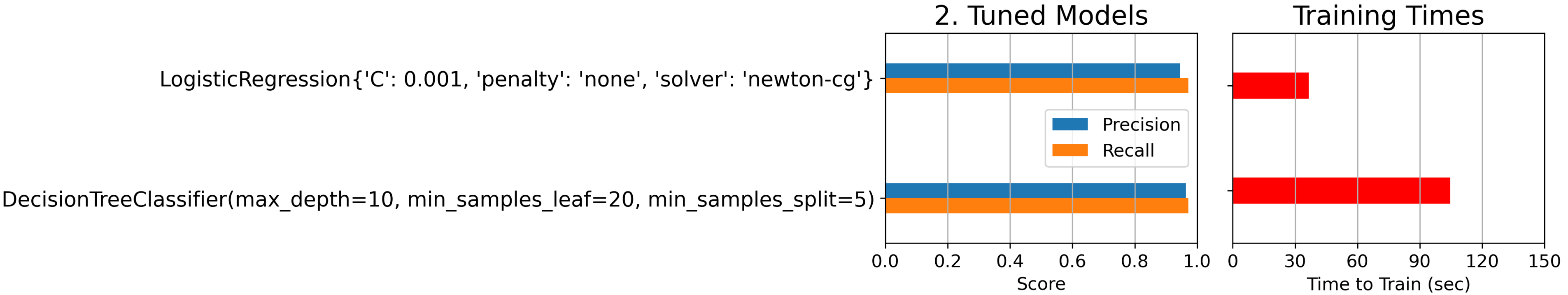
Model Tuning

I followed a two-step approach to find the best machine-learning techniques for prediction and feature selection:

1. I compared different methods based on their performance and training times.
2. I selected Logistic Regression and Decision Tree models because they fit our problem well and could be trained quickly. I tuned their hyperparameters using grid search and cross-validation.

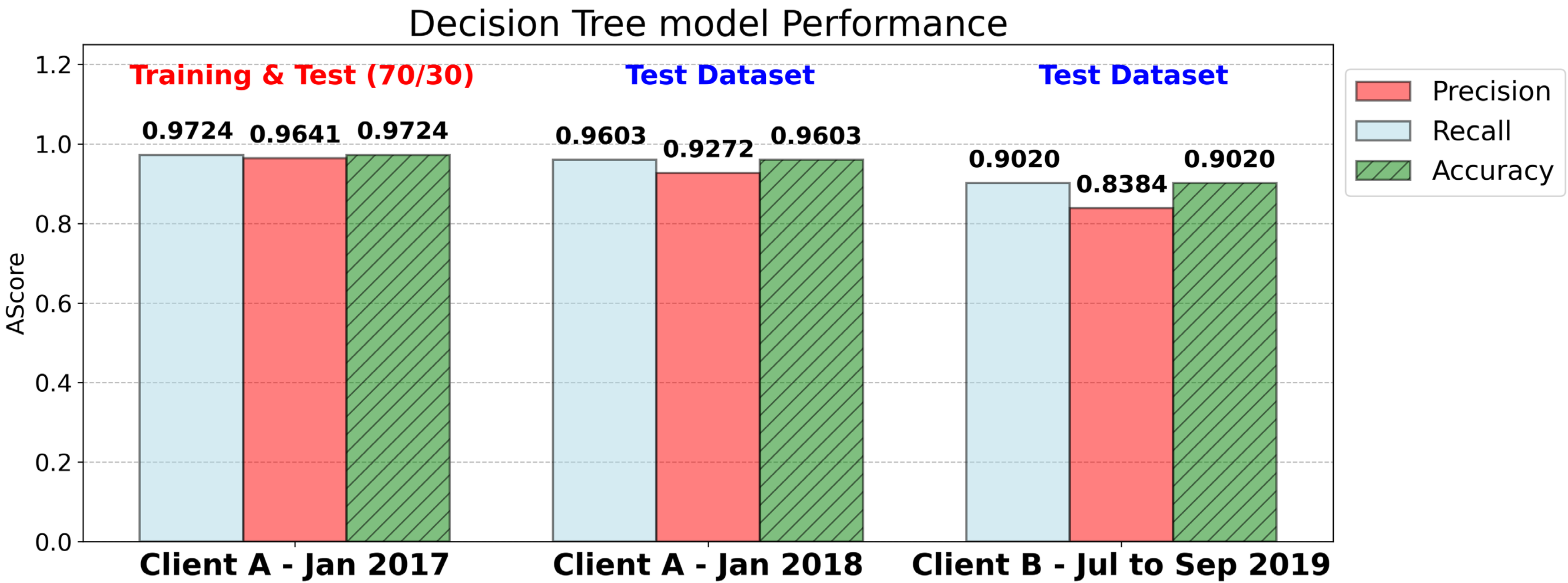
Both models performed well with high Precision and Recall.





Model Performance

Model Trained on the Client A's Jan 2017 datasets, performed well for different dates and another client. This suggests that the identified operational drivers are applicable beyond our training dataset.

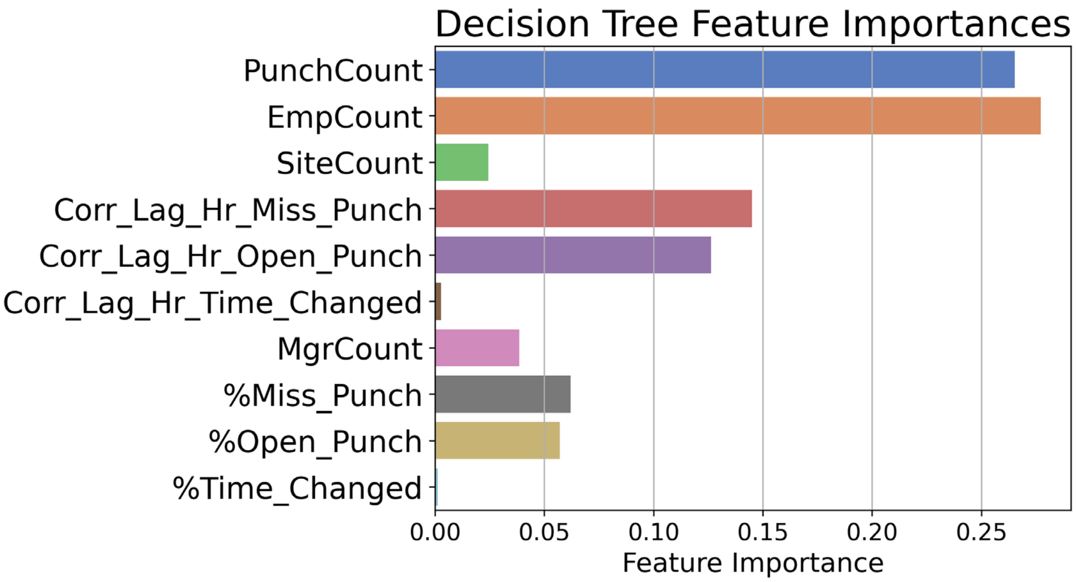


Feature Impacts

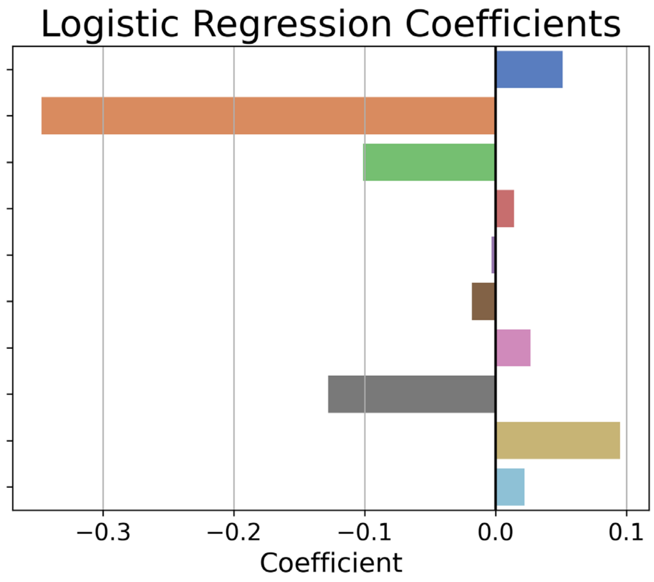
Given the difference in the way that decision trees and logistic regression learn from the data, it's expected that the two models would produce different results when applied to the same dataset. The Decision Tree model is much better suited to handle complex relationships between the features and our target variable. However, the logistic Regression model is easier to interpret and understand:

- Feature Importance in the Decision Tree describe their impact on the OT occurrences
- Feature Coefficient in Logistic Regression describe their correlations (positive vs negative) with OT occurrences

Feature Importance



Feature Coefficient



Findings

Our analysis of different machine learning models shows that tuned Decision Tree or Logistic Regression models can perform well in predicting overtime occurrence. Both models showed high Precision and Recall scores, meaning they can correctly classify overtime occurrence cases and minimize false positives and negatives. I also identified the most influential features that affect overtime occurrence, based on the feature importance scores of the models. The features that have a positive correlation with overtime are:

- **High percentage of incomplete or open punches**: This feature measures the percentage of punches that are not properly recorded or closed by the employees. A high value of this feature indicates a lack of compliance or awareness of the punching compliance, which may lead to overtime occurrence.
- **Higher number of daily punches by the same employee**: This feature measures the number of punches per day by each employee. A high value of this feature indicates a high frequency or intensity of work, which may increase the likelihood of overtime occurrence.
- **High percentage of time changes**: This feature measures the percentage of punches modified or corrected by the supervisors. A high value of this feature indicates a high variability or uncertainty in the working hours, which can result in overtime occurrence.
- **High correction lag**: This feature measures the average number of days between the date of a punch exception and the date of its correction. A high value of this feature indicates a delay or inefficiency in resolving the punch exceptions, which may cause overtime occurrence.
- **Higher number of supervisors per employee**: I determined the number of supervisors based on who made the daily corrections for each employee. A higher number may indicate lack of direct supervision that may lead to higher overtime occurrence.

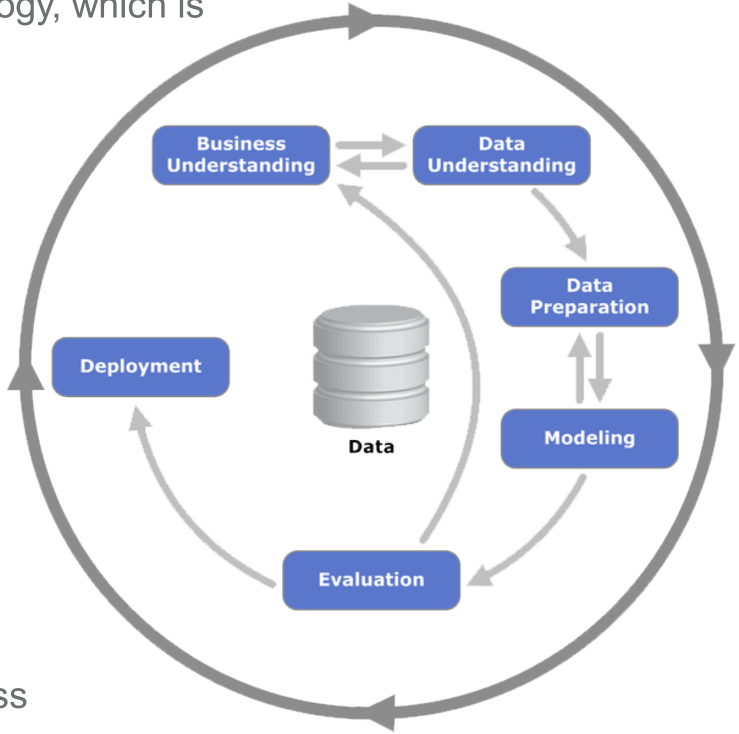
The factors that have a negative correlation with overtime are:

- **Number of employees**: This feature measures the number of employees who perform the same task or belong to the same group. A low value of this feature indicates a low availability or redundancy of resources, which may increase the demand or pressure on the existing employees, leading to overtime occurrence.

These findings can help managers and supervisors monitor and control these factors and reduce the risk of overtime occurrence.

Next Steps

To carry out this case study, I followed the CRISP-DM methodology, which is shown here:



I trained and tested our machine-learning models using a month of data from one client (Client A, Jan 2017). I then evaluated the models with new data from the same client (Client A, Jan 2018) as well as another client (Client B, Jul to Sept 2019).

To achieve more reliable and general results, we need to expand our data sources. We should include data from various clients and time periods. This will help us evaluate and enhance our models' accuracy and relevance before deployment.

Machine learning models can help us address important business problems. We should continue to build and refine our models, but we should also look for the optimal way to deploy them.

Deployment

API or web services components:

- Using API or web services component allows us to load one model into multiple applications by calling the API or web service, which optimizes the model performance.
- **Microsoft Azure** provides ML/AL API and web services deployment services to build, test, deploy, and manage our APIs and web services. Azure supports various deployment scenarios, such as real-time processing, batch processing, or generative AI. Azure also offers both Python SDK and Azure Machine Learning Studio, which gives us a flexible and user-friendly web interface for machine learning, to create, train, and deploy models with drag-and-drop or code-based options.

Dashboards:

- Display the model prediction in a dashboard that provides visual and interactive ways for users to monitor, explore, and understand the data and the model outputs.
- **Microsoft Power BI** provides business intelligence and data visualization tools. Power BI also connects with the Azure Machine Learning models, to create reports and dashboards with charts, maps, tables, and filters.

Other Relevant Modeling Applications

One of the services we could offer to our clients is **Overtime Cost Reduction**. This is a data-driven approach to help our clients minimize the expenses associated with paying overtime to their employees. We use time-series analysis methods such as ARMA or LSTM to model the patterns and trends of overtime over time and predict future overtime levels. This can provide our clients with actionable insights and recommendations on how to optimize their workforce management and reduce overtime costs.

Another service we could offer is **Demand Forecasting**. This is a process of data analysis to predict the future demand for labor and resources in a business. It can help our clients to optimize their budgeting and scheduling decisions and reduce costs and inefficiencies. By using historical data and seasonal patterns, we can build models that can forecast the number of hours and employees needed for each position on a specific date. These models are based on time-series or sequential methods (ARMA, LSTM), which capture the temporal dynamics and trends of the data. Demand Forecasting can provide our clients with valuable insights and recommendations on allocating their workforce and resources effectively and efficiently to avoid overstaffing or understaffing, which can lead to unnecessary overtime costs or missed service levels.

A third service we could offer is **Employee Retention** Analysis. This is a data science technique to identify the factors influencing employee retention and satisfaction. We can build predictive models that classify employees into groups based on their likelihood of leaving or staying. We can also use regression models to estimate the effects of various variables on employee turnover. Some possible models we can use are Linear, SVM, Decision Tree, Random Forest or Gradient Boosting. These models can help us find the most important features that affect employee turnover and provide insights for developing effective retention strategies.